



**HAL**  
open science

## Learning methods for digital imaging

Prakhar Amba

► **To cite this version:**

Prakhar Amba. Learning methods for digital imaging. Artificial Intelligence [cs.AI]. Université Grenoble Alpes, 2018. English. NNT : 2018GREAS011 . tel-02370906

**HAL Id: tel-02370906**

**<https://theses.hal.science/tel-02370906>**

Submitted on 19 Nov 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

## THÈSE

Pour obtenir le grade de

### **DOCTEUR DE LA COMMUNAUTÉ UNIVERSITÉ GRENOBLE ALPES**

Spécialité : **CIA - Ingénierie de la Cognition, de l'interaction, de  
l'Apprentissage et de la création**

Arrêté ministériel : 25 mai 2016

Présentée par

**Prakhar Amba**

Thèse dirigée par **Martial MERMILLOD**, Prof. UGA  
et codirigée par **David ALLEYSSON**, CR CNRS

préparée au sein du **Laboratoire de Psychologie et NeuroCognition** et  
de l' **École doctorale Ingénierie pour la Santé, la Cognition et  
l'Environnement**

## **Learning methods for digital imaging.**

Thèse soutenue publiquement le **3 Mai 2018**

Jury composé de :

**Monsieur LUDOVIC MACAIRE**

PROFESSEUR, UNIVERSITÉ DE LILLE, Rapporteur

**Monsieur EDOARDO PROVENZI**

PROFESSEUR, UNIVERSITÉ DE BORDEAUX, Rapporteur, Présidente

**Monsieur CLAUDE BERROU**

PROFESSEUR, IMT ATLANTIQUE, ACADÉMIE DES SCIENCES , Examineur

**Monsieur ALEJANDRO RIBÉS**

CHERCHEUR, EDF PARIS-SACLAY , Examineur

**Monsieur MARTIAL MERMILLOD**

PROFESSEUR DES UNIVERSITÉS, UNIV-GRENOBLE ALPES & CNRS, Directeur  
de thèse

**Monsieur DAVID ALLEYSSON**

CHARGÉ DE RECHERCHE, CNRS DÉLÉGATION ALPES, Co-Directeur de thèse





# Learning methods for digital imaging.

## Méthodes d'apprentissage pour l'imagerie numérique.

by

Prakhar AMBA

### Abstract

In this work we discuss about shift-invariant problems in digital imaging which could be solved by learning linear least square solution over exemplars. We focus our research on color demosaicing which is one of the famous shift-invariant problem in digital imaging; the ability to estimate a full resolution color image from a subsampling one, acquired through a matrice of different color filters.

To produce color images we need information of three primary colors (notably Red, Green and Blue) at each pixel point. To capture this information most digital cameras utilize a Color Filter Array (CFA); a mosaic arrangement of these primary colors is overlaid on the sensor such that only a single color is sampled at one pixel. Here we ask whether the most commonly used CFA called Bayer is the best compromise between space and color or if we should used larger shift-invariant pattern or different color filters. Furthermore, we ask what could be the ideal spectral transmittance of the filters for a given application.

This way of acquiring images is similar to the Human Visual System (HVS) wherein a mosaic of LMS cones (for sensitivity to Long, Medium and Short wavelength) forms the surface of the retina. For HVS, the arrangement is random and differs between individuals, whereas for cameras we use a regular arrangement of color filters. We show by simulation that there is an advantage of having a random sampling of colors instead of a regular.

The operation for a digital camera to interpolate the missing colors to recover the full color image is known as demosaicing. Demosaicing could be simulated on a subsampled image having a single color per pixel correspondingly to the CFA. Due to regular or periodic arrangement of color filters the output demosaiced image is susceptible to false colors and artifacts. In literature, the demosaicing algorithms proposed so far cater mainly to regular CFAs.

We propose an algorithm for demosaicing which can be used to demosaic any random as well as regular CFA by learning statistics on an image database. We show that the solution obtained is unique because it correspond to the linear least square estimate of a shift-invariant inverse problem. Managing the neighborhood into the vectors improves redundancy of the estimate and provides a stable solution. Surprisingly, this solution is close to the state of the art demosaicing. Based on our method, we optimize and propose new CFAs such that they outperform even the state of art algorithms on regular (Bayer) CFAs. At the same time the demosaiced images from proposed CFAs are free from false colors and artifacts.

We extend our algorithm such that it is not limited to only three colors but can be used for any number of spectral filters. Having more than three colors allows us to not only record an image but to record a spectral signature of the scene. We called Spectral Filter Arrays (SFAs) a mosaic for which we know the transmittance of the filter and not only their "color" (part of the spectrum with maximum transmittance). Recent technological advances give us greater flexibility in designing the spectral filters and open the door to new applications. Because silicon is inherently sensitive to Near-Infrared (NIR) radiation, both Visible and NIR filters can be combined on the same mosaic. We show that our method applies very well on RGB-NIR mosaic as far as we have an accurate database to learn from.

Beyond simulation, we apply our algorithm on several real cameras having SFAs by using the RAW image extracted. We demonstrate that our method outperforms the state of art algorithms in image quality and computational efficiency. We propose a method to optimize filters transmittance and their arrangement such that it gives best results depending on evaluation metrics and application chosen.

Our method is linear and therefore very fast and suitable for real time applications. Finally, to challenge the linear nature of LMMSE we propose a demosaicing algorithm using Neural Networks trained on a small database of images which provides slightly better reconstruction than the linear demosaicing, however, it is computationally more expensive.

**Keywords: Demosaic, Color Filter Array, Spectral Filter Array, LMMSE, Neural Network**

# Learning methods for digital imaging.

## Méthodes d'apprentissage pour l'imagerie numérique.

by

Prakhar AMBA

### Abstract

Dans ce travail, nous discutons des problèmes invariants par translation en imagerie numérique qui peuvent-être résolus par apprentissage sur des exemples de la solution linéaire aux moindres carrés. Nous concentrons notre recherche sur le démosaïçage couleur qui est le problème invariant par translation le plus fameux en imagerie numérique; la possibilité d'estimer une image couleur résolue à partir d'une image sous-échantillonnée, acquise à travers une matrice de filtres de différentes couleurs.

Pour produire des images couleurs nous devons obtenir l'information relative aux trois couleurs primaires (généralement Rouge, Vert et Bleu) à chaque pixels de l'image. Pour capturer cette information la plupart des caméras numériques utilisent une matrice de filtres couleurs (CFA – Color Filter Array en anglais), c'est-à-dire qu'une mosaïque de couleurs recouvre le capteur de manière à ce qu'une seule couleur soit mesurée à chaque position dans l'image. On peut se demander si le plus utilisé des CFA, dit de Bayer, est le meilleur compromis entre espace et couleur ou s'il faut utiliser un motif invariant par translation plus grand ou différents filtres de couleur. De plus, on peut demander quels seraient les transmittances idéales des filtres pour une application donnée.

Cette méthode de mesure à travers une mosaïque est similaire à celle du système visuel humain (HVS – Human Visual System en anglais) pour lequel les cônes LMS (sensibles aux longues L, moyenne M et courte S (short en anglais)) forment également une mosaïque à la surface de la rétine. Pour le système visuel, l'arrangement est aléatoire et change entre les individus alors que pour les caméras nous utilisons des arrangements réguliers. Nous montrons par simulation qu'il y a un avantage d'avoir un échantillonnage aléatoire plutôt que régulier.

L'opération qui consiste à interpoler les couleurs manquantes dans une caméra est appelé démosaïçage. Le démosaïçage peut-être simulé sur une image sous-échantillonnée ayant une seule couleur par pixel correspondante à la matrice de filtres (CFA). A cause de l'arrangement régulier ou périodique des filtres couleurs l'image reconstruite est susceptible de contenir des fausses couleurs et des artefacts. Dans la littérature, les algorithmes de démosaïçage proposés s'appliquent principalement aux CFA réguliers.

Nous proposons un algorithme de démosaïçage par apprentissage statistique, qui peut être utilisé avec n'importe quelle mosaïque régulière ou aléatoire. Nous montrons que la solution obtenue est unique parce qu'elle correspond à l'estimée au moindre carré de la solution d'un problème inverse invariant par translation. L'utilisation d'un voisinage dans le modèle vectoriel améliore la redondance et l'estimation est

plus robuste. De manière surprenante la solution obtenue est proche de l'état de l'art en performance. Grâce à cette méthode nous optimisons et proposons de nouvelles matrices de filtres de couleur (CFA) qui dépasse les meilleurs algorithmes sur le CFA de Bayer. En même temps, les images démosaïquées avec ces nouveaux arrangements sont sans fausses couleurs et artefacts.

Nous avons étendu l'algorithme pour qu'il ne soit pas limité à trois couleurs mais puisse être utilisé pour un arrangement aléatoire d'un nombre quelconque de filtres spectraux. Avoir plus de trois couleurs permet non seulement de mieux représenter les images mais aussi de mesurer des signatures spectrales de la scène. Nous appellerons une mosaïque, matrice de filtres spectraux (SFA – Spectral Filter Array en anglais) en opposition à matrice de filtres couleurs (CFA) lorsque nous connaissons les transmittance des filtres et pas seulement leur "couleur" (partie du spectre de transmittance maximale). Les technologies récentes nous offrent une grande flexibilité pour définir les filtres spectraux et ouvrent la porte à de nouvelles applications. Le substrat silicium dans lequel les photodiodes du capteur sont réalisées est sensible aux radiations proche infra-rouge et donc des filtres visibles et proche infra-rouge peuvent-être combinés dans la même mosaïque. Nous montrons que notre méthode s'applique très bien sur les mosaïques RGB-NIR à condition que nous ayons une base de donnée appropriée pour apprendre.

Plus loin que la simulation nous appliquons notre algorithme sur plusieurs cameras réelles équipées de SFA en extrayant l'image RAW. Nous démontrons la supériorité de notre méthode sur les algorithmes de l'état de l'art en terme de qualité d'image et de vitesse de calcul. Nous proposons une méthode pour optimiser les transmittances des filtres et leur arrangement de manière à ce qu'ils délivrent les meilleurs résultats en fonction des métriques d'évaluation et de l'application choisie.

La méthode est linéaire et par conséquent rapide et applicable en temps réel. Finalement, pour défier la nature linéaire de notre algorithme, nous proposons un deuxième algorithme de démosaïçage par réseaux de neurones qui à des performances légèrement meilleures mais pour un coût de calcul supérieur.

**Mots clefs: Démosaïçage / Dématricage, Matrice de filtres couleurs, Matrice de filtres spectraux, Minimisation linéaire de l'erreur quadratique moyenne, Réseaux de Neurones**

# Acknowledgments

First I would like to thank Dr. David Alleysson for having faith in me and guiding me through this work. It wouldn't be possible without the endless discussion with him over cups of coffee. I thank him for introducing me to the wonderful and complicated world of science of colors and that of its perception.

Further I wish to thank my wife Aastha Amba for being there for me always and willing to change continents with a baby in her arms.

I would like to dedicate this thesis to my daughter Pranavi for being the joy in my life. Pranavi has dual meaning in Hindi, one 'full of *Prana* (life)' and secondly *Parvati* consort to Shiva, daughter of mountain king Himavan (personification of Himalayas) in Hindu mythology. It is to her I credit to bringing us to Grenoble, the capital of Alps.

I would also like to thank my parents, my mother Meena Amba for inculcating my love of books, my father Vinay Amba for letting me explore my dreams.

I would like to thank all my teachers who contributed to my education and becoming what I am today, from teaching me how to hold a pencil, to working out calculus and teaching me how to code. Teachers who kindled in me a joy to learn and to discover the unknown.

Finally, I would like to thank all the wonderful people of my lab LPNC, they are too many to name here. For welcoming me, bearing my incessant demands to "translate into French svp", sharing laughter and having fun. Its been great being part of this tribe for the last 3 years.





# Contents

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introduction</b>   | <b>23</b> |
| 1.1      | Thesis Outline . . . . .  | 29        |
| <b>2</b> | <b>Light and Color</b>  | <b>31</b> |
| 2.1      | Physics of light . . . . .  | 32        |
| 2.2      | Human Visual System . . . . .                                       | 33        |
| 2.3      | Digital Camera . . . . .  | 37        |
| 2.3.1    | Color Filter Arrays . . . . .                                       | 38        |
| 2.3.2    | Spectral Filter Arrays . . . . .                                    | 39        |
| 2.4      | Analogy between Camera and the Human Visual System . . . . .        | 41        |
| 2.5      | Image processing pipeline for a camera . . . . .                    | 42        |
| 2.5.1    | Demosaicing for Bayer CFA . . . . .                                 | 44        |
| 2.6      | Linear solution to the inverse problem . . . . .                    | 47        |
| <b>3</b> | <b>Demosaicing for Random RGB CFAs</b>                              | <b>51</b> |
| 3.1      | Introduction . . . . .  | 51        |
| 3.2      | Matrix model of image formation . . . . .                           | 52        |
| 3.2.1    | Sliding vs constant neighborhood . . . . .                          | 58        |
| 3.3      | Simulation . . . . .  | 62        |
| 3.4      | Results . . . . .   | 63        |
| 3.4.1    | Systematic evaluation for 2x2 super-pixel size of the CFA . . . . . | 63        |
| 3.4.2    | Systematic evaluation for 3x3 super-pixel size of the CFA . . . . . | 66        |
| 3.4.3    | Evaluation of 4x4 super-pixel size of the CFA . . . . .             | 68        |

|          |  |            |
|----------|--|------------|
| 3.4.4    | Comparison of CFAs under LMMSE . . . . .   | 69         |
| 3.4.5    | Effect of gaussian noise on LMMSE performance . . . . .                          | 74         |
| 3.4.6    | Comparison with other methods on Bayer . . . . .                                 | 76         |
| 3.4.7    | Effect of database on learning . . . . .   | 78         |
| 3.5      | Simulating totally random CFAs . . . . .   | 81         |
| 3.6      | Discussion . . . . .   | 85         |
| 3.7      | Conclusion . . . . .   | 87         |
| <b>4</b> | <b>Multicolor CFA's</b>  | <b>89</b>  |
| 4.1      | Introduction . . . . .   | 89         |
| 4.2      | Matrix Model for Demosaicing for multicolor CFAs . . . . .                       | 91         |
| 4.3      | Finding Optimum CFA arrangement . . . . .  | 93         |
| 4.3.1    | Results . . . . .  | 95         |
| 4.3.2    | DFTs of proposed CFAs . . . . .  | 97         |
| 4.4      | Discussion . . . . .   | 97         |
| 4.5      | Conclusion . . . . .   | 99         |
| <b>5</b> | <b>Demosaicing for Spectral Filter Arrays (SFAs)</b>                             | <b>101</b> |
| 5.1      | Introduction . . . . .   | 101        |
| 5.2      | Linear model for SFA demosaicing . . . . .                                       | 103        |
| 5.2.1    | Model of linear SFA image formation . . . . .                                    | 103        |
| 5.2.2    | N-LMMSE formulation . . . . .  | 107        |
| 5.3      | JB Camera . . . . .  | 110        |
| 5.3.1    | Quantitative analysis in simulation . . . . .                                    | 110        |
| 5.3.2    | Analysis on real images . . . . .  | 113        |
| 5.3.3    | Visualization of color images . . . . .  | 116        |
| 5.4      | Conclusion . . . . .   | 118        |
| <b>6</b> | <b>Spatio-Spectral Optimization of spectral filters transmission in the SFAs</b> | <b>123</b> |
| 6.1      | Introduction . . . . .   | 123        |

|          |  |            |
|----------|--|------------|
| 6.2      | Image formation workflow and quality metrics . . . . .                   | 124        |
| 6.2.1    | Note about filter response $F$ . . . . .                                 | 126        |
| 6.2.2    | Performance metrics . . . . .  | 128        |
| 6.2.3    | Parameter for Optimization by Correlation matrix . . . . .               | 132        |
| 6.3      | Evaluation of metrics in terms of Gaussian filters . . . . .             | 134        |
| 6.4      | Design of optimal spectral transmission sensitivity of filters . . . . . | 146        |
| 6.5      | Effect on demosaicing performance due to illuminant . . . . .            | 151        |
| 6.6      | Radiance recovery for Visible+NIR filter array . . . . .                 | 151        |
| 6.7      | Conclusion . . . . .   | 155        |
| <b>7</b> | <b>Demosaicing using Dual Layer Feedforward Neural Network</b>           | <b>157</b> |
| 7.1      | Introduction . . . . .   | 157        |
| 7.2      | Neural Network Configuration . . . . .                                   | 159        |
| 7.3      | Results . . . . .  | 161        |
| 7.3.1    | Testing for Color Filter Arrays . . . . .                                | 161        |
| 7.3.2    | Testing for Spectral Filter Arrays . . . . .                             | 164        |
| 7.3.3    | Monno5ch SFA with images from 5-band multispectral dataset               | 164        |
| 7.3.4    | Monno5ch SFA with RAW images captured from actual camera                 | 166        |
| 7.4      | Discussion . . . . .   | 168        |
| 7.5      | Conclusion . . . . .   | 170        |
| <b>8</b> | <b>Conclusion</b>  | <b>171</b> |
| 8.1      | Summary . . . . .  | 171        |
| 8.2      | Future Work . . . . .  | 174        |
|          | <b>Appendices</b>  | <b>176</b> |
| <b>A</b> | <b>Image databases used for training</b>                                 | <b>177</b> |
| <b>B</b> | <b>Demosaicing comparison on CFAs</b>                                    | <b>185</b> |
| B.1      | LMMSE on Different CFAs . . . . .  | 185        |
| B.2      | Comparison of LMMSE with other Demosaicing algorithms . . . . .          | 185        |

|          |   |            |
|----------|---|------------|
| B.3      | Effect of Leave one out . . . . .                               | 189        |
| B.4      | DFTs on Different CFAs . . . . .                                | 190        |
| <b>C</b> | <b>Spectral Filter Arrays</b>                                   | <b>191</b> |
| C.1      | Estimating Sensor Spectral Sensitivity . . . . .                | 191        |
| <b>D</b> | <b>Comparison with Compressive Sensing demosaicing for SFAs</b> | <b>195</b> |
| D.1      | Testing . . . . .   | 195        |

# List of Figures

|      |  |    |
|------|--|----|
| 1-1  | CFA on a single sensor camera . . . . .  | 24 |
| 2-1  | From Physics to Vision and Camera capture . . . . .  | 33 |
| 2-2  | Spectral Radiance measurements using Konica CS2000 for different<br>light sources . . . . .  | 34 |
| 2-3  | Color Matching Functions and LMS response . . . . .  | 36 |
| 2-4  | Images of cone mosaics of 10 human subjects . . . . .  | 37 |
| 2-5  | (a) Three sensor system, (b) Bayer CFA, (c) Random Human retina<br>mosaic <sup>1-3</sup> . . . . .   | 38 |
| 2-6  | Foveon Sensor <sup>4</sup> . . . . .   | 39 |
| 2-7  | Spectral Filter Arrays . . . . .   | 40 |
| 2-8  | Image Processing Pipeline <sup>5</sup> . . . . .   | 43 |
| 2-9  | Bayer CFA <sup>3</sup> . . . . .   | 44 |
| 2-10 | Different CFA types and corresponding spatial frequency response:<br>Left to right: Bayer, Diagonal, Yamanaka, Holladay, CNRS <sup>6</sup> . . . . . | 46 |
| 2-11 | LMMSE the idea of minimising the MSE . . . . .   | 47 |
| 3-1  | Image formation model for RGB CFA . . . . .  | 53 |
| 3-2  | Matrix unfolding of images by super-pixel . . . . .  | 56 |
| 3-3  | Matrix unfolding of images by super-pixel with sliding neighborhood . . . . .  | 57 |
| 3-4  | Illustration of sliding vs toy-constant and constant neighborhood . . . . .  | 59 |
| 3-5  | Illustration of expanded toy-constant and constant neighborhood . . . . .  | 60 |
| 3-6  | All possible CFAs of 2x2 size arranged by decreasing $\mu$ for Kodak<br>database . . . . .   | 64 |

|      |   |    |
|------|---|----|
| 3-7  | Histogram of $\mu$ for Kodak and McM database for all 2x2 CFAs . . . .  | 65 |
| 3-8  | Top Row: Left to Right (Lowest $\sigma_{rgb}$ , Highest $\mu$ , Lowest $\sigma$ ) for Kodak database. Bottom Row: Left to Right (Lowest $\sigma_{rgb}$ , Highest $\mu$ , Lowest $\sigma$ ) for McM database . . . . . | 65 |
| 3-9  | Top 100 CFAs by $\mu$ for Left(Kodak), Right (McM) for neighborhood 7   | 66 |
| 3-10 | Histogram of $\mu$ for Kodak and McM database for all 3x3 CFAs . . . .  | 67 |
| 3-11 | Top Row: Left to Right (Lowest $\sigma_{rgb}$ , Highest $\mu$ , Lowest $\sigma$ ) for Kodak database. Bottom Row: Left to Right (Lowest $\sigma_{rgb}$ , Highest $\mu$ , Lowest $\sigma$ ) for McM database . . . . . | 67 |
| 3-12 | Histogram of PSNR for 4x4 CFAs . . . . .  | 68 |
| 3-13 | Top 100 4x4 CFAs by average PSNR for Kodak database . . . . .   | 69 |
| 3-14 | Different CFA patterns used for evaluation . . . . .  | 70 |
| 3-15 | Evaluation of CFAs with LMMSE with increasing neighborhood . . .  | 71 |
| 3-16 | Leave one out evaluation . . . . .  | 80 |
| 3-17 | Random CFA super-pixel of size $512 \times 768$ applied on the Lighthouse image . . . . .   | 82 |
| 3-18 | Crop of CFA super-pixel, showing $32 \times 32$ pixels here . . . . .   | 82 |
| 3-19 | Histogram of average PSNR $\mu$ for Kodak image database for 1000 totally random CFAs . . . . .   | 83 |
| 3-20 | Evaluation of LMMSE on lighthouse image for totally random CFA .  | 83 |
| 3-21 | Left: Demosaiced Lighthouse image using KCS <sup>7</sup> . Right: Demosaiced using Group Sparse <sup>7</sup> . . . . .  | 84 |
| 3-22 | Left: Spectral Sensitivity Curves of Kodak Film 2273,3273/ESTAR Base <sup>8</sup> Right: Spectral Sensitivity curves of a Nikon D300 camera color filter . . . . .  | 85 |
| 4-1  | CFAs with RGB and multicolor CFAs <sup>3,9,10</sup> . . . . .   | 90 |
| 4-2  | Average DFT for Kodak database. First row (LtoR): Bayer <sup>3</sup> , Yamanaka <sup>11</sup> , Holladay <sup>12</sup> Second row(LtoR): Bai <sup>13</sup> , Hao40, Hao50 <sup>14</sup> . . .                         | 91 |

|      |   |     |
|------|---|-----|
| 4-3  | Matrix model of the multicolor CFA image formation without neighborhood . . . . .   | 93  |
| 4-4  | All multicolor CFAs evaluated along with proposed ones . . . . .  | 95  |
| 4-5  | Crop of Lighthouse image for proposed $2 \times 2m$ , $4 \times 4m_1$ , $4 \times 4m_2$ , $6 \times 6m$ , $8 \times 8m$ , and $10 \times 10m$ CFAs . . . . .                                      | 98  |
| 4-6  | Frequency Response of proposed $2 \times 2m$ , $4 \times 4m_1$ , $4 \times 4m_2$ , $6 \times 6m$ , $8 \times 8m$ , and $10 \times 10m$ CFAs calculated using Discrete Fourier Transform . . . . . | 98  |
| 5-1  | Image formation model from reflectance objects. . . . .   | 104 |
| 5-2  | Image formation Model illustrating the steps involved in getting the SFA image $X$ from the reflectance images $Z$ . . . . .  | 106 |
| 5-3  | Illustration of image formation model on reflectance images showing matrix multiplication . . . . .   | 108 |
| 5-4  | Linear matrix model of the SFA image formation without neighborhood. . . . .  | 108 |
| 5-5  | Illustration of linear matrix model of SFA image formation incorporating sliding neighborhood. . . . .  | 109 |
| 5-6  | Spectral Filter Array(SFA) mosaic arrangement of filters & their spectral response for the JB camera . . . . .  | 111 |
| 5-7  | Binary tree results on the 8 spectral channels. . . . .   | 115 |
| 5-8  | N-LMMSE results on the 8 spectral channels. . . . .   | 115 |
| 5-9  | Visualization of SFA images demosaiced by the binary-tree and rendered in sRGB space. . . . .   | 118 |
| 5-10 | Visualization of SFA images demosaiced by N-LMMSE and rendered in sRGB space. . . . .   | 118 |
| 6-1  | Image formation model from reflectance objects with the workflow to get sRGB images . . . . .   | 125 |
| 6-2  | Sensor spectral response for two different Silicon based sensors . . . . .  | 127 |
| 6-3  | Image formation model illustrating the different parameters for optimization . . . . .  | 131 |
| 6-4  | Effect of increasing $\lambda_\sigma$ on filter's spectral response . . . . .   | 135 |



|      |  |     |
|------|--|-----|
| 6-5  | Filters for SFAs with $\lambda_\sigma=4$ . . . . .   | 136 |
| 6-6  | Effect of $\lambda_\sigma$ on demosaicing for the Macbeth chart in Finlayson database for Bayer SFA . . . . .                        | 138 |
| 6-7  | Effect of $\lambda_\sigma$ on demosaicing with white-balancing for the Macbeth chart in Finlayson database for Bayer SFA . . . . .   | 139 |
| 6-8  | Effect of $\lambda_\sigma$ on demosaicing for the Macbeth chart in Finlayson database for 2x2 SFA size, 4 filters . . . . .          | 140 |
| 6-9  | Effect of $\lambda_\sigma$ on demosaicing for the Macbeth chart in Finlayson database for 3x3 SFA size, 9 filters . . . . .          | 141 |
| 6-10 | Effect of $\lambda_\sigma$ on demosaicing for the Macbeth chart in Finlayson database for 4x4 SFA size, 16 filters . . . . .         | 142 |
| 6-11 | Illustration of patches on macbeth chart for evaluating $\mu_z(\text{chart})$ . . .  | 143 |
| 6-12 | Recovered reflectance values for the 24 colors in the Macbeth chart image  | 143 |
| 6-13 | Optimized filters with $\Delta E_3$ as the criteria for Bayer SFA . . . . .  | 146 |
| 6-14 | Optimized filters for different criteria for neighborhood of 7, optimized on the Finlayson database . . . . .                        | 148 |
| 6-15 | Optimized filters with $\mu_3$ as the criteria . . . . .   | 149 |
| 6-16 | Simulated Gaussian Filters on the Visible+NIR wavelength . . . . .   | 152 |
| 6-17 | Recovered Radiance from 4 different pixel points on the StandfordDish image from SCIEN for SFA 3x3 $\lambda_\sigma=6$ . . . . .      | 153 |
| 6-18 | Recovered Radiance from 4 different pixel points on the StandfordDish image from SCIEN for SFA Bayer $\lambda_\sigma=0.25$ . . . . . | 153 |
| 7-1  | Neural Network mode . . . . .  | 160 |
| 7-2  | Lighthouse Image crop. Demosaiced using Neural Network method for different CFAs . . . . .   | 163 |
| 7-3  | CFAs and SFA tested using Neural Network . . . . .   | 164 |
| 7-4  | Demosaiced Image from 5-band TokyoTech set data, example 1 . . .   | 165 |
| 7-5  | Demosaiced Image from 5-band TokyoTech set data, example 2 . . .   | 166 |
| 7-6  | Demosaiced RAW Image from Monno5ch SFA, example 1 . . . . .  | 167 |

|     |   |     |
|-----|---|-----|
| 7-7 | Demosaiced RAW Image from Monno5ch SFA, example 2 . . . . .   | 167 |
| 7-8 | Demosaiced RAW Image from Monno5ch SFA, example 3 . . . . .   | 168 |
| 7-9 | Demosaiced RAW Image from Monno5ch SFA camera, median filtered                                      | 168 |
| 8-1 | Proposed 4x4 CFA . . . . .  | 172 |
| A-1 | Kodak <sup>15</sup> image database . . . . .  | 178 |
| A-2 | McM <sup>16</sup> image database . . . . .  | 179 |
| A-3 | Finlyason <sup>17</sup> hyperspectral database images . . . . .                                     | 180 |
| A-4 | Cave <sup>18</sup> hyperspectral database images . . . . .  | 181 |
| A-5 | SCIEN <sup>19</sup> hyperspectral database without polarising filter images . . .                   | 182 |
| A-6 | TokyoTech 5-band multispectral database <sup>20</sup> images . . . . .                              | 182 |
| A-7 | TokyoTech 31-band hyperspectral database <sup>21</sup> . . . . .                                    | 183 |
| B-1 | LMMSE on Different CFAs for Kodak . . . . .   | 186 |
| B-2 | LMMSE on Different CFAs for McM . . . . .   | 186 |
| B-3 | Comparison of LMMSE with other Demosaicing algorithms for Kodak                                     | 187 |
| B-4 | Comparison of LMMSE with other Demosaicing algorithms for McM                                       | 188 |
| B-5 | Effect of Leave one out . . . . .   | 189 |
| B-6 | Effect of Leave one out . . . . .   | 189 |
| B-7 | FFT on different CFAs. Refer to table B.3 for ordering . . . . .                                    | 190 |
| C-1 | Check for Linearity in data from RAW file . . . . .   | 192 |
| C-2 | Photographing different wavelengths with a Nikon D300 camera for<br>exposure time 1/5000s . . . . . | 192 |
| C-3 | Estimated Filter response for the Nikon D300 . . . . .  | 193 |
| D-1 | Comparison of demosaicing for the JB Sensor simulated on SCIEN<br>database image . . . . .          | 197 |



# List of Tables

|     |  |     |
|-----|--|-----|
| 3.1 | Summary of the various methods to unfold and construct the demosaicing operator as per Linear Least Square method . . . . .  | 61  |
| 3.2 | Comparison of CFA for Kodak database . . . . .   | 72  |
| 3.3 | Comparison of CFA for McM database . . . . .   | 73  |
| 3.4 | Comparison of CFA for Kodak database with additive gaussian noise . . . . .  | 74  |
| 3.5 | Comparison of CFA for McM database with additive gaussian noise . . . . .  | 75  |
| 3.6 | Comparison between the best 4x4 with LMMSE on Kodak database and other methods . . . . .   | 76  |
| 3.7 | Comparison between the best 4x4 with LMMSE on McM database and other methods . . . . .   | 77  |
| 4.1 | LMMSE for Kodak database for multicolor CFAs, comparison with state of art . . . . .   | 96  |
| 4.2 | LMMSE for McM database for multicolor CFAs, comparison with state of art . . . . .   | 97  |
| 5.1 | Result of PSNR, SSIM, BLIIND-II and BRISQUE for reconstruction based on the benchmark binary tree demosaicing applied to the simulated images from SCIEN database. Column $S_i, i = 1..8$ indicates the spectral channels the camera delivers. . . . . | 113 |
| 5.2 | Result of PSNR, SSIM, BLIIND-II and BRISQUE for reconstruction based on the N-LMMSE demosaicing applied to the simulated images from SCIEN database. We used a LOO method. . . . .   | 114 |

|     |  |     |
|-----|--|-----|
| 5.3 | Result of reconstruction for the benchmark binary-tree applied to the real SFA images. . . . .   | 120 |
| 5.4 | Result of reconstruction for the N-LMMSE applied to the real SFA images. . . . .   | 121 |
| 6.1 | Result of varying $\lambda_\sigma$ for different SFA size. All values are averages across images of the Finlayson database, except $\mu_{Z(chart)}$ which is average across 24 colors in the Macbeth chart . . . . . | 144 |
| 6.2 | Result of varying $\lambda_\sigma$ for different SFA size. All values are averages across images of the Cave database. . . . .   | 145 |
| 6.3 | Result of optimized filters for different SFA size for the Finlayson database, neighborhood size 7. . . . .  | 150 |
| 6.4 | Result of $\mu_Y$ for demosaicing operator trained on $D65$ and used to demosaic SFA images rendered for different standard illuminants, for different SFA sizes and $\lambda_\mu = 3$ . . . . .                     | 151 |
| 6.5 | Result of varying $\lambda_\sigma$ for different SFA sizes for Visible+NIR filters. All values are averages across images of the SCIEN database. . . . .   | 154 |
| 7.1 | Performance of Neural Network, expressed as averages across images in Kodak database . . . . .   | 162 |
| 7.2 | Result of Average PSNR for Kodak database for our method compared with state of art . . . . .  | 162 |
| 7.3 | Result of Average PSNR across 5 channel and time for Monno5ch SFA on 5 channel multispectral Tokyo dataset . . . . .   | 165 |
| 7.4 | Result of Average PSNR across 5 channel and time for Monno5ch SFA with filter sensitivities as implemented simulated on Finlayson, Cave and TokyoTech 31 database . . . . .  | 169 |
| B.1 | Ordering of images B-1 B-2 . . . . .   | 185 |
| B.2 | Ordering of images B-3 B-4 . . . . .   | 185 |
| B.3 | Ordering of images B-7 . . . . .   | 190 |

## Table of notations

|                    |   |
|--------------------|---|
| $H$                | number of rows in an image  |
| $W$                | number of columns in an image   |
| $P$                | number of color channels in a color image   |
| $P_C$              | number of color channels in a multicolor image  |
| $P_s$              | number of spectral channels in a spectral image   |
| $P_\lambda$        | number of spectral channels in a reflectance image  |
| $X$                | CFA (Color Filter Array) / SFA (Spectral Filter Array) / RAW image  |
| $Y$                | Full resolution color image having $P$ color channels<br>Full resolution spectral image having $P_s$ spectral filter channels |
| $Z$                | Full resolution reflectance image having $P_\lambda$ hyperspectral channels   |
| $D$                | Demosaic operator matrix  |
| $h$                | number of rows in super-pixel   |
| $w$                | number of columns in super-pixel  |
| $x$                | unfolded $X$ into a matrix of vectors   |
| $y$                | unfolded $Y$ into a matrix of vectors   |
| $\hat{y}$          | estimated $y$ after demosaicing   |
| $n_h$              | neighborhood pixel size along height  |
| $n_w$              | neighborhood pixel size along width   |
| $n'_h$             | expanded neighborhood pixels along height for constant neighborhood   |
| $n'_w$             | expanded neighborhood pixels along width for constant neighborhood  |
| $x_1$              | unfolded $X$ with neighborhood  |
| $y_1$              | unfolded $Y$ with neighborhood  |
| $M$                | projection matrix which applies the mosaic on $y$ to get $x$  |
| $M_1$              | projection matrix which applies the mosaic on $y_1$ to get $x_1$  |
| $K$                | Number of vector elements used for learning   |
| $N_{DB}$           | number of images in an image database   |
| $E_{j=1..k}(N(j))$ | Expectation operation on a matrix $N(j)$ over the $k$ exemplars used to learn   |
| $R_y$              | Expectation of correlation matrix in $y$  |
| $R_{y_1}$          | Expectation of correlation matrix in $y_1$  |
| $S_1$              | matrix which removes the neighborhood in $y_1$ to get $y$   |
| $MSE$              | Mean Square Error   |
| $PSNR$             | Peak Signal to Noise Ratio  |
| $\mu$              | Average color PSNR over images in a database (e.g. Kodak, McM)  |
| $\mu R$            | Average PSNR of the red channel over images in a database   |
| $\mu G$            | Average PSNR of the green channel over images in a database   |
| $\mu B$            | Average PSNR of the blue channel over images in a database  |
| $\sigma$           | Variance of color PSNR over all images in a database  |
| $\sigma_{rgb}$     | Variance of PSNR per RGB channel, averaged for all images   |
| $\sigma_{5ch}$     | Variance of PSNR for the 5 channels of the Monno5ch SFA   |

|                                      |  |
|--------------------------------------|--|
| SSIM                                 | Structural similarity index  |
| $P_C$                                | number of possible colors in multicolor CFA                              |
| $C$                                  | color image for multicolor CFAs  |
| $c$                                  | unfolded $C$ image   |
| $c_1$                                | unfolded $C$ image with neighborhood                                     |
| $\alpha\beta\gamma$                  | mutiplying factor for the RGB  |
| $A$                                  | matrix composed of $[\alpha\beta\gamma]$                                 |
| $A_1$                                | matrix $A$ accounting for neighborhood                                   |
| $Tr$                                 | Trace of a matrix  |
| $\Delta E$                           | Euclidean color difference in Lab color space                            |
| $\lambda$                            | Wavelength   |
| $\rho_j$                             | integrated signal recorded by a sensor overlaid with filter $F_{P_s}$    |
| $z$                                  | unfolded reflectance image   |
| $z_1$                                | unfolded reflectance image with neighborhood                             |
| $R_{z1}$                             | Expectation of correlation matrix in $z_1$                               |
| $F$                                  | Matrix of $P_s$ spectral filters interpolated at $P_\lambda$ wavelengths |
| $f$                                  | unfolded $F$ for the super-pixel   |
| $f_1$                                | unfolded $F$ for the super-pixel and neighborhood                        |
| $L$                                  | Matrix of illuminant interpolated at $P_\lambda$ wavelengths             |
| $l$                                  | diagonalized $L$ accounting for the super-pixel                          |
| $l_1$                                | $l$ matrix accounting for neighborhood                                   |
| $T_X$                                | XYZ to sRGB transform  |
| $T_F$                                | Filter space to sRGB transform   |
| $Q$                                  | XYZ color matching function  |
| $I_{sRGBlinear}$                     | linear values of sRGB for $Y$ obtained from spectral domain              |
| $I_{sRGB}$                           | sRGB for $Y$ obtained from spectral domain                               |
| $G$                                  | Spectral response of silicon sensor                                      |
| $\lambda_\sigma$                     | the standard deviation of a gaussian filter                              |
| $\lambda_\mu$                        | the wavelength of peak sensitivity of gaussian filter                    |
| $A_m$                                | peak amplitude of the gaussian filter                                    |
| $RI$                                 | Radiance image   |
| $RI_{sRGB}$                          | sRGB image obtained by rendering $RI$ for a given $F$                    |
| $Y_{sRGB}$                           | sRGB image obtained by rendering spectral $Y$                            |
| $\hat{Y}_{sRGB}$                     | sRGB image obtained by rendering demosaiced spectral $\hat{Y}$           |
| $\mu_Z$                              | average PSNR in reflectance images for all images in a database          |
| $\mu_Z(chart)$                       | average PSNR in reflectance images for 24 macbeth chart                  |
| $\mu_Y$                              | average PSNR between demosaiced and full resolution SFA images           |
| $\mu_1$                              | average PSNR between $RI_{sRGB}$ and $Y_{sRGB}$                          |
| $\mu_2$                              | average PSNR between $Y_{sRGB}$ and $\hat{Y}_{sRGB}$                     |
| $\mu_3$                              | average PSNR between $RI_{sRGB}$ and $\hat{Y}_{sRGB}$                    |
| $SSIM_1, SSIM_2, SSIM_3$             | SSIM averages as per $\mu_{1,2,3}$ respectively                          |
| $\Delta E_1, \Delta E_2, \Delta E_3$ | $\Delta E$ averages as per $\mu_{1,2,3}$ respectively                    |
| $D_s$                                | demosaicing operator for recovering reflectance                          |
| $J$                                  | gradient of XYZ to Lab transform around a white point                    |
| $n$                                  | number of neurons in the hidden layer of Neural Network                  |

# Chapter 1

## Introduction

Photography has made immense gains in the last 24 years since the advent of the first consumer digital cameras. Beginning from a modest 0.3 MP (Megapixels) resolution of the Apple QuickTake cameras<sup>22</sup> (built by Kodak) to today's 100+ MP medium format cameras<sup>23</sup>. One of the reasons that the camera is ubiquitous today is because of Color Filter Array (CFA) sampling that allow a single shot acquisition of full color of a scene. Figure 1-1 shows an example of the acquisition of a color image with a single sensor covered with a CFA. Only a single sensor is needed because each pixel value correspond to a single color sensitivity obtained from passing through the matrice of three color filters. So each pixel has a single color but several pixels have potentially every colors like a mosaic. It is easy to imagine the compromise between space and color such a spatio-spectral sub-sampling acquisition system implies. If there is a change in red level at a position not sampled by the sensor, the location of the red change would be imprecise. In reverse a contour is sampled only for one color channel and the two others should follow the same slope otherwise false color could appear. Hopefully, natural images are spatially and chromatically correlated which allow the learning of a demosaicing solution.

Before the introduction of CFAs, digital cameras required three CCD sensors. Each one was fully covered by a filter, for instance Red, Green and Blue, such that it simultaneously captured all the three colors. It is easy to understand that such systems are expensive, heavy and cumbersome. Having a camera in your mobile phone



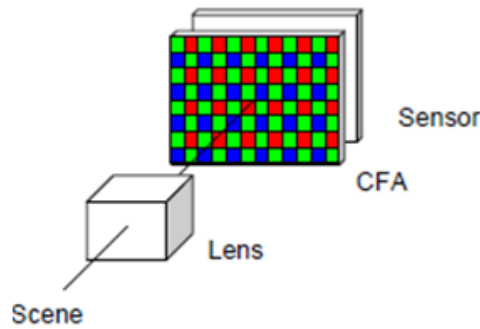


Figure 1-1: A single sensor digital camera uses a color mosaic to acquire a color image. In this schema only a single sensor is required which is covered with a mosaic of three different color filters to allow color sensitivity. Shown is the Bayer CFA<sup>3</sup>.

wouldn't have been possible without the invention of a single sensor covered by a CFA. But this way of acquiring color image has its drawbacks. From the sub-sampled images captured by a camera with CFA, one needs to estimate the missing colors and this step is known as demosaicing, the problem we tackle in this thesis. Technically, demosaicing is a difficult problem because it is a spatio-spectral subsampling which in turn challenges the digital imaging market. First, consumers want images without false colors and artifacts which are a by product of the demosaicing algorithms. Secondly, because of the ubiquity of the camera, we want the demosaicing algorithm to be as power efficient as possible. Another complementary advancement has been our ability to display a wide gamut of colors on screen size ranging from huge projectors to a wearable watch. So, we are not only looking for less color noise but also more accurate colors. The main goal of this thesis is to propose a demosaicing algorithm which has good performance in terms of accurate color reproduction, less color noise and is less computationally expensive.

Color as a sense is so natural to a human being, we never consciously reflect on the fact that color vision of another fellow being could be different from ours. A color-blind person is not consciously self aware that he/she is missing something. It is a realization which comes from specially designed tests. Color means different things to different people. To a physicist, color is defined by the wavelength of electromagnetic

radiation. To a computer graphics designer, color is the value of Red, Green and Blue pixel value. To an artist, color has its language, it could be warm or cold, it could have emotions and other attributes. It is what philosophers call qualia; the subjective feeling of color. To a psycho-physicist, color is the perception of light in the human brain. Color is actually sum of all of this and more. Color is defined by the physics of an object in a particular environment and its perception by the Human Visual System (HVS). When we talk about color/spectral we are at an intersection of three domains, namely physics of light, human visual physiology and psychology of perception. Also this perception varies between individual, my red is not same as your red. This inter individual difference was classically demonstrated by the image of the blue-black or white-golden dress<sup>24</sup> which people reported seeing in different combinations of colors. For purposes of this thesis we don't delve into the psychological aspects and limit ourselves to physics and two sensory systems, human and a digital camera.

Color vision is trichromatic in nature. In the HVS, the LMS cones, so called as they are sensitive to the long (L), medium (M) and short (S) wavelength of the visible light spectrum. They are overlaid in a random arrangement to form the inner surface of retina. Color capture for digital cameras follows trichromacy, but this is not enough. A complex regulation of signals is needed to improve the image issue from a camera. Regulation involves white balancing, gamut mapping and chromatic adaptation which are fine tuned together.

In the state of art, demosaicing for Bayer CFA is widely studied and several approaches to estimate the color image have been proposed. However, due to its CFA being a regular/periodic pattern gives rise to false colors/artifacts/moire specially in high frequency regions of the image. More complicated algorithms (e.g. compressive sensing, edge aware demosaicing, etc.) have been proposed in literature which avoid these issues however they are computationally more expensive, therefore not suitable for real-time applications. Typically camera manufacturers employ an Optical Low Pass (OLP) filter on the mosaic to blur the projected image to avoid these false colors. However, due to race towards increasing Megapixels count and to gain any meaningful image resolution they are forced to remove these filters making it more relevant today

to find a real-time and powerful demosaicing algorithm. Another approach to avoid the false colors/artifacts is to rearrange the filters on the CFA mosaic. Fujifilm in order to improve image quality proposed its XTRANS sensor<sup>25</sup> which is supposed to avoid the problem of moire and therefore gain resolution. Several authors have also proposed optimized CFAs and algorithms tuned to their particular arrangement of color filters.

The new challenge in digital color imaging is to gain more light sensitivity to allow night vision capabilities and to capture more dynamic range. This is done by adding a White or a Near-Infrared (NIR) pixel to the CFA mosaic. Due to difference in sensitivities between the White/NIR and the color filters, this makes demosaicing for such CFAs more complicated. Also recent technological advances have allowed us to go beyond three color filters. One can implant any number of spectral filters and customize their frequency response according to application and lay them in form of a mosaic to form Spectral Filter Arrays (SFAs). In this thesis, if the spectral sensitivity of the color filters (transmittance) is known we refer to their mosaics as SFAs, otherwise as CFAs. SFA is a more generic term. Demosaicing for SFAs is a new challenge and a subject of active research. Also these SFAs could be used to predict the spectral signature of the scene from an image capture. It opens new applications in field of computer vision, robotics, autonomous vehicles and agriculture (chlorophyll detection or fruit ripeness detection). In the case of RGB CFAs, there were only three kind of color filters with high inter channel correlation and small distance between two neighboring pixels of same color filter. For the SFAs the number of spectral filters are limited only by manufacturing processes, already mosaic having 32 different filters are in production. The spatio-spectral sampling in this case is very large in scope and requires defining a general framework for demosaicing.

Physics gives us a spectral model to describe the image formation as product of illuminant spectral function with that of scene reflectance as modulated by the spectral response of the color/spectral filters. There is no such model for spatio-spectral sampling (optics, lens, sampling area, etc. ) by the mosaics.

In state of art demosaicing involves interpolation based on neighboring pixels to

guess missing pixels for RGB CFAs by implicitly exploiting spatio-spectral correlation. For SFAs, this is complicated due to less inter channel correlation and increased spatial distance between same color pixels. Therefore, one needs to better understand spatio-spectral correlation between color pixels to infer a spatio-spectral model of image formation. We do so by modeling spatio-spectral statistics of vectors sampled on a database of true color RGB or hyperspectral image databases. On these vectors, we learn the transformation between correlation in full resolution color images and correlation of spatio-spectral sub-sampled image acquired through a CFA. Databases are representative of natural images and by having sufficient number of images one ensures generality of the learning. Natural images have very wide spatial correlation, the mosaicking process tends to limit the correlation within the confines of the shift-invariant basis block. We will use a large neighborhood around each pixel in the mosaic block, which allows to extend this correlation and therefore better estimate the demosaicing operator. We show by experiment that it outperforms other approaches.

We consider demosaicing as an inverse problem. Inverse problems are those in which we try to estimate the cause by observing its effect. As the mosaicking operation is a linear process, it is natural to consider a linear inverse solution. For this we used the linear least square approach to solve the problem of minimizing the mean square error between the cause and an estimation of cause. This approach utilizes the correlation between the cause, in our case, natural images, where we know correlation is high. Therefore it is well suited to this problem. So, we provide a linear solution by training on an image database and using a large neighborhood to ensure that our linear system is over determined.

In this thesis we propose to develop a generic demosaicing algorithm for both color and spectral filter arrays based on statistics of natural images. By generic, we imply that the algorithm should be applicable for any random or periodic arrangement of color/spectral filters. One of the conditions is that the algorithm should be fast, therefore applicable for real-time applications which is a requisite for embedded systems. We know false colors/moire in the case of Bayer CFA are due to regular nature of the CFA. While in the case of human vision where the mosaic is random<sup>1</sup> we

never experience these false colors. Therefore, we hypothesize that **random CFAs are better than regular ones**. The idea being that color noise in a regular pattern is spatially structured therefore visually more distinct whereas a random mosaic will tend to randomize the color noise therefore better hiding it in the real signal. In the literature to overcome the problem of false colors, solutions based on edge directed demosaicing, local-polynomial approximations or compressive sensing<sup>26,27</sup> are provided, but they are non-linear in nature. However, non-linear solutions tend to be computationally more expensive whereas our requirement is for embedded systems. Therefore, we wish to remain within confines of a linear solution. Therefore, we hypothesize that **by considering a neighborhood of pixels around pixel position being interpolated and by learning statistics on an image database for random CFA patterns using a linear operator, we can achieve performance metrics similar to existing non-linear solutions**.

The solution we consider is block-shift invariant (a block corresponds to the basis pattern in the CFA), the same demosaicing operator can be applied to each basis pattern along with its neighborhood pixels. In this thesis we will discuss two different strategies, sliding vs constant for populating the neighborhood around a pixel in a block. We show that by enlarging the neighborhood size for the constant, compared to the sliding ensures equivalence in performance. Then removing the redundant neighboring pixels in constant neighborhood, reduces the memory footprint and gains in execution speed.

We prove both our hypothesis by simulation on a database. Doing a systematic evaluation of all RGB filters arrangement on a 2x2, 3x3 and 4x4 basis pattern and demonstrating that the best performing CFAs were random and not periodic and had a quasi-equal distribution of RGB filters. Further, we present a framework for optimizing 'multicolor CFAs', CFAs having colors filters which are linear combination of RGB filters. We demonstrate by simulation that the RGB and multicolor CFAs so proposed outperform the state of art and are computationally more efficient. We then demonstrate our algorithm on real SFA sensor having both visible and NIR filters and show that our algorithm gives sharper result with less color artifacts. We then also

propose optimized SFAs based on different evaluation metrics and develop a method to recover scene reflectance/radiance from RAW images.

Finally, we challenge our solution by considering a non-linear solution, using Neural Networking to solve the inverse problem. Artificial neural networks (NN) have been successfully used for solving several fitting and pattern matching problems. They are inspired by biological neurons in the sense that each neuron outputs a non-linear function of the sum of its inputs modulated by weights and a bias. In context of demosaicing, a deep learning based neural network having 15 layers<sup>28</sup> has been demonstrated to have good performance. Again we hypothesize that **by considering a large neighborhood around the pixels being interpolated it maybe possible to achieve similar performance using a simple neural network architecture (less number of layers, neurons, etc.), thereby reducing computational complexity.**

## 1.1 Thesis Outline

In Chapter 2 we provide the background to the problem. We define the vocabulary required to follow this thesis. We define the terminology associated with measurement of light/color. We introduce the digital camera as a capturing instrument of color and present the image formation pipeline. We briefly introduce the Human visual system and Colorimetry standards. Finally, we introduce the problem of demosaicing and the concept of linear inverse.

In Chapter 3, we review the state of art for demosaicing the Bayer CFA. We present our Linear Minimum Mean Square Error (LMMSE) algorithm for demosaicing random RGB (Red, Green, Blue) CFAs. We show two possibilities of considering the neighborhood pixels: 'sliding' and 'constant'. We perform a systematic evaluation of all possible arrangement of color filters on a 2x2, 3x3 and 4x4 CFA layout and provide the best performing CFAs. We also evaluate our algorithm on optimized CFAs proposed in literature. Further, we compared our algorithm with state of art demosaicing algorithms. Finally, we provide result for demosaicing for totally random

CFAs (at the size of image) and compare with compressive sensing approach.

In Chapter 4, we provide a method of optimizing spatio-spectrally the RGB filters of CFAs, limiting ourselves to considering filters as linear combination of RGB filters. In state of art, several such CFAs have been proposed (based on optimizing separation of luminance and chrominance in fourier domain). So this serves to compare our approach with such CFAs and shows the strength of our linear approach.

In Chapter 5, we extend our algorithm to the spectral domain. We present the image formation model for a camera with mosaic using spectral data. We present a way to train our algorithm on a hyperspectral database which can be used to construct the demosaicing operator for any Spectral Filter Array. We present the results on a prototype camera having both Visible and Near Infrared Filters. We present results on both simulated images from hyperspectral image database and RAW images coming from real camera. We present the use of No-Reference metrics for evaluating demosaiced RAW images. We compared our method with the Miao's Binary Tree algorithm<sup>29</sup>.

In Chapter 6, we provide a methodology to optimize filter spectral sensitivity based on different applications, like demosaicing, spectral reflectance recovery, selection of filters for color-wheel based camera. We present results by considering gaussian filters as approximation of transmission given Faber-Perot process.

In Chapter 7, we provide a dual layer neural network architecture for demosaicing using neighborhood. We compare with the state of art for demosaicing CFAs. For SFAs we consider the 5 channel<sup>21</sup> prototype camera from Tokyotech and demonstrate that we can gain resolution and reduce color noise at the cost of some artifacts using the neural network or LMMSE approach compared to Monno et al.'s method using Weighted Guided Filters.

We conclude the thesis in Chapter 8 by summarizing our contributions and give a perspective on limitations and scope for further improvement.

# Chapter 2

## Light and Color

Color vocabulary is very specific and not very clear as its definition changes across disciplines like physics, psychophysics, computer graphics, etc. The purpose of this chapter is to first introduce the vocabulary required to specify the problem of color mosaicing and its rendering on displays for human observers. Light is electromagnetic radiation and in context of human vision and color imaging, we are interested in part of the spectrum where the human vision is sensitive. The term light is also inclusive of the ultraviolet and the infrared spectrum. One of the seminal works on the understanding of relationship between light and colors came from Isaac Newton in the 17th century when he demonstrated that passing white sunlight through a prism splits it into a rainbow of colors. Thus, associating wavelength of monochromatic light with a certain color. However, this is not strictly true as physics is not color. One can have different light sources having a similar visual color appearance. For example a halogen bulb, a fluorescent lamp and an LED might produce the same 'warm white' we are used to. However, they have different spectral properties. Therefore, it is important to distinguish between the physics of light and the perception of color. Colorimetry has defined metrics from physics which can describe a light source and also define a color vision space describing how human observer transforms these physical metrics.



## 2.1 Physics of light

A light source is an emitter of electromagnetic radiation. One can generally measure its Spectral Power Distribution (SPD), which describes the power per unit area per unit wavelength of an illuminant, the spectral irradiance. The same when received on a unit projected area is the spectral radiance. One can have different light sources from the sun to the night sky, incandescent, fluorescent or LEDs for visible spectra. Blacklights can be sources for ultraviolet and sauna lamps can be sources for infra-red radiation. International Commission on Illumination (CIE) is an organization which has defined standards for different light sources. For example *D65* describes the average midday light in West/North Europe, hence known as a daylight illuminant. Similarly, *A* describes tungsten light source, *Fx* series describes the fluorescent light sources. Any light source can be measured using device known as spectroradiometers. These devices usually employ a diffraction grating which disperses the incident spectrum linearly which is then measured using a sensor like CCD array, refer to spectroradiometer in Figure 2-1. Figure 2-2 shows the spectral radiance measurement for four different light sources measured using Konica CS2000 spectroradiometer.

When light emitted from any light sources falls on any object, it is either reflected, absorbed or transmitted. These again can be represented using spectral data. The ratio of reflected light to the the incident light is known as reflectance. Reflectance when multiplied with the light irradiance give us the spectral radiance of the object which is then observed/recorded by human or a camera system. Using a spectroradiometer gives us the spectra for radiance of a single point. Using scanning technique it is possible to record the spectra of large number of pixels. Such a system is called a hyperspectral camera and outputs a cube with two spatial dimensions and a third dimension in wavelength. When accounted for illuminant one can have a reflectance cube for an object. One can use these reflectance images to render images for any spectral filter response and illuminant combination. For example for a digital camera having three color filters, Red, Green and Blue, see Figure 2-1. One needs to multiply scene reflectance with illuminant with the red filter's and silicon's combined spectral

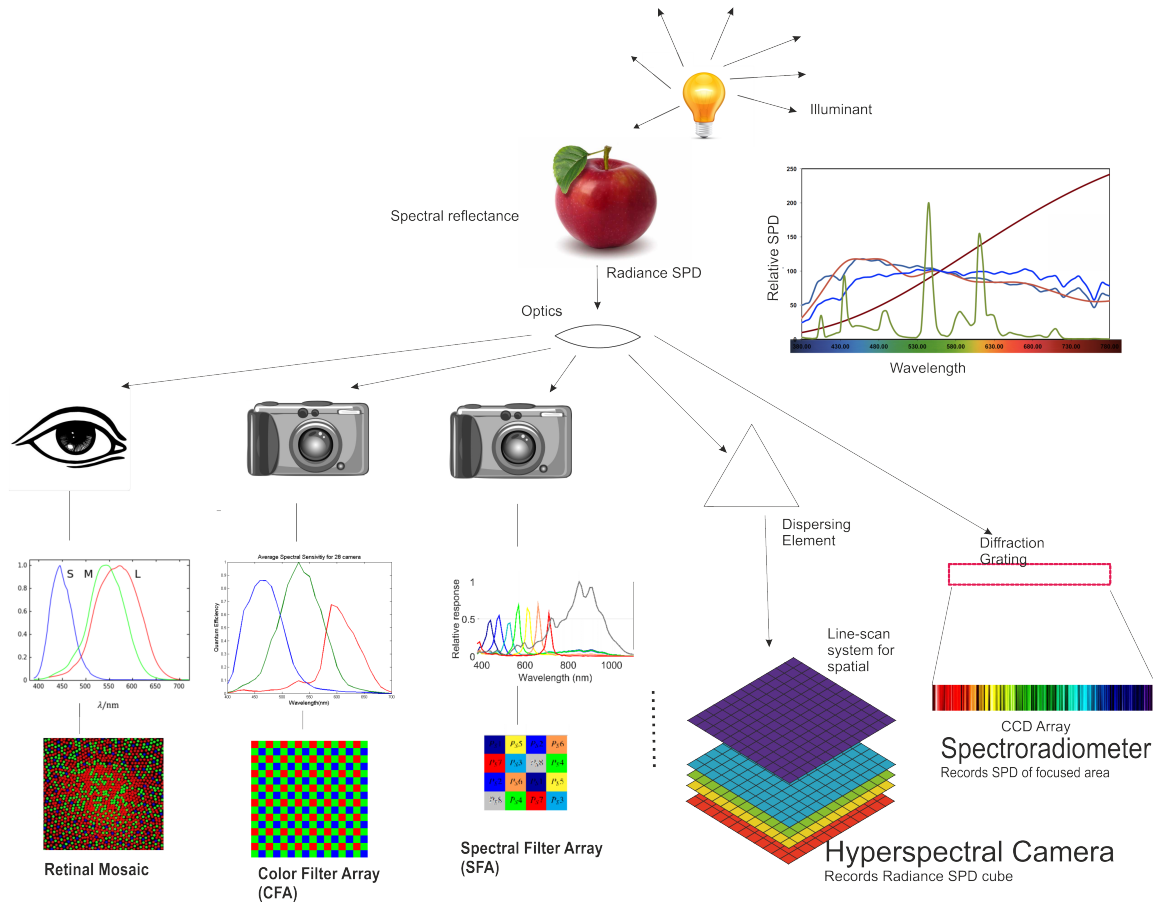


Figure 2-1: From Physics to Vision and Camera capture

response to obtain the pixel values for Red channel. Similarly for the other colors filters to obtain the green and blue pixel values.

When light falls on an object, there is also possibility of phosphorescence and fluorescence due to object properties, which is absorption of photon and emission of another at a different wavelength which we don't account for in this thesis.

## 2.2 Human Visual System

The human visual system (HVS) is composed of the eye, optical nerve and the processing centers in the brain. The eye is similar to a camera system, wherein we have a lens to focus, iris and pupil to control aperture and retina which is the sensory element. The difference between HVS and sensor is that the HVS is dynamic and

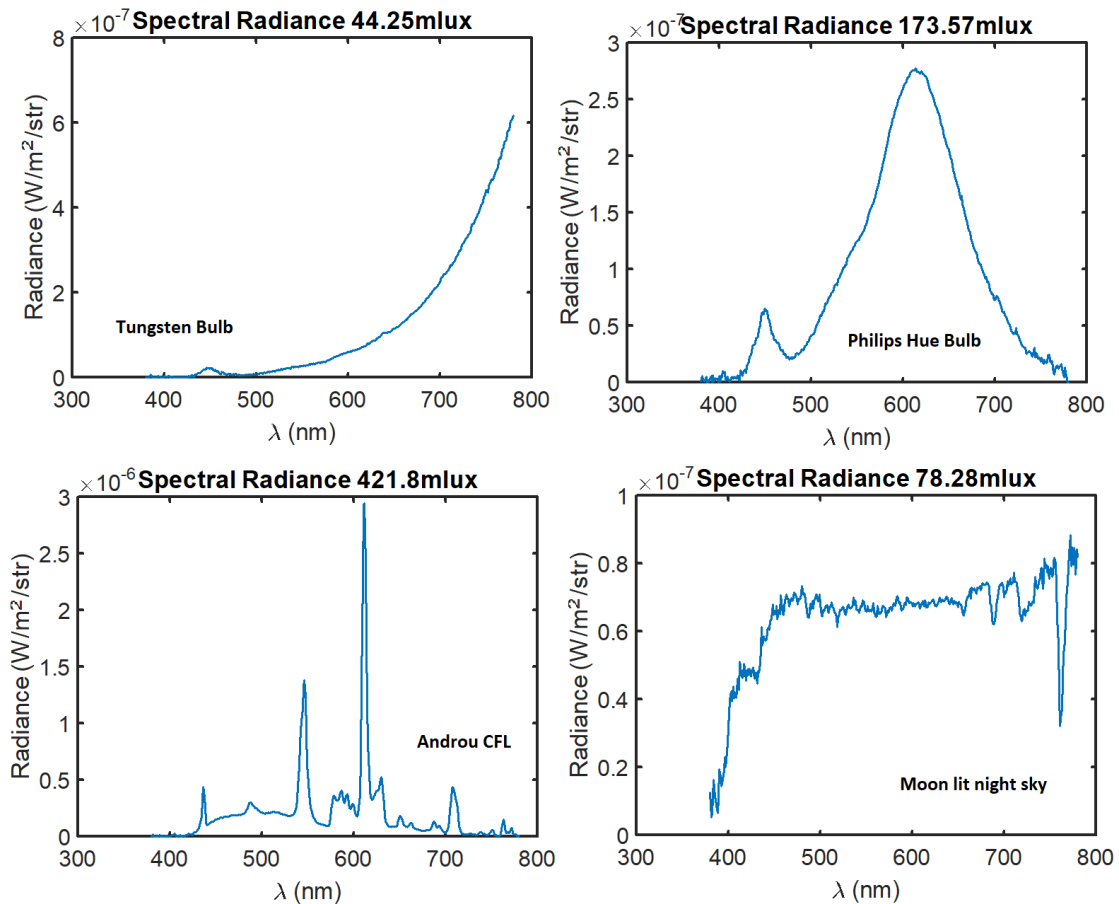


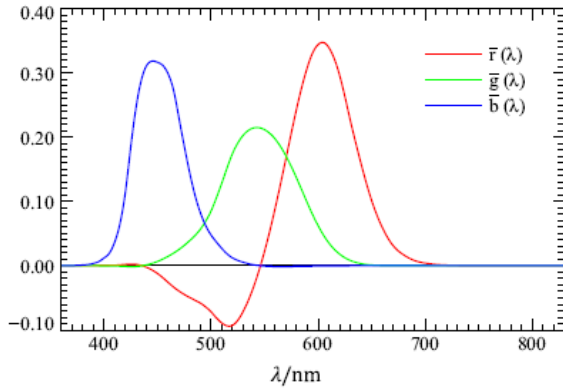
Figure 2-2: Spectral Radiance measurements using Konica CS2000 for different light sources

can adjust its sensitivity to the incoming light. The retina has two kinds of photoreceptors, cones and rods which absorb photons and emit a neural signal. Rods are more sensitive to light than cones, they are sensitive to low light conditions known as scotopic vision as they get saturated in normal lighting conditions. Rod vision or night vision is monochromatic. Cones are sensitive to normal illumination conditions known as photopic vision. There are three kinds of cones having different spectral sensitivities, known as LMS cones, because their spectral sensitivities peak in the long (L), medium (M) and short (S) wavelength of the visible spectrum which correspond roughly to red, green and blue light respectively. Figure 2-3-d, shows the normalized relative spectral sensitivities of the cones. There are about 120 million rods and 8 million cones in the retina which could be understood as the MP (Megapixel)

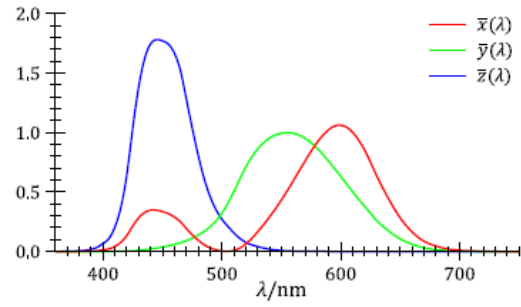
count of the HVS. Photoreceptors are connected to several neural cells (like horizontal, bipolar, amacrine and ganglion). **Each neural cell like the horizontal is connected to several photoreceptors, so there is an integration of signal from neighboring photoreceptors.** Moreover the three types of cones are laid in a random mosaic to form the inner surface of the retina. Another thing to note with the HVS is that not only the mosaic is random but also that its arrangement varies from individual to individual<sup>1,30</sup>. See Figure 2-4 which shows the random mosaics for ten different individuals. The human brain performs the task of estimating the full color image from these random mosaics. It is easy to appreciate the complex color processing that the human brain performs that despite these large difference in anatomy we all seem to agree well in our perception of colors and we are unaware of any artifacts/anomalies in our perception due to this mosaicking.

CIE has defined several standards related to human vision. CIE in 1924 defined the luminous efficiency function  $V(\lambda)$  to define the average eye spectral sensitivity. It describes the relative sensitivity across wavelength. One can easily infer from this that humans are more sensitive to green monochromatic light compared to red or blue light of same power. In 1931, CIE defined the Color Matching Functions ( $\bar{r}(\lambda), \bar{g}(\lambda), \bar{b}(\lambda)$ ) based on tristimulus colorimetry, mixing monochromatic primaries Red, Green and Blue light in different ratio can produce all possible test colors (Figure 2-3-a). We see from the figure that  $\bar{r}$  has a negative component, meaning that it was added to the test side. So a linear transform was found which made the RGB color matching functions, positive therefore transforming it into  $(\bar{x}(\lambda), \bar{y}(\lambda), \bar{z}(\lambda))$  space (Figure 2-3-b). This also ensure that  $\bar{y}(\lambda)$  is actually same as  $V(\lambda)$ .

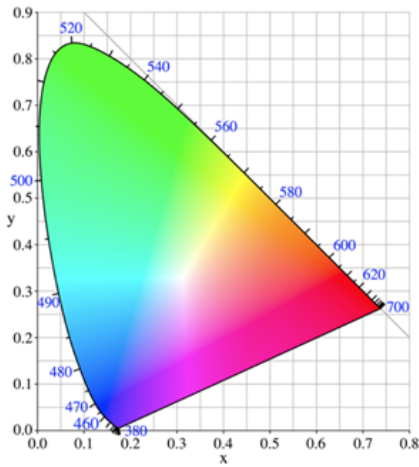
Doing an integral of the product of spectral radiance with  $\bar{x}$ , one can find its corresponding  $X$  value. Similarly for  $Y$  and  $Z$ . Giving us the values of the given illuminant in the XYZ color space. Two different illuminants having different spectra, however if they have same XYZ values will be perceptually same. One can normalize the individual X,Y,Z tristimulus values by  $X + Y + Z$  to give the chromaticity values  $xyz$ . This helps in visualizing on a 2D plane. Figure 2-3-c shows the xy Chromaticity Diagram for a standard observer. This is known as the gamut of human observer.



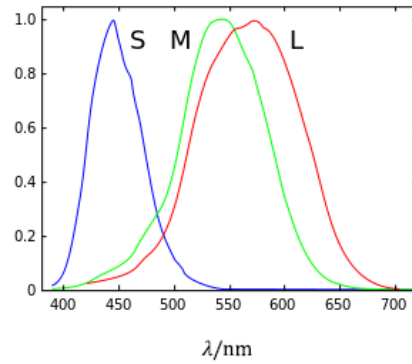
(a) Color RGB Matching Functions



(b) Color XYZ Matching Functions



(c) xy Chromaticity space



(d) Normalized Cone spectral Sensitivity

Figure 2-3: Color Matching Functions and LMS response

Any radiance of a visible light can be plotted as a point in this gamut. If one was to plot all such (x,y) coordinates for a display by varying its digital value for three phosphors/leds one would obtain a triangle (or polygon if more than three phosphors are used) within this space which is called the color gamut of that particular display. Similarly standard colors spaces like sRGB or Adobe RGB are defined which encompass a triangular region within this gamut. It represents all possible colors within a particular color space.

The XYZ space is not perceptually uniform, small changes in its values may result in large perceived changes. Therefore CIE developed the Lab color space by applying a nonlinear transform on XYZ, modeling for non-linearity and chromatic adaptation,

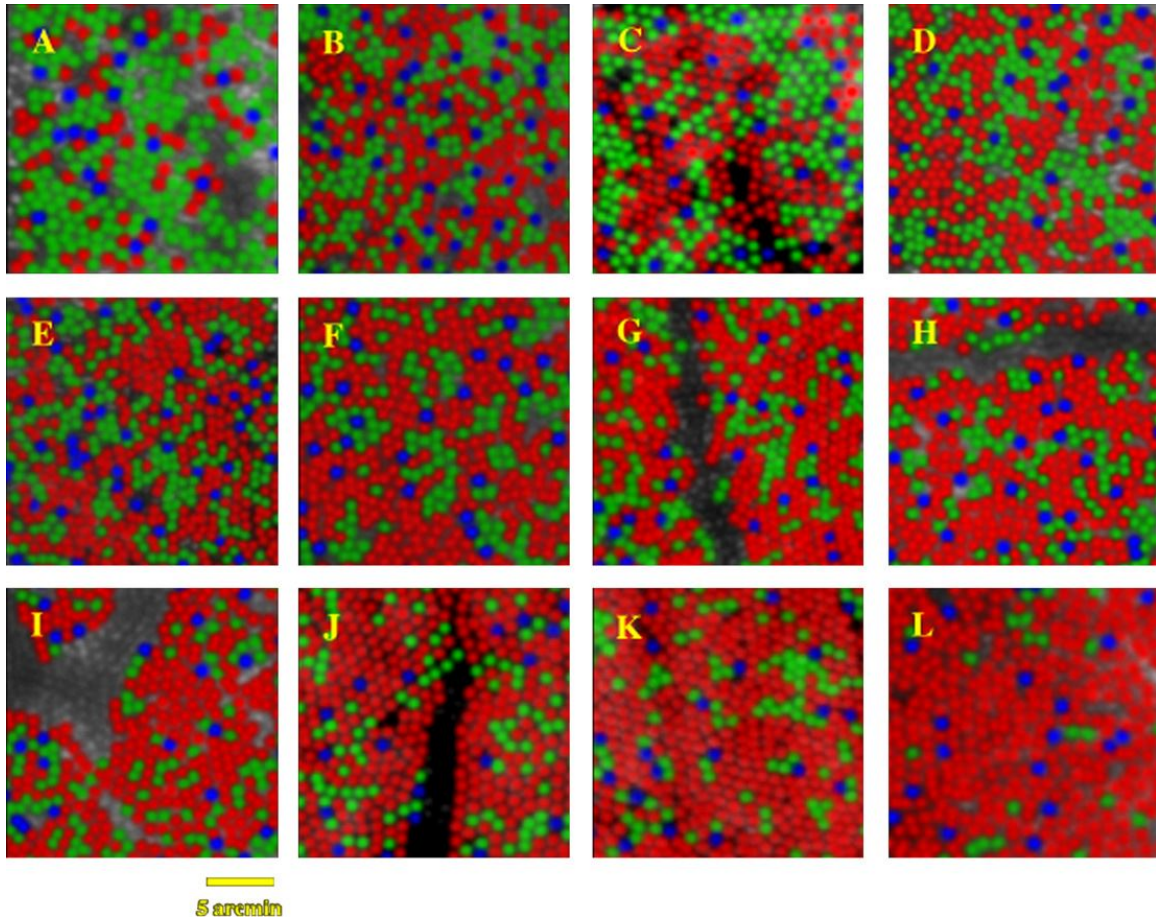


Figure 2-4: Images of cone mosaics of 10 subjects with normal color vision using adaptive optics imaging and retinal densitometry.<sup>1,30</sup> Observe differences in cone proportion and arrangement

and finally normalizing with a white point. What it allows is that for two colors, difference in Lab values is supposed to be equivalent to the perceived color difference. Therefore this color space allows us to define a metric  $\Delta E$  to measure color differences as the usual Euclidean distance.

## 2.3 Digital Camera

A digital camera consists of the optical system which projects scene radiance onto a sensor. Aperture can be opened/closed to control the amount of light passing through. Shutter control regulates the amount of time light falls on the sensor. The sensor may be covered with micro lens to better focus the light on the pixels. Sensor

is composed of a grid of pixel-elements which convert photons to electrons. Silicon is sensitive to both the visible and near-infrared (NIR) spectrum of the electromagnetic radiation. Commercial cameras generally employ a hot-mirror which is an IR cut filter to block the NIR components as it tends to blur the images. NIR has different focusing distance compared to visible radiation which leads to blurring. Also, objects have different reflectance properties to NIR for instance water absorbs NIR, which can create images different from what we perceive. However, for application involving night-vision it may be interesting to omit this filter as it allows greater sensitivity. Further analog signal is converted to digital by using an Analog to Digital Converter (ADC) which can be then read out as image RAW data.

From the color matching experiments, Figure 2-3 we see that we need information for three color primaries to produce all possible colors. For this experiment monochromatic light of 700nm, 546.1nm and 435.8nm were chosen which correspond to Red, Green and Blue colors. Similarly, red, green and blue filters are employed in digital cameras for color capture.

### 2.3.1 Color Filter Arrays

The most straightforward solution for color capture would be to use a three sensor system each overlaid with a single color filter, coupled with a beam splitter which directs the focused light to each sensor simultaneously.

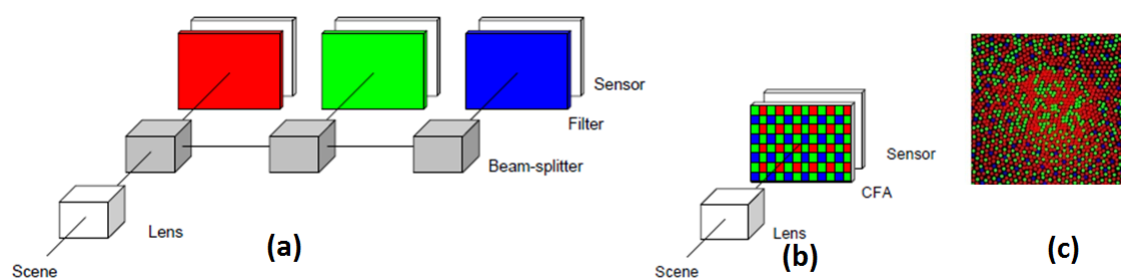


Figure 2-5: (a) Three sensor system, (b) Bayer CFA, (c) Random Human retina mosaic<sup>1-3</sup>

Figure 2-5-a shows an example of such a system. As can be clearly seen, such a system would be heavy, cumbersome and expensive due to use of three sensors.

Studio video cameras and earlier camcorders did employ such a mechanism having three CCD or three CMOS sensors. A simpler optical system is to overlay a mosaic of Color Filter Array (CFA) on a single sensor, for example, Figure 2-5-b shows Bayer CFA<sup>3</sup> and then to recover the missing colors from sampled ones, using an algorithmic approach. This process is called **demosaicing**. Such an approach could be compared to one deployed by the Human Visual System (HVS), wherein a random mosaic of cones form the inner surface of the retina.

Another approach for capturing three colors at same position was proposed by Foveon<sup>4</sup> wherein it exploits the property of silicon to absorb different wavelengths of light at different depth, see Figure 2-6. However this approach has had limited market acceptance due to limited spatial resolution and higher noise in the blue pixels, compared to the CFA approach and therefore remains a niche product to date.

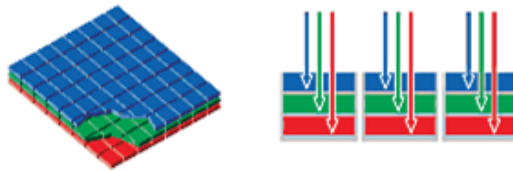


Figure 2-6: Foveon Sensor<sup>4</sup>

The Bayer CFA as proposed by Kodak remains the most widely used and therefore has been studied in detail and several demosaicing algorithms have been proposed. However, it remains susceptible to color noise and moiré.

In this thesis, we will study demosaicing for CFAs like Bayer, other regular mosaics and random RGB mosaics. We also discuss demosaicing for CFAs having colors filters other than red, green and blue.

### 2.3.2 Spectral Filter Arrays

Recent technological advances have allowed manufacturers to go beyond three color filters by selecting the spectral response of filters. We term them Spectral Filter Arrays (SFA) in a more generic term. We use the term SFA instead of CFA when we have data for the filters spectral transmittance available. Several manufacturers have



proposed mosaics with panchromatic filter, a white filter to improve low light sensitivity. Silicon which is the material used for building photodiodes is sensitive to both visible and Near-Infrared (NIR) light. Digital cameras usually employ a hot-mirror which basically cut-offs the NIR radiation from reaching the sensor, this was done to keep recorded images sharper as NIR tends to focus at different distance compared to visible radiation. Complex optics are used to compensate for that and by removing the hot-mirror one can also have NIR pixels on the mosaic. Figure 2-7 shows several commercial/academic propositions. For RGB CFAs, generally dyes were used as constituent of the colors and their spectral response tends to be broadband with high correlation between filters. Today using Faber-Perot type of interferometers gives us greater control on designing the spectral response of the filters. They primarily use thin film deposition and control the thickness of deposit which decides the transmission characteristics of filters. So it is possible to have narrow-band filters.

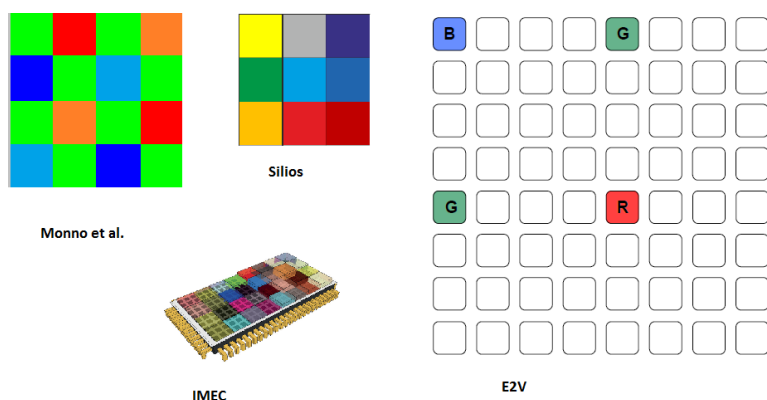


Figure 2-7: Spectral Filter Arrays. Monno et al<sup>21</sup> has 5 color filters. E2V<sup>31</sup>'s proposed sparse sensor, Silios<sup>32</sup> offers SFAs with 9 or 16 narrow band spectral filters. IMEC<sup>33</sup> offers SFAs with upto 32 filters.

SFAs potentially open up new applications for the digital cameras. SFAs proposed by Monno et al.<sup>21</sup> have five color filters, orange and cyan in addition to traditional RGB. Authors claim that they allow more accurate color reproduction. SFAs having white or NIR filter are designed for night-vision capabilities. Having several color filter bands like those proposed by IMEC or Silios<sup>33,32</sup>, allow us to capture a spectral

signature of the scene and thus can be used for applications like chlorophyll detection, fruit ripening, computer vision or object recognition. Demosaicing for SFAs is a new challenge as it is a recent development and a mere extension of demosaicing algorithms developed for RGB CFAs is not possible due to differences in spectral correlation between filters and more complex spatial arrangement of filters.

## 2.4 Analogy between Camera and the Human Visual System

Both the Digital Camera and the Human Visual System work on the same principle of trichromacy, modulating the scene radiance by the spectral sensitivities of the color filter/cones respectively. We saw that the spectral sensitivities for the LMS cones (Figure 2-1) are very different from the RGB color filters. In case of the HVS the L and M cone sensitivities are very similar and overlapping whereas for the digital camera although the RGB filters are overlapping (inter channel correlation allows better demosaicing as missing colors could be guessed from neighboring pixels), they still are distinguished. Further, the LMS cones are arranged in a random mosaic whereas for the RGB camera, the mosaic of choice is the Bayer CFA which is a regular/periodic pattern. Demosaicing for this regular pattern using simpler algorithms, gives false colors, artifacts and aliasing in regions of high frequency. However, in the case of the HVS we are not aware of any such artifacts. A random pattern is less susceptible to perceived noise as even though an equal amount of color noise might be present it is better hidden within the mosaic's random structure. In the case of Bayer it is due to the regularity of the mosaic's pattern that the color noise also gets a regular structure therefore becoming visually disturbing. In the next chapter we demonstrate this by comparison of demosaicing on Bayer, other regular CFAs, non periodic CFAs and totally random CFAs. We present results both visually and in terms of performance metrics.

The HVS employs two types of photoreceptors, cones and rods. Rods are active

in low-light conditions and color insensitive. This is somewhat analogous to SFAs being developed today which incorporate both color filters and a panchromatic pixel which is similar to the rods as it has a higher sensitivity. Sparse sensors (E2V, Figure 2-7) has the goal to imitate the proportion of the cones and rods in the retina with White and RGB pixels.

In term of photoreceptors counts, the retina has approximately 108 million while today sensors are approaching 50 million count. The retina has only 8 million cones which are only present in the center of the fovea. In the case of the retina there is a lot of integration of signals from neighboring cones happening at the layer of horizontal and bipolar cells. By analogy, demosaicing algorithms employ inter channel correlation between neighboring pixels to improve the estimated image. In this thesis, we would consider large neighborhood around each pixel and present formally different strategies useful to populate this neighborhood. An important distinction to remember is that a camera is basically a recording device while the HVS is for perception. The eye doesn't record, it sees.

## 2.5 Image processing pipeline for a camera

Having the camera producing an image from the sensor is a complex procedure. Figure 2-8 shows the image processing pipeline for a standard digital camera<sup>5</sup> which describes the steps performed in processing the RAW data recorded by the sensor and transforming it into a color image which can be displayed or printed. This chain is also referred to as RAW conversion. A digital camera's ADC outputs the RAW image which first needs to be 'pre-processed'. Pre-processing here involves correcting for differences in photo diodes sensitivities, dark current shot-noise removal, hot-pixel correction for pixels permanently stuck have to be discounted. Next would be white-balancing, although certain workflows do this step after demosaicing. Human vision has property of color constancy, perceived color of an object is consistent despite changes in illumination. A white paper under incandescent or daylight always appear

white. These changes in illumination are not automatically accounted for in a digital camera. They need to ensure that the white patch is white. Next step would be the demosaicing, which we discuss in more detail in the next section. The full resolution color image now obtained, adjusted such that the white is white, is in the camera filter space, in the color space as defined by the color/spectral filters. One needs to do a color transform from the camera to the unrendered space like the XYZ space. This rendering allows further to map it to other color spaces like sRGB, Adobe RGB or color space for printers. Finally, gamma correction is applied to compensate for non-linear perception of brightness by human beings. Next, we would have post processing like sharpening, noise removal, chromatic aberration correction, levels correction or deconvolution. Finally, the image can be displayed and compressed for storage purpose. In this thesis, we are primarily concerned with the demosaicing aspect and in context of SFAs, we also look at camera filter space to standard color space conversion as it helps to visualize the output image and test metrics.

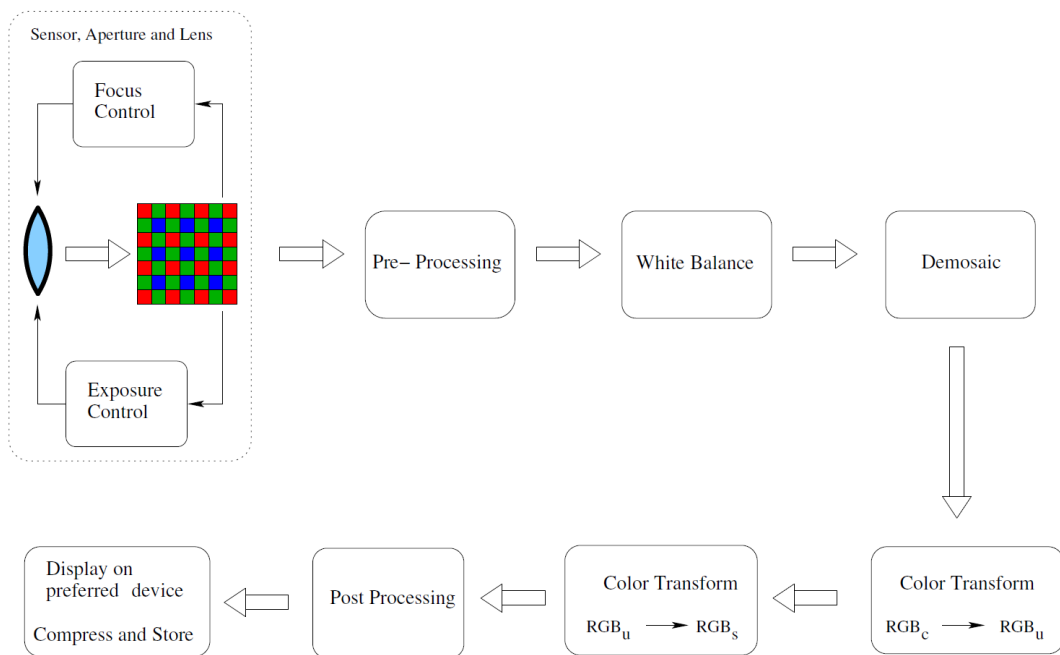


Figure 2-8: Image Processing Pipeline<sup>5</sup>

### 2.5.1 Demosaicing for Bayer CFA

Demosaicing for the Bayer CFA can be easily understood as an interpolation problem. A quincunx interpolation for the green pixels and a rectangular interpolation for the red and blue pixels.

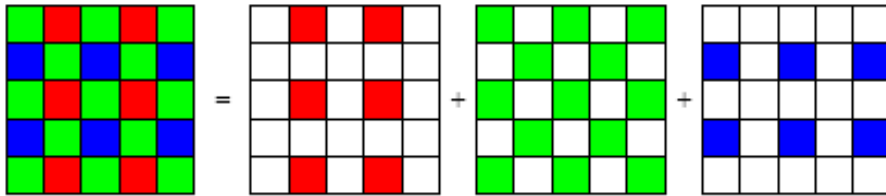


Figure 2-9: Bayer CFA<sup>3</sup>

Bayer proposed having twice the Green than Red or Blue based on the property of the Human Visual System having higher luminous efficiency in the green part of the spectrum, therefore allowing better sampling of luminance. However, it has not been demonstrated that whether in the brain, image formation follows similar approach to cameras, or if there are some higher level processes which modify the input signal. We know that there are inter individual differences in arrangement and ratio of cones<sup>1</sup> but despite that we seem to agree on perceived colors, which indicate that human vision is not so simple. Already the spectral sensitivity curves of the color filters on CFA are very different from those of the LMS cones. Using similar curves to LMS cones response we cannot get good colors from digital sensors. This is because the  $L$  and  $M$  cone response has too much overlap and is almost yellow. Separating these highly correlated channels results in noise increasing. In the next chapter we show that it is actually not a good idea to use more green than other colors because we find that it is the CFAs with quasi-equal distribution of RGB color filters that provide better reconstruction.

Several demosaicing approaches have been proposed in literature which makes use of the fact that color channels are heavily correlated and are broadly based on spatial interpolation, luminance-chrominance decomposition by frequency selection, graph theory, sparse learning, statistical and probabilistic models among others. To

provide some guidelines these methods are based on following ideas. 1. Spatial domain approaches like bilinear/bicubic/spline interpolation. The simplest demosaicing algorithm would be a bilinear interpolation. For the Green channel one can average the four neighboring pixels. For the Red and Blue channel one can average along the height and width. Also, do the average of four neighbors for the missing central pixels. Such an algorithm gives rise to false colors and aliasing in the high frequency areas of the CFA image. More complicated algorithms interpolate the Green channel first, determine local edge directions to further improve results<sup>26</sup>. 2. Constant Hue Assumption<sup>34</sup>, Hue (color ratio or differences) are constant within an object. So Green is first interpolated and then the color differences (Red-Green, Blue-Green) are interpolated. To reduce false colors, edge adaptive techniques are used to interpolate along edges and not across edges (where constant hue assumption fails). 3. Algorithms based on sparse learning utilize the fact that natural images are sparse when transformed in Discrete Cosine Transform or Wavelet space. This allows using techniques developed for compressive sensing<sup>27</sup> for demosaicing. 4. Several optimal CFA arrangements have also been proposed based on frequency representation and selection<sup>13,35</sup>. The CFA mosaic can be modeled as a spatial chromatic sampling and expressed in the Fourier domain<sup>36-38</sup>. The Fourier representation of the CFA, see Figure 2-10 allows designing of simple linear filters which are used to select part of the frequency corresponding to luminance and chromatic components. Despite proposal for optimal arrangement of CFAs and filters, this approach has its drawbacks. It supposes that every frequency component belongs to either luminance or chrominance and can be easily separated. It is not easy to design filters for random CFAs as the luminance and chrominance have lot of cross talk, as being assessed with the frequency approach. However the technique we will develop in this thesis allows us to demosaic such random CFAs and will be described in next chapter.

It would be correct to say that demosaicing is an extensively studied problem, a Google scholar search throws up 7,720 papers on 'color demosaicing' with 2,370 papers being published between 2015-2017, the period of this thesis. Needless to say, to do an extensive survey/presentation of these methods would be another thesis

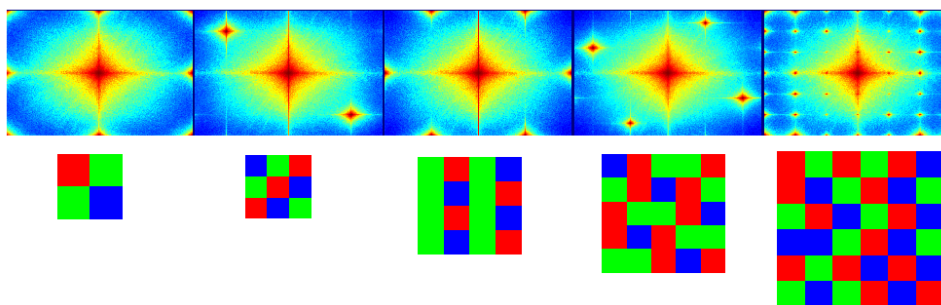


Figure 2-10: Different CFA types and corresponding spatial frequency response: Left to right: Bayer, Diagonal, Yamanaka, Holladay, CNRS<sup>6</sup>

altogether and beyond the scope here. As most of these algorithms are optimized for the Bayer CFA and cannot be easily ported to a random CFA. Whereas in this thesis we are concerned with demosaicing for random CFAs as it reflects more closely the mosaic of LMS cones in our eyes and our goal is to propose a generic algorithm for this.

Demosaicing can be understood as an inverse problem, in which we are trying to guess the original true color image from the subsampled image recorded by the camera. However it has no general solution. Let us consider the solution to be a black box which has the CFA image as the input and the output as the full color image. We have a database of true color images, so we can simulate a CFA image, pass it through the black box to get a reconstructed image. The goal in designing the black box is to minimize the difference between the original and reconstructed image. One can choose from a family of linear or non linear solutions to design our black box.

It is straightforward to consider linear solution<sup>39</sup> considering that demosaicing is a linear application from mosaicked image to a full color image space. Therefore one can derive a linear least square approximation of the full color image by minimizing the expectation of squared error of reconstruction<sup>40</sup> over an image database. Figure 2-11 presents the idea visually over a single image. Linear least square has the advantage that the solution is unique for a given problem. A given problem for us is the availability of a mosaic pattern with its color or spectral transmittance and a database containing exemplars of full resolution images.

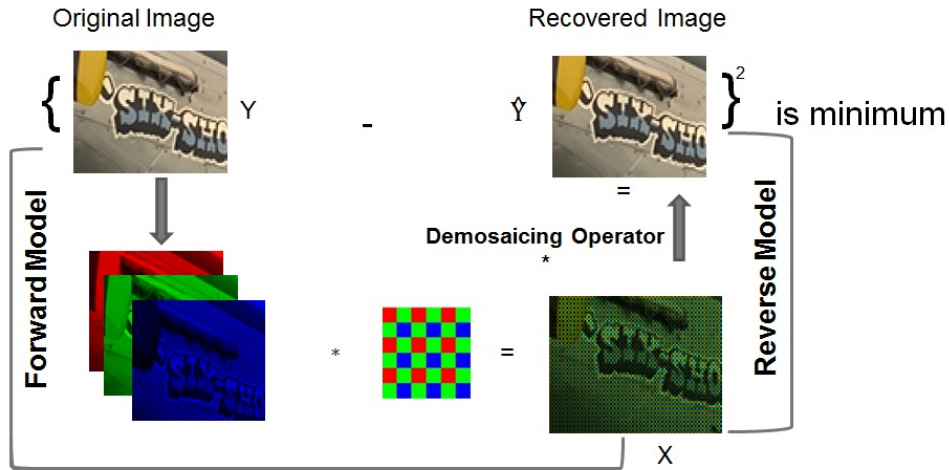


Figure 2-11: LMMSE the idea of minimising the MSE

## 2.6 Linear solution to the inverse problem

An inverse problem is so called because it attempts to guess the stimuli/physical properties by observing the effect. The forward problem is the other way around, when you have a cause and you predict the effect. Imaging problems are often inverse problems as our camera systems observe photon falling in it sensitive area and we often attempt to guess the physical properties of light<sup>39</sup>. For example, the problem of spectral reconstruction where one wants to retrieve the spectrum of light from its measurement on responses to few spectral filters. Linear solutions to inverse problems are preferred because of their simplicity and computational efficiency which is of importance as we need to process megapixels of data. Linear Least Squares provides an approach to solve a system of linear equations. Let us consider

$$x^{(k)} = My^{(k)} \quad (2.1)$$

which describes a linear model where  $x^{(k)} = [x_i^{(k)}]$  is the  $k^{th}$  realization of a random vector  $x^{(k)}$ , with component  $x_i^{(k)}$ , of an observation and we are trying to estimate the vector of the cause  $y^{(k)} = [y_i^{(k)}]$ . We call the estimate as  $\hat{y}^{(k)} = [\hat{y}_i^{(k)}]$ .

In most of the case, imaging problems are underdetermined because we want to retrieve properties of the light from few measurements. But, as we will see later, in



case of shift-invariant problems, we could favorably use the neighboring pixels to increase the redundancy. So for now we consider overdetermined problems.

Let us consider size of  $M$  being  $e \times d$  ( $e$  rows and  $d$  columns) with  $e > d$ . So,  $y^{(k)}$  is of size  $d \times 1$  and  $x$  of size  $e \times 1$ . In this case, solution can be written as<sup>39</sup>:

$$\min_{M^+} \|\hat{y}^{(k)} - y^{(k)}\|^2$$

$$\hat{y}^{(k)} = M^+ x^{(k)} \text{ where } M^+ = (M^t M)^{-1} M^t$$

The above equation describes  $M^+$  (also called the pseudo-inverse) being multiplied by the observations to get an estimate of the cause. In context of demosaicing  $M$  is the mosaicing operation,  $x^{(k)}$  an exemplar vector taken from the CFA image and  $y^{(k)}$  an exemplar vector taken from the corresponding color image. But this direct solution using pseudo-inverse unfortunately doesn't work. The reason belongs to  $M$  which is an idealization of the mosaicing process. In real camera, nothing prevents that a particular pixel receive light from its closed neighbors which will result in complex forward model containing complex spatio-spectral functions. Because we don't know how to properly describe the spatio-spectral of the optic process of acquiring an image through a CFA, we usually choose as  $M$  a selection function which selects which pixel from the color image belongs to the mosaic image. In this case,  $M$  would be solely filled by zeros and one values and it could be shown that the pseudo-inverse results in a transpose,  $M^+ = M^t$ . With  $M$  designed as selection operator, the pseudo-inverse  $M^+$  is not useful because it corresponds to a copy of the measured pixels into a color image filled with zero value at positions not measured.

It is more reliable to learn the mosaicing operator statistically from image database and compute its inverse. Actually, we could directly learn the inverse mosaicing operator as being an operator that transforms a mosaic image into a color image. We restrict the statistical learning to the second order statistics. In order two, statistics of random vector is assessed by cross-correlation. For example the cross-correlation matrix constructed from a single vector  $x^{(k)}$  is given by  $R_x^{(k)} = x^{(k)}(x^{(k)})^t = [x_i^{(k)} x_j^{(k)}]$ . It is a matrix of size  $e \times e$  containing the product of two of the components of the

vector  $x^{(k)}$ . For the vector  $y^{(k)}$ , cross-correlation is  $R_y^{(k)} = y^{(k)}(y^{(k)})^t$ .

Considering  $K$  several exemplars of the random vectors, noting  $x = [x^{(1)}, \dots, x^{(K)}]$  and  $y = [y^{(1)}, \dots, y^{(K)}]$  the matrices build from concatenation of random vectors, we can write:

$$x = My \tag{2.2}$$

So, the model apply also for the concatenation of different realization of the random vectors into matrix. Actually, this means that the relation works for any corresponding couples of vectors taken from CFA and color image. As a consequence, we can directly compute the expectation of the correlation for  $k$  several realization of the random vector using  $x$  and  $y$ :

$$R_x = E_{k=1..K}\{R_x^{(k)}\} = \frac{1}{K}xx^t \tag{2.3}$$

$$R_y = E_{k=1..K}\{R_y^{(k)}\} = \frac{1}{K}yy^t \tag{2.4}$$

This equation shows that expected correlation of vectors build from color images and CFA images taken from a database could be done by a single matrix product. We can even relate the correlation based on CFA image  $R_x$  with the correlation based on color image:

$$R_x = MR_yM^t \tag{2.5}$$

This thesis exploit this property and show how to design the different vectors  $x$  and  $y$  depending on the CFA arrangement and colors. But again,  $M$  is not a good operator to inverse and it is better to write directly an expression in term of demosaicing operator  $D$ . Actually, we are looking for a  $D$  operator that does the inverse of  $M$ . In our case of second order statistics, resume to an operator that predict  $R_y$  from  $R_x$ :

$$R_y = DR_xD^t \tag{2.6}$$

With this framework, demosaicing could be seen as an operator that predict missing

value in expected correlation of a color image based on the correlation of a CFA image that is measured. Wiener already shown in the context of stochastic process that there is a linear solution to such equation. Demosaicing operator is given by:

$$D = R_{yx} (R_x)^{-1} = yx^t (xx^t)^{-1} \quad (2.7)$$

where  $R_{yx} = \frac{1}{K}yx^t$ . Notice that  $K$  disappear in the equation meaning that the solution is consistent whatever is the number of realization to learn with. Actually, this solution is also described as being the indirect reconstruction of an inverse problem and corresponds to the least square estimate<sup>39</sup> that minimized the mean square error. In the rest of the thesis we will use this notation without explicit construction. We will detail the content of  $x$  and  $y$  matrices of vectors and how  $D$  is compute. We will also introduce a  $z$  variable which is the concatenation of vectors containing hyperspectral data. Actually,  $D$  could be written as  $D = R_y M^t (M R_y M^t)^{-1}$  which shows that even  $M$  is not representative of the real mosaicing process it intervenes in the computation of the demosaicing operator. But, notice its non trivial utilization.

Equation 2.7 is consistent for any number of realization of the random variable. When considering a database to learn with several images from which we could extract several number of random vectors, the parameter  $K$  will be specified. Because we will used several sizes for  $x$  and  $y$  depending on the neighborhood and other factors, we are not able to fix it now.

# Chapter 3

## Demosaicing for Random RGB CFAs

### 3.1 Introduction

In the previous chapter we introduced elements of the human visual system and the digital camera with its associated image processing workflow. We learnt that both the human eye and camera capture a sub-sampled color image. The biology does it in a random manner while our cameras apply a regular sampling like for the Bayer CFA. Straightforward demosaicing algorithms like those based on spatial interpolation or frequency selection give false colors and artifacts on demosaicing for the Bayer CFA which could be due to the regularity of sampling of color filters. The interpolation algorithms miscalculate in the region of high frequency content and due to regularity in color arrangement, the miscalculation is also regular and therefore becomes visible as false colors. It is possible to do more complicated algorithms like edge aware ones which avoid this problem by interpolating along the edge and not across it. However they are computationally more expensive.

Our goal here is to propose a demosaicing algorithm which works well with any random CFA and also to propose an optimal CFA with respect to ability to recover true color images. We present our method for demosaicing based on LMMSE with neighborhood, learnt on a database of full resolution images. Further, we use our

method to do a systematic evaluation of all possible arrangement of colors in a 2x2, 3x3 and 4x4 CFA size. Finally, we compare our method on proposed CFAs with the state of the art algorithms.

Because most of the CFAs are a replication of a basis pattern, a shift invariant solution could be found, which simplifies calculation by considering only the basis pattern (called super-pixel) replicated on the surface of the CFA, a 2x2 for Bayer pattern<sup>41,42</sup>. Despite the generality of the method which allows optimizations<sup>12,43</sup>, the solution obtained with such a procedure is not good, because the number of unknowns is larger than the number of inputs. An elegant way for improving the number of inputs is to consider a closed neighborhood around the position to be interpolated. Intuitively, this reinforces the statistical learning of the solution with existing data and provides good reconstruction results<sup>44-49</sup>. Furthermore, this framework allows the use of a random pattern inside the super-pixel<sup>50</sup>. Based on this method we can compare the performance of several random CFAs in reconstructing the color image from the CFA image. In the next section we formally describe our method of demosaicing with linear minimum mean square error by learning over a database with neighborhood. We then use this method to select optimal spatial arrangement of CFA for higher image fidelity.

## 3.2 Matrix model of image formation

Figure 3-1 shows the basic model of image formation of a CFA image from a full color image. Let us consider  $Y$  to be full color image, which is transformed by a mosaic  $M$  into a CFA image  $X$ .

The goal of our demosaicing algorithm is to estimate an operator  $D$  that will recover color image  $\hat{Y}$  from the CFA image  $X$  such that the difference between  $Y$  and  $\hat{Y}$  is minimized. We train this operator on a database of images as it leads to a more general and good performing solution. Now let us consider a color image  $Y$  having  $H$  rows,  $W$  columns and  $P$  color channels and the mosaiced image  $X$  having  $H$  rows and  $W$  columns. Writing a matrix model of image formation requires unfolding the

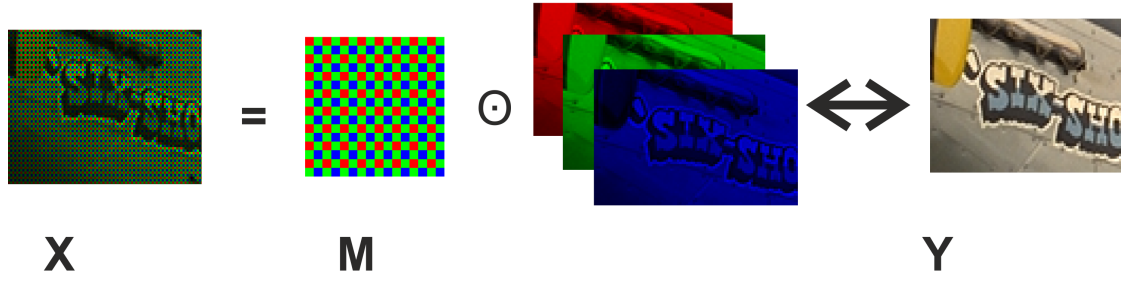


Figure 3-1: The image formation model. The color image  $Y$  can be considered to be composed of 3 separate color channels, here Red, Green and Blue images. Similarly  $M$  can be decomposed into three channels, for instance for the Red channel, we put 1 where Red is present, 0 otherwise and so on for other channels. These three  $M$  channels matrices are dot multiplied with three  $Y$  color channels, then summed to obtain CFA image  $X$ . The illustration approximates this procedure visually to allow easier understanding. The dot multiplication is in context for images, in the matrix model we are doing a vector-matrix multiplication.

matrix representing images into vectors, then finding a matrix-vector multiplication that relates the expected image from the acquired one<sup>41</sup>. In the case of demosaicing we suppose that the mosaicked image results from a color image multiplied by a projection matrix<sup>45</sup>. But there are many ways of unfolding images that results in different models. Classically an entire image is unfolded into a single, large column vector. For the demosaicing problem it is expressed as follows: We can construct the column vector  $y$  of size  $PHW \times 1$  corresponding to the color image and  $x$  of size  $HW \times 1$  corresponding to the mosaicked image<sup>41</sup>. In this case the model of image formation can be expressed as:

$$\begin{aligned}
 x &= My & (3.1) \\
 x &= [x^{(1)}, \dots, x^{(K)}] \quad , \quad y = [y^{(1)}, \dots, y^{(K)}]
 \end{aligned}$$

where  $x^{(k)}$  and  $y^{(k)}$  are exemplars vectors taken from respectively the CFA image and the color image extracted from a database.  $M$  is a  $HW \times PHW$  matrix that transforms the matrix  $y$  corresponding to a color image into a matrix  $x$  corresponding to the mosaicked image. As shown in section 2.6 the learned demosaicing operator taken  $K$  several exemplars of vectors (here  $K$  is the number of exemplar vector build

from images in the database) and the estimated matrix of vectors  $\hat{y}$  is expressed as:

$$\begin{aligned} D &= yx^t (xx^t)^{-1} \\ \hat{y} &= Dx \end{aligned} \quad (3.2)$$

In this model  $D$  is of size  $PHW \times HW$ . This model implies huge matrices as a model because the dimension of  $M$  or  $D$  is of size of the number of pixels in the images. This unfolding was one of the earliest works in approaching this problem was proposed by Trussell et al<sup>40</sup>. Images obtained from camera can be in order of 10s of megapixels and doing matrix multiplication and inverse operations on such a data size is unwieldy, therefore we need to find a way to reduce the size to be manageable.

A better model, Figure 3-2 is given by considering the block shift invariant property of the mosaic<sup>44</sup>. Since the mosaic is composed by a super-pixel of size  $h \times w$  replicated on the whole CFA of size  $H \times W$ , we can unfold the image for  $hw$  instead of  $HW$ . In this case the model formulation (equation 3.1) remains the same but  $y$  is now a  $Phw \times HW/(hw)$  column wise matrix containing the set of vectors built from one super-pixel in the color image. There are two ways to unfold  $y$ , either by color first and then by super-pixel or by super-pixel first and then by color. Now  $x$  is a  $hw \times HW/(hw)$  matrix corresponding to the set of vectors built from one super-pixel of the mosaicked image. For example, in Figure 3-2, we write the first column of  $x$  and  $y$  unfolded by color first.

$$\begin{aligned} x &= \begin{bmatrix} R1 & G2 & G3 & B4 \end{bmatrix}^t \\ y_{colorfirst} &= \begin{bmatrix} R1 & G1 & B1 & R2 & G2 & B2 & R3 & G3 & B3 & R4 & G4 & B4 \end{bmatrix}^t \\ &\text{unfolded color first and then super-pixel} \end{aligned} \quad (3.3)$$

There exists another way to unfold  $y$  by super-pixel first,  $\begin{bmatrix} R1 & R2 & R3 & R4 & \dots \end{bmatrix}^t$ .

In the rest of the document we will use color first unfolding.

For the Bayer CFA,  $h = 2$ ,  $w = 2$ , and  $P = 3$ , the matrix  $M$  when unfolding is by color first is:

$$M = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.4)$$

$M$  would be different if we unfolded by super-pixel first. We could choose either procedure for unfolding, however we need to ensure to use the correct  $M$  accordingly to the way  $x$  and  $y$  were unfolded. Thus, the matrix  $M$  selects the four pixels out of twelve pixels from the color image's  $y$ , to form each column in the CFA image's  $x$ . Thus  $M$  is a  $hw \times Phw$  matrix (i.e.  $4 \times 12$  for the  $2 \times 2$  super-pixel of the Bayer CFA) and  $D$  is a  $Phw \times hw$  matrix. This new formulation reduces the computational need to calculate  $D$  and apply the reconstruction to the acquired data. We thus write the inverse solution given by  $D$  as given by modifying the matrices sizes in equation 3.1 and 3.2 and accounting for the matrix  $M$  in equation 3.4 as follows:

$$\begin{aligned} D &= R_y M^t (M R_y M^t)^{-1} \\ R_y &= E_{k=1..K} \{ R_y^{(k)} \} \\ &= \frac{1}{K} y y^t \end{aligned} \quad (3.5)$$

with  $K = N_{DB} HW / (hw)$ , where  $N_{DB}$  is the number of images taken in the database.

This matrix model of CFA image formation was proposed by Parmar et al.<sup>12</sup> and used for selection of optimal spectral sensitivities of color filters. But with this model like the previous ones, the number of values to be retrieved is  $P$  times larger than the acquired values making the estimate quite unstable. To reinforce the stability of the solution, a neighborhood of  $x$  could be used. Let's  $x_1$  be a vector built from  $x$  and its close sliding neighborhood of size  $n_h \times n_w$ ,  $x_1 = N_{(n_h, n_w)}(X)$  is a function that increases the number of rows of a vector by the  $n_h \times n_w$  neighbors. By analogy



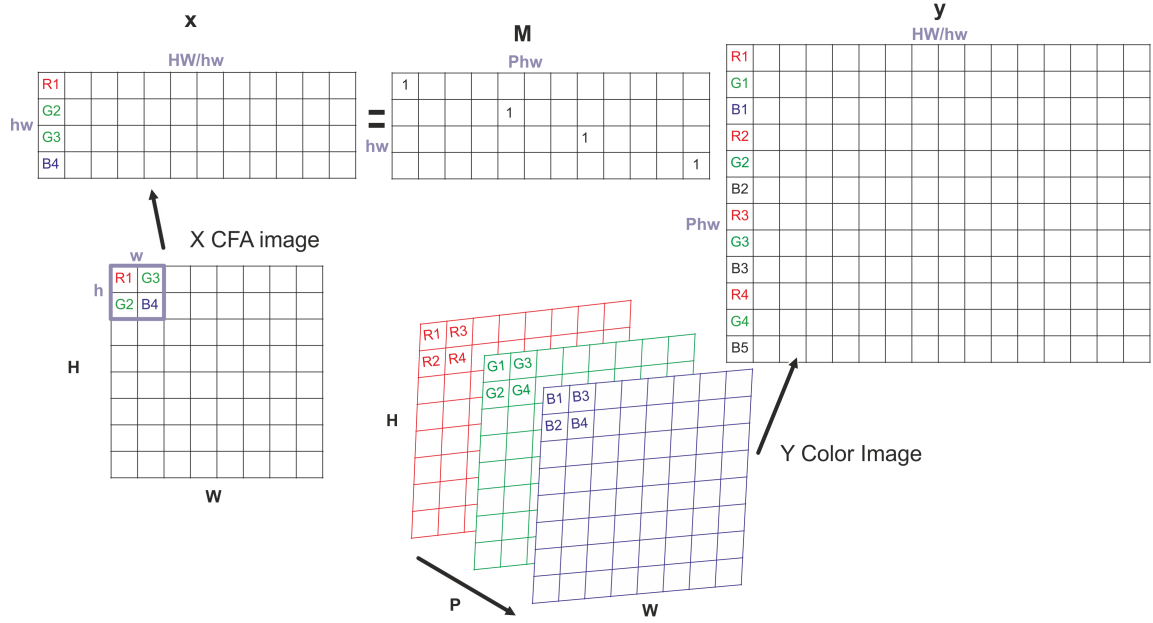


Figure 3-2: Matrix unfolding for  $Y$  and  $X$  into column vectors  $y$  and  $x$  by super-pixel size.  $y$  is unfolded by color first and then by super-pixel. We show unfolding for a single super-pixel in  $X$  to a single column in  $x$ . For populating the second column we shift the super-pixel across the surface of image first along  $H$  and then across  $W$ . This shifting is done super-pixel by super-pixel, thus making it block-shift invariant as we have the same  $M$ .

with convolution, we choose the neighborhood window to slide with the pixel into the super-pixel. So each of the  $hw$  pixels into the super-pixel has its own neighborhood window of size  $n_h \times n_w$  that is also unfolded into  $n_h n_w$ . In this case,  $x_1$  is of size  $hwn_h n_w \times HW/(hw)$  and the number of rows of  $x_1$  could be easily larger than  $Phw$ . Similarly, we consider  $y_1$  to be unfolded  $Y$  considering neighborhood of  $n_h \times n_w$  pixels for each pixel in the super-pixel. Figure 3-3 shows this unfolded along with the neighborhood.

This later formulation allows by analogy to design the demosaicing operator as:

$$\hat{y} = Dx_1 \tag{3.6}$$

$$D = yx_1^t (x_1 x_1^t)^{-1}$$

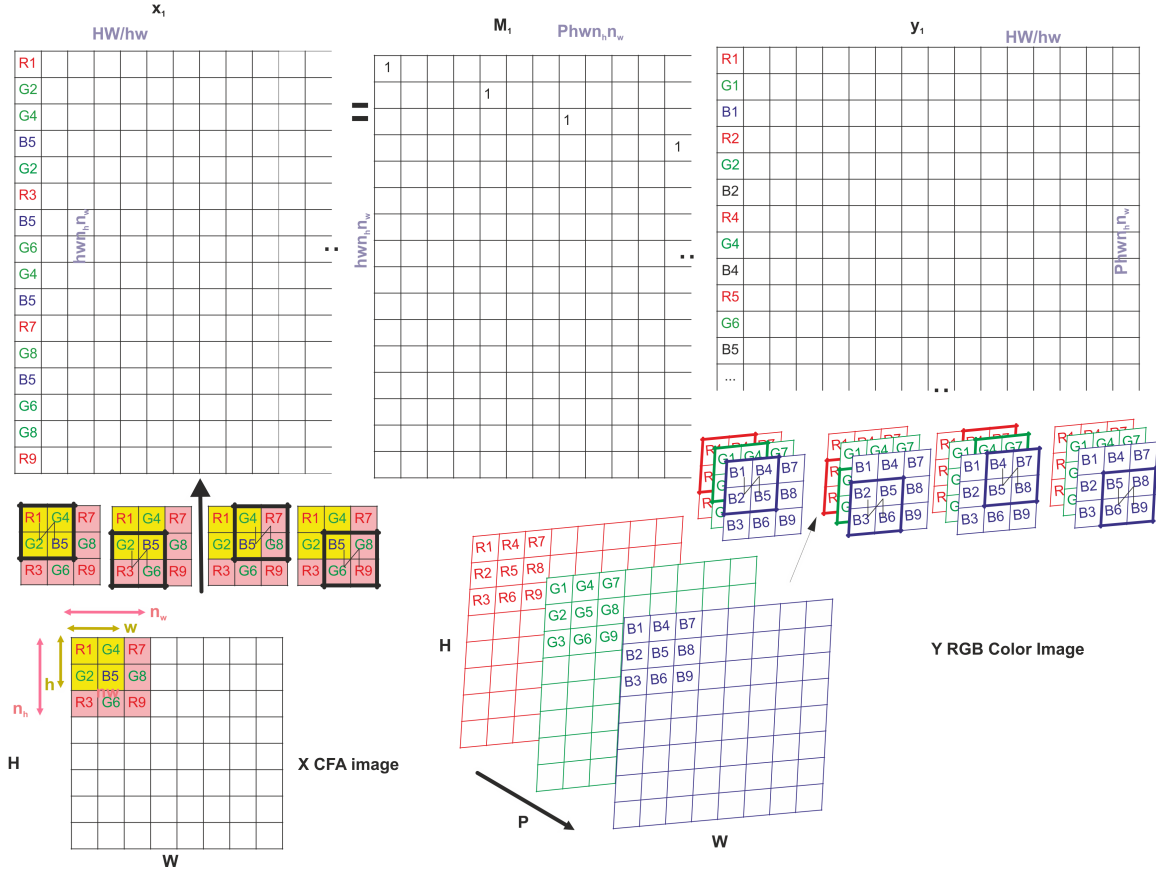


Figure 3-3: Unfolding for  $Y$  and  $X$  into column vectors  $x_1$  and  $y_1$  for neighborhood of size 2 using sliding neighborhood. For  $X$ , the CFA image, in yellow we see one super-pixel. Above it in the four sub-windows, the black square shows the neighborhood for each pixel in the super-pixel. The pink shows the extent of the neighborhood for all pixels in the super-pixel. To the right we see the same operation in  $Y$ , however it is repeated across the  $P$  color channels. The unfolding in this example is by color first, super-pixel second and finally the neighborhood for the  $Y$ .

This formulation using a sliding neighborhood was first proposed by Chaix et al.<sup>44</sup>. We extend it further by considering that similarly to the equation 3.2, it is possible to design a matrix  $M_1$  that transform a neighborhood in the color image (vector  $y_1$ ) into a neighborhood of the mosaicked image  $x_1$ , where  $x_1 = M_1 y_1$  (Figure 3-3). As for  $y$  earlier, equation 3.3, we can choose to unfold our images in many ways. Previously for  $y$  we had two options, color and super-pixel. Here we have three choices, color, super-pixel and neighborhood. The only difference it will make is in our writing of  $M_1$ .

It is also possible to design a matrix  $S_1$  that transform the vector  $y_1$  into the

vector  $y$ ,  $y = S_1 y_1$ , such that it suppresses the neighborhood and selects the central pattern. With these two matrices,  $D$  can be expressed as:

$$\hat{y} = Dx_1 \quad (3.7)$$

$$\begin{aligned} D &= S_1 R_{y_1} M_1^t (M_1 R_{y_1} M_1^t)^{-1} \\ R_{y_1} &= \frac{1}{K} y_1 y_1^t \end{aligned} \quad (3.8)$$

with  $K = N_{DB} HW / (hw)$

Equation 3.7 implies that we need to calculate the sliding neighborhood correlation  $R_{y_1}$  only once from the color images with their neighborhoods in the database. Then, for a particular CFA into consideration, we can construct  $M_1$  and  $S_1$  and compute the optimal demosaicing filter in the least square sense. Thus, with the same  $R_{y_1}$  we can compare the performance of any CFA by considering its corresponding  $M_1$  and  $S_1$  matrices.

In Lu et al.<sup>46</sup> a similar notation to Equation 3.6 is provided, but the neighborhood size is restricted to an integer number of the size of the super-pixel which becomes intractable when super-pixel size increases or for extended neighborhood use and is less flexible. Here, we proposed a generalization for any CFA with any super-pixel size and any size of neighborhood window and any arrangement of colors inside the super-pixel. The construction of  $M_1$  and  $S_1$  for a particular arrangement and a particular neighborhood is not trivial and cannot be described more here.

### 3.2.1 Sliding vs constant neighborhood

In the previous section we considered the sliding neighborhood method. We have other possibilities for managing the neighborhood. Figure 3-4 shows two possibilities. The yellow color denotes the super-pixel, the pink denotes the spatial extent of neighborhood for all pixels in the super-pixels. The black square shows the neighborhood for each individual pixel in the super-pixel. To recall, for the sliding neighborhood for every pixel in the super-pixels ( $h \times w$ ) we consider its neighborhood window ( $n_h \times n_w$ ), giving us  $hwn_h n_w$  values, rows in  $x_1$ . The same operation is repeated super-pixel

wise to populate all the columns in  $x_1$ . Let us consider toy-constant neighborhood

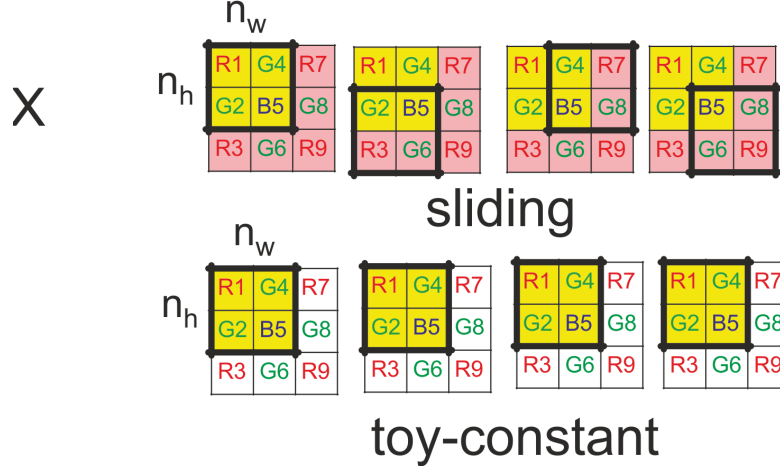


Figure 3-4: Illustration of sliding vs toy-constant neighborhood for 2x2 super-pixel, considering a neighborhood of 2x2.

method where for all the pixels in the super-pixels share the same neighborhood window ( $n_h \times n_w$ ), giving us  $hwn_hn_w$  values, rows in  $x_1$ . We use the notation as before for writing matrices and not distinguish neighborhood scheme for sake of simplicity. Depending on the neighborhood scheme one needs to ensure using the corresponding  $M_1, S_1, D, R_{y1}$  matrices. Toy constant case has the same computational complexity with the sliding case because matrices have the same size.

By simulation we show experimentally that the result is not the same for reconstruction of  $y_1$  as toy-constant provides less quality. From the Figure 3-4, it can be seen that the spatial extent of neighborhood for all the pixels in the super-pixel (denoted by pink) is smaller for toy-constant than for sliding. To ensure the same spatial extent we consider toy-constant with an expanded neighborhood window ( $n'_h \times n'_w$ ) for every pixel in the super-pixel ( $h \times w$ ), giving us  $hwn'_hn'_w$  rows in  $x_1$ . Actually, it is the extent of the neighborhood of all pixels in the super-pixel that count. For having the same performance in case of expanded toy-constant neighborhood we come to the following relation between  $n'_hn'_w$  and  $n_hn_w$ .

$$\begin{aligned} n'_h &= n_h + h - 1 \\ n'_w &= n_w + w - 1 \end{aligned}$$

We confirm experimentally that sliding and expanded toy-constant neighborhood give the same performance. It can be seen that the expanded toy-constant neighborhood is redundant in  $hw$ , therefore we can remove this redundancy and the number of rows in  $x_1$  is now  $n'_h n'_w$ . We call it constant neighborhood. Figure 3-5 shows the expanded toy-constant and the constant neighborhood. It can be written as follows:

$$\begin{aligned}\hat{y} &= Dx_1 \\ D &= S_1 R_{y1} M_1^t (M_1 R_{y1} M_1^t)^{-1} \\ R_{y1} &= \frac{1}{K} y_1 y_1^t\end{aligned}\quad (3.9)$$

with  $\hat{y}$  is of size  $Phw \times HW/(hw)$  containing the estimated vectors of the reconstructed color image,  $M_1$  is a  $n'_h n'_w \times Pn'_h n'_w$  matrix,  $D$  is a  $Phw \times n'_h n'_w$  matrix,  $S_1$  is a  $Phw \times Pn'_h n'_w$ .  $x_1$  is of size  $n'_h n'_w \times HW/(hw)$  and  $y_1$  is of size  $Pn'_h n'_w \times HW/(hw)$  and  $K = N_{DB} HW/(hw)$ .

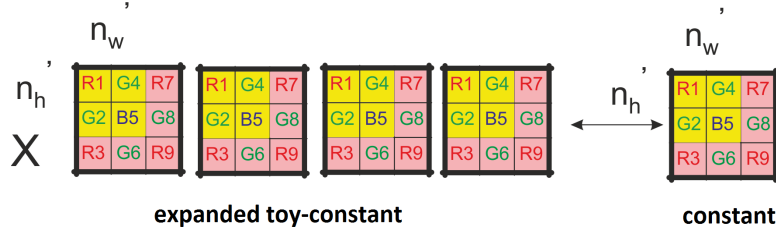


Figure 3-5: Illustration of expanded toy-constant and constant neighborhood for 2x2 super-pixel. For expanded toy-constant we consider a neighborhood of 3x3. This neighborhood was expanded from the sliding case to ensure same spatial extent of neighborhood for all pixels in the neighborhood. As the expanded-toy constant has redundancy by  $hw$  we simple reduce it and call it the constant.

In summary, we design a forward and reverse model that uses only neighborhood window for CFA image and provides reconstructed colors for the whole super-pixel without having repetition for the super-pixel in the CFA image. It is not trivial to obtain such a result, but introducing the toy-constant example allows us to show the different steps needed. Table 3.1 summarizes the various models presented so far.

| Matrix    | Trussell et al. <sup>41</sup> | Parmar et al. <sup>12</sup> | Chaix et al. <sup>44</sup> Sliding           | Ours constant                                |
|-----------|-------------------------------|-----------------------------|--|--|
| $x$       | $HW \times 1$                 | $hw \times HW/(hw)$         | $hw \times HW/(hw)$                          | $hw \times HW/(hw)$                          |
| $y$       | $PHW \times 1$                | $Phw \times HW/(hw)$        | $Phw \times HW/(hw)$                         | $Phw \times HW/(hw)$                         |
| $M$       | $HW \times PHW$               | $hw \times Phw$             | $hw \times Phw$                              | $hw \times Phw$                              |
| $x_1$     |                               |                             | $hwn_h n_w \times HW/(hw)$                   | $n'_h n'_w \times HW/(hw)$                   |
| $y_1$     |                               |                             | $Phwn_h n_w \times HW/(hw)$                  | $Pn'_h n'_w \times HW/(hw)$                  |
| $M_1$     |                               |                             | $hwn_h n_w \times Pn_h n_w hw$               | $n'_h n'_w \times Pn'_h n'_w$                |
| $S_1$     |                               |                             | $Phw \times Phwn_h n_w$                      | $Phw \times Pn'_h n'_w$                      |
| $D$       | $yx^t(xx^t)^{-1}$             | $R_y M^t (M R_y M^t)^{-1}$  | $S_1 R_{y_1} M_1^t (M_1 R_{y_1} M_1^t)^{-1}$ | $S_1 R_{y_1} M_1^t (M_1 R_{y_1} M_1^t)^{-1}$ |
| $\hat{y}$ | $Dx$                          | $Dx$                        | $Dx_1$                                       | $Dx_1$                                       |

Table 3.1: Summary of the various methods to unfold and construct the demosaicing operator as per Linear Least Square method.  $n'_h = n_h + h - 1$ ,  $n'_w = n_w + w - 1$ .

### 3.3 Simulation

With the framework given in the previous section, we can easily compare the performance of several CFAs with any super-pixel size and any arrangement of colors inside the super-pixel as well as any size of the neighborhood used for controlling redundancy. The framework works as follows: for any color image taken from the database, we compute  $y_1$ , composed by the set of vectors constructed for every pixel inside the super-pixels and their neighbors. From all  $y_1$  taken from all images in the database, we compute  $R_{y_1}$  according to Equation 3.9. Then we design  $S_1$  and  $M_1$  for the CFA and the neighborhood size. We compute  $D$  with Equation 3.9. The performance of the demosaicing is then computed as follows: for each image in the database, we compute the mosaicked image by sub-sampling the color image according to the CFA. Then we compute the vector  $x_1$  using the neighborhood. We apply  $D$  on  $x_1$  as in Equation 3.9 to reconstruct the estimate  $\hat{y}$  and compare it to  $y$  by calculating  $PSNR$  (A border equivalent to neighborhood size was removed in the calculation). Because all our images are normalized between 0 and 1, the PSNR is computed from the mean square error MSE as follows:

$$MSE = \frac{\sum \sum (\hat{y} - y)^2}{HWP}$$

$$PSNR = 10 \log_{10} \frac{1}{MSE}$$

We compute a PSNR from the whole mean square difference between the original and reconstructed image for all pixels. We use the average of whole PSNR over all the images in the database,  $\mu$  as an estimator of the overall quality of the reconstruction. The variance of the whole PSNR along image number,  $\sigma$  gives an estimate of the adequacy of the method to encode any particular image from a database. To test the method to equally encode any colors, we used the average of the PSNR per channel,  $\mu R$ ,  $\mu G$  and  $\mu B$  as well as the average of the variance of PSNR per channel,  $\sigma_{rgb}$ . Finally, the SSIM<sup>51</sup> is also provided to estimate the quality of the image in term of visual factors. We perform the analysis on two databases (Kodak<sup>15</sup>, McM<sup>16</sup>)

for comparing the performances. The Kodak database is known to have much higher frequency compared to the number of pixels and with low colorfulness which favor the edge directed and post processing methods. The McM database has been proposed as having more realist images in term of high frequency and colorfulness. We generally used all the images from the database for learning the demosaicing operator. We also implement a leave-one-out simulation where the image to reconstruct is not in the set of images used to learn.

## 3.4 Results

### 3.4.1 Systematic evaluation for 2x2 super-pixel size of the CFA

As a first example of performance comparison, we consider all the different combination of three colors R, G, B on a 2x2 super-pixel. The number of different possible arrangements is  $3^4 = 81$ , the choice of 3 colors over 4 different positions. Notice that a lot of them are symmetrical than others.

Figure 3-6 shows the performance of all 2x2 CFAs, arranged in decreasing order with respect to average PSNR ( $\mu$ ) for learning/reconstructing the entire Kodak database. From Figure 3-7, histogram of  $\mu$  for Kodak and McM database we clearly see three zones of distinction. Lowest performance is for CFAs with single color, middle for CFAs with two colors and higher when all three colors are present. When all three colors are present the performance is pretty similar showing the generality of our algorithm for any CFA arrangement. For the Kodak database in term of average PSNR,  $\mu$ , the best arrangement is not the Bayer RG;GB but slightly modified one where the arrangement is RG;BG. If we look at the average variance between PSNR calculated on individual color channels in the reconstructed images over the database,  $\sigma_{rgb}$ , the best is RB;GB arrangement. Also, if we look at the average variance of the overall PSNR,  $\sigma$ , along all the images in the database, the BR;GG is the best. This shows the following criterion that either twice of green or blue is preferred for



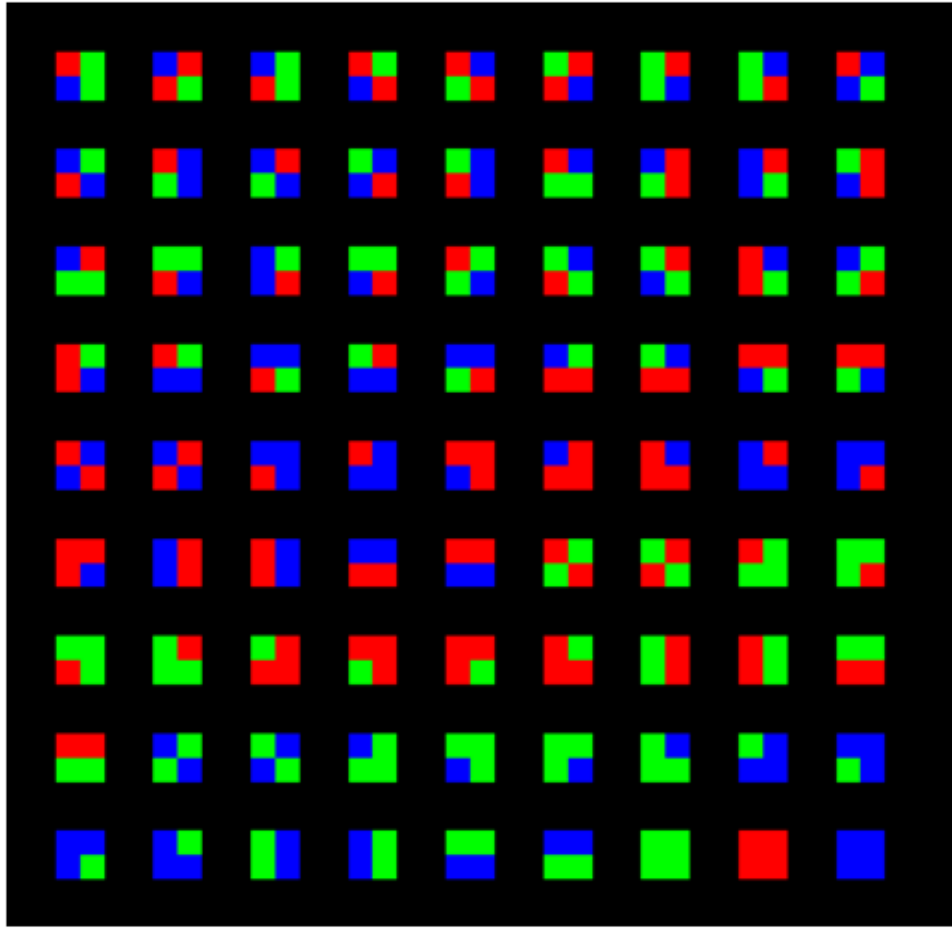


Figure 3-6: All possible CFAs of 2x2 size arranged by decreasing  $\mu$  (left to right then top to bottom) for Kodak database for a neighborhood of 7

variance criteria. (Figure 3-8). For the McM database we see that for lowest  $\sigma$  the CFA we found was all red, which is an anomaly. However this metric just denotes the variance between PSNR for all images in the databases, it is not concerned if they are equally good or bad. In this case they are equally bad. So we are searching for CFAs which should have both low  $\sigma$  and high  $\mu$ .

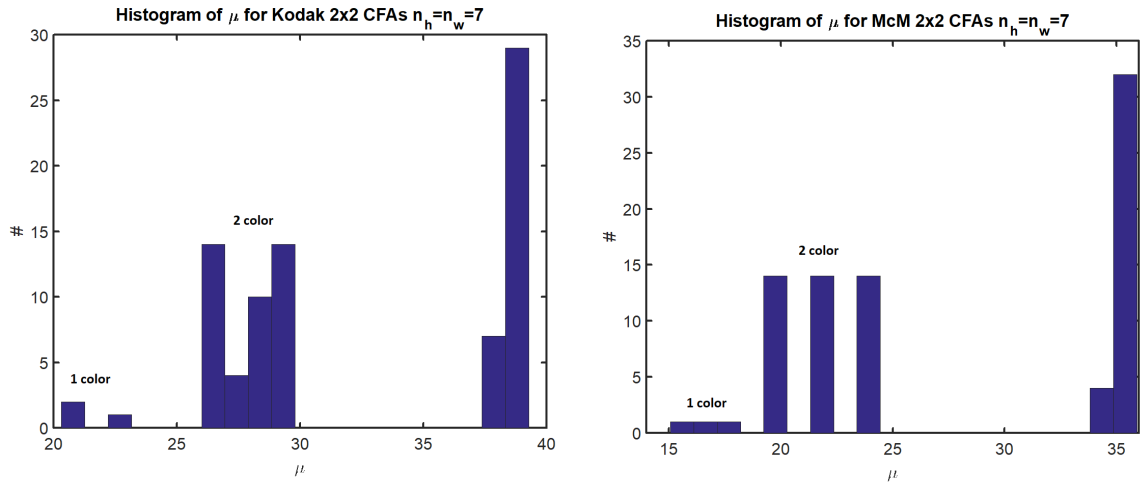


Figure 3-7: Histogram of  $\mu$  for Kodak and McM database for all 2x2 CFAs

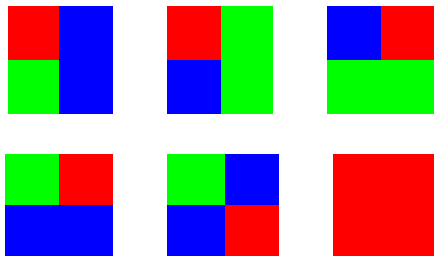


Figure 3-8: Top Row: Left to Right (Lowest  $\sigma_{rgb}$ , Highest  $\mu$ , Lowest  $\sigma$ ) for Kodak database. Bottom Row: Left to Right (Lowest  $\sigma_{rgb}$ , Highest  $\mu$ , Lowest  $\sigma$ ) for McM database

### 3.4.2 Systematic evaluation for 3x3 super-pixel size of the CFA

We consider all the different combinations of colors in the 3x3 super-pixel, there are  $3^9 = 19683$  different ones. Figure 3-9 shows the result for evaluating all the different

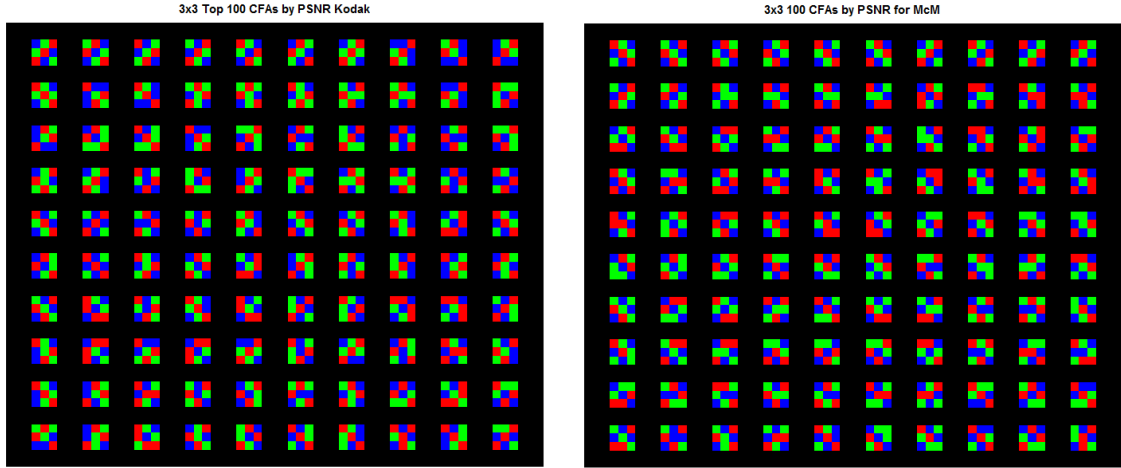


Figure 3-9: Top 100 CFAs by  $\mu$  for Left(Kodak), Right (McM) for neighborhood 7

3x3 CFAs. The top 100 CFAs by average PSNR  $\mu$  are shown. For the Kodak the top 7 CFAs are diagonal in nature, whereas for McM, the top 12 CFAs are diagonal. For the Kodak database each color in order (Red, Green and Blue) occurs a minimum of (3,2,2) times and maximum of (4,4,4) times. The PSNR ranges between 38.90 and 38.61 for the top hundred for neighborhood of 7 for Kodak database. For the McM each color (Red, Green, Blue) occurs a minimum of (2,2,3) and maximum of (4,4,4) times. It can be concluded that a quasi-equal distribution of colors favors ability to reconstruct image. As shown in Figure 3-9, among the best hundred 3x3 CFAs, some are symmetrical but none are perfectly periodic. Figure 3-10 shows the distribution of average PSNR for all possible 3x3 CFAs. From the top row, in each histogram (either for Kodak or McM), three different regions (single, two color and all three colors) can be easily visualized. Values less than 30 indicate at least 1 color was missing. Bottom row shows a zoom when all three colors are present. Again depending on the criterion  $(\sigma_{rgb}, \mu, \sigma)$  three different optimal ones are shown (figure 3-11).

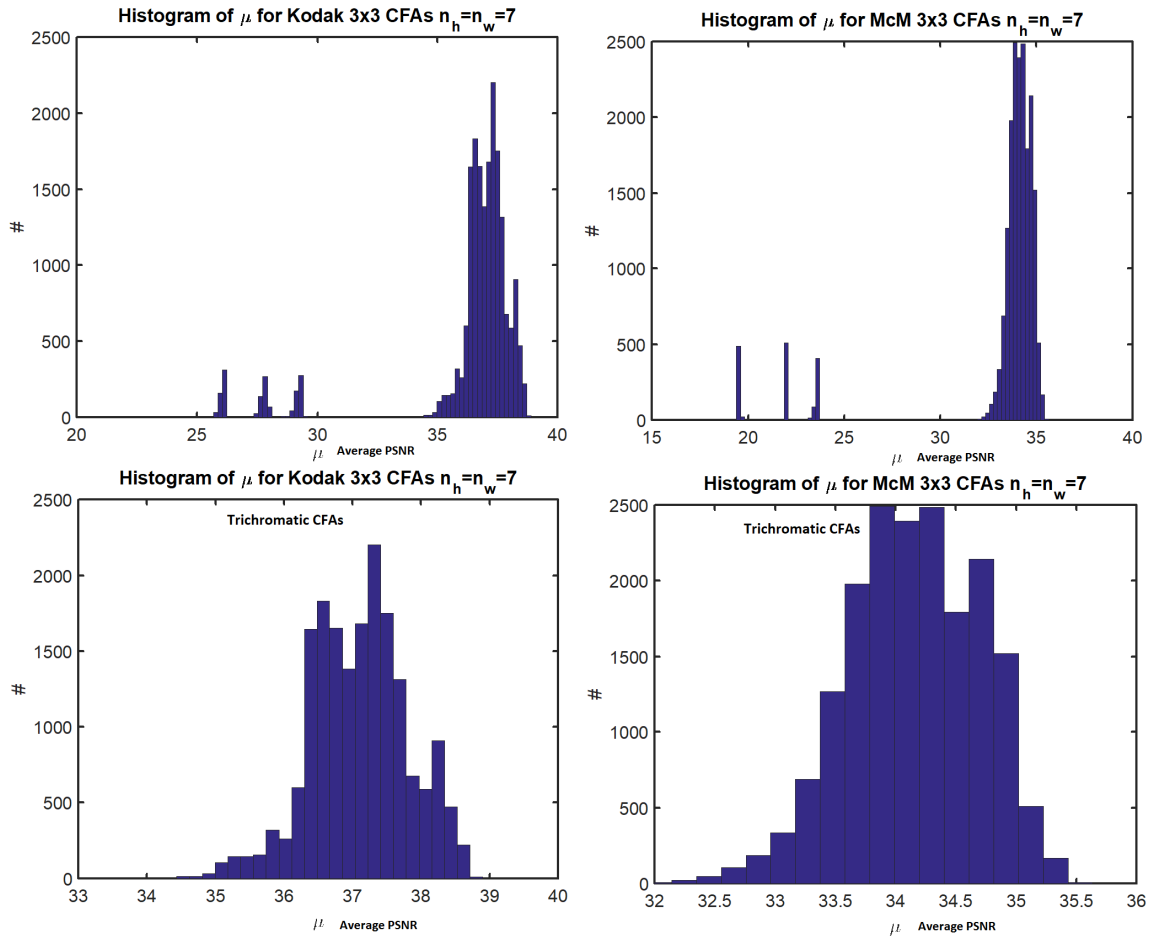


Figure 3-10: Histogram of  $\mu$  for Kodak and McM database for all 3x3 CFAs. Left is Kodak, Right is McM. Top Row shows for cases. Bottom row shows distribution when all three colors are present

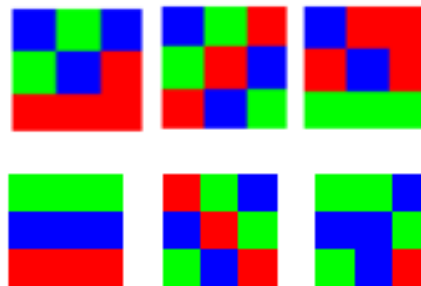


Figure 3-11: Top Row: Left to Right (Lowest  $\sigma_{rgb}$ , Highest  $\mu$ , Lowest  $\sigma$ ) for Kodak database. Bottom Row: Left to Right (Lowest  $\sigma_{rgb}$ , Highest  $\mu$ , Lowest  $\sigma$ ) for McM database

### 3.4.3 Evaluation of 4x4 super-pixel size of the CFA

There are  $3^{16}=43046721$  possible combinations of three colors on a 4x4 super-pixel, which is computationally expensive to test systematically. On our machine (Xeon e5-1603 v3 @ 2.8 GHz, 16 GB RAM) to systematically test each CFA for 24 images of Kodak database for neighborhood of 7 and evaluate for PSNR would require 17.6 years of computational time with the sliding neighborhood. Therefore we imposed rules to restrict maximum occurrence of each color to 11, minimum occurrence to 1 and that no color occurs in a cluster of 2x2 anywhere in the CFA, bringing down the total CFAs to 31483428. Also, we evaluated PSNR for a single image, Lighthouse of Kodak (half sized) for a neighborhood of 3. It brings down the computational time to a manageable 9 days. Refer to figure 3-12 for distribution of PSNR. For comparison PSNR for Bayer CFA for half sized Lighthouse image is 31.22. Only 0.11% of 4x4 CFAs tested performs worse than Bayer. So, it is clear that the majority of 4x4 CFAs thus selected (i.e. random) perform better than Bayer CFA (periodic).

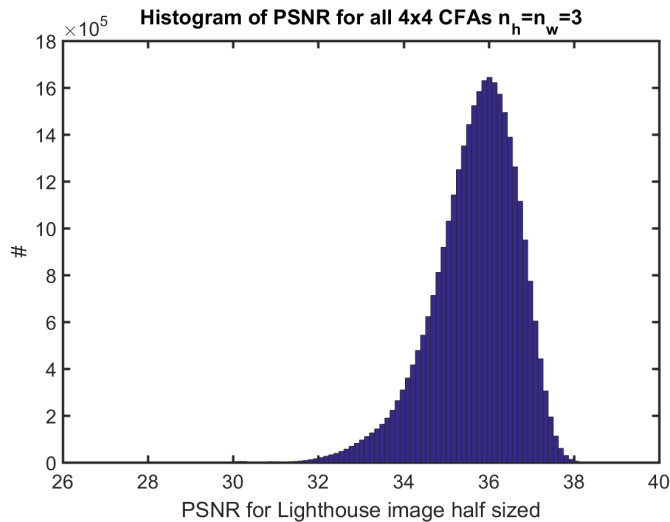


Figure 3-12: Histogram of PSNR for 4x4 CFAs

From this exercise we narrow down the top 1000 CFAs and then do the complete evaluation of average PSNR for entire database on them. Figure 3-13 shows the top 100 CFAs so obtained. Each color (Red, Green, Blue) occurs a minimum of (6,4,4) and maximum of (7,6,6) so we see a bias towards having more of Red than

Green or Blue. Again none of these CFAs are periodic in nature and be described as totally random distribution of colors. In the literature CNRS CFA<sup>52</sup> is proposed as

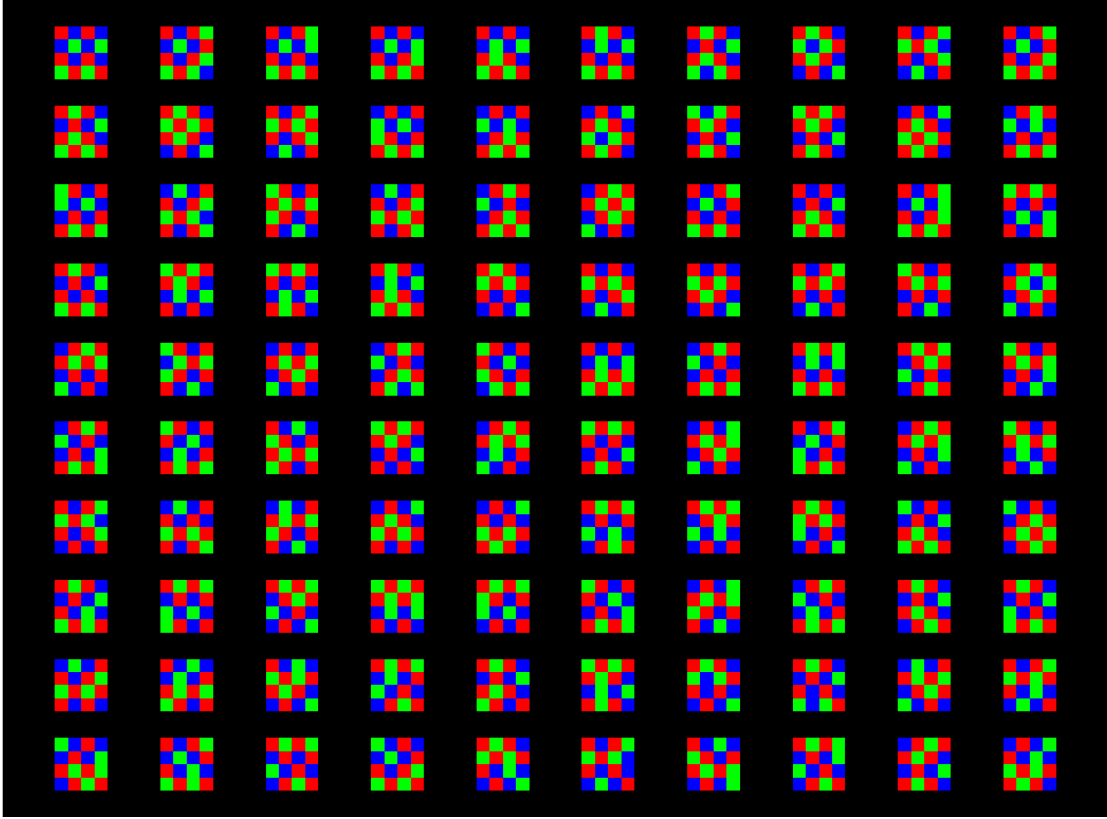


Figure 3-13: Top 100 4x4 CFAs by average PSNR for Kodak database

having good performance. We found a total of 978 CFA with better average PSNR ( $\mu$ ) performance than CNRS. Of these 94 were found to even have a lower average 3 channel variance ( $\sigma_{rgb}$ ) than the CNRS. In these each color (Red, Green, Blue) occurs a minimum of (5,4,5) and maximum of (6,5,6) times, i.e. we see again a quasi-equal distribution of colors.

An evaluation method will be presented in the next chapter that confirms that the best performance CFA for  $\mu$  is same even with evaluation on the whole database.

### 3.4.4 Comparison of CFAs under LMMSE

We select several CFAs (Figure 3-14) proposed in the literature that we have tested with LMMSE demosaicing. We also added the Bayer, best 2x2 and best 3x3 selected

based on Kodak database. We also selected the best 4x4 for highest PSNR  $\mu$ , called 4x4#1 based on procedure described above. We have also selected another 4x4 CFA which not only has a high average PSNR ( $\mu$ ) but also a lower 3 channel variance ( $\sigma_{rgb}$ ) called 4x4#2.

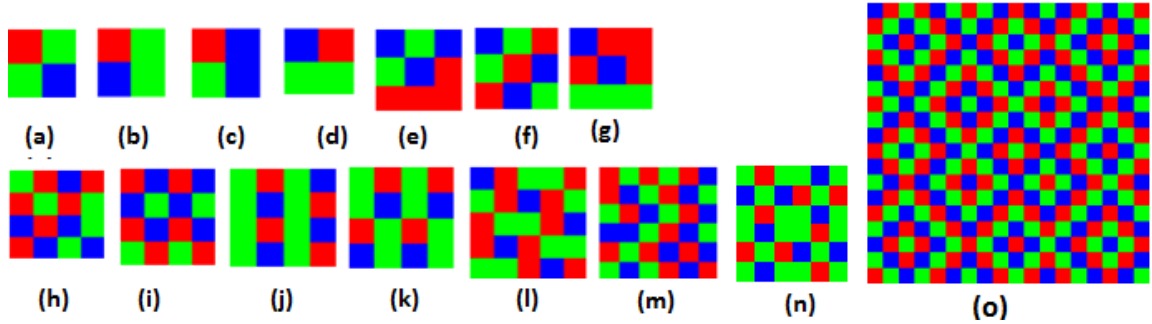


Figure 3-14: Different CFA patterns used for comparison. (a) Bayer, (b) 2x2#1 (Highest  $\mu$ ), (c) 2x2#2 (Lowest  $\sigma_{rgb}$ ), (d) 2x2#3 (Lowest  $\sigma$ ), (e) 3x3#1 (Lowest  $\sigma_{rgb}$ ), (f) 3x3#2 (Highest  $\mu$ ), (g) 3x3#3 (Lowest  $\sigma$ ) (h) 4x4#1, (i) 4x4#2, (j) Yamanaka<sup>11</sup>, (k) Lukac<sup>53</sup>, (l) Holladay halftone<sup>12</sup>, (m) CNRS<sup>52</sup>, (n) Fuji<sup>25</sup>, (o) Condat CFA of size 18x18<sup>54</sup>.

Figure 3-15 shows the performance of CFAs along the neighborhood size, which clearly favor the best 4x4#2 for a neighborhood of size larger than 3. This CFA is also performing well with variance estimators showing its ability to equally encode colors and perform well for any image in the database. In general we observe that average PSNR improves with increasing the size of neighborhood, with 7-10 being good criteria.

In the Figure 3-15-d we can see that a spatial neighborhood extent for all pixels in the super-pixel of 3 (inner black outline) for Bayer, implies that for each missing color, there are at least 4 pixel in the neighborhood extent where that color is known, which explain the first great increase in average PSNR when we go from neighborhood of 1 to 3. For neighborhood extent of 7 for Bayer, for each missing color there are at least 16 pixels in the neighborhood extent where that color is known therefore providing an even better estimate of the missing color. Beyond 10 for the average PSNR there is diminishing rate of return, i.e. a very small gain for a proportionally greater increase in computational time. Generally, CFAs with higher  $\mu$  and lower  $\sigma_{rgb}$

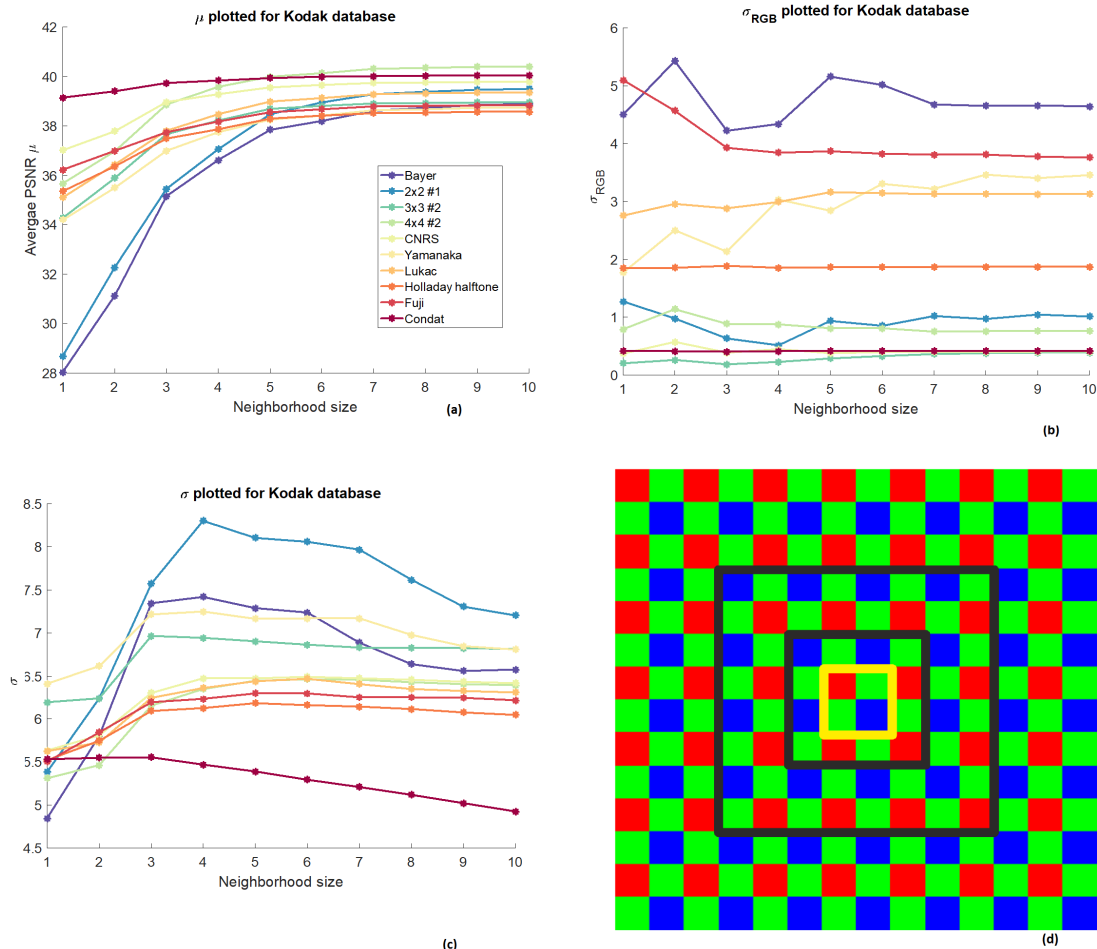


Figure 3-15: Evaluation of CFAs with LMMSE with increasing neighborhood (a) Average PSNR,  $\mu$ . (b) Variance of PSNR per channel  $\sigma_{rgb}$ . (c) Variance of PSNR for all images  $\sigma$ . A border equivalent to neighborhood size was considered (d) Spatial extent of pixels considering neighborhood of 3 (inner outline in black) and 7 (inner outline in black) on Bayer CFA.

seems to perform well visually, i.e. very less false colors.

Table 3.2 and 3.3 shows the number of the evaluation parameter estimated based on a neighborhood of 10 for the CFAs. We show the result for the 4x4#1 because even if it is not the best for average PSNR, it has very good visual performance on the fence of the lighthouse image. For Condat CFA of size 18x18 we could only test until a neighborhood of size 7 with sliding neighborhood. For neighborhood of size 10, 70.6GB of memory is required to store the correlation matrix which is not possible for us to test. For comparison for CFA of size 2x2 we require 4MB, 3x3 its 14Mb,



Table 3.2: Comparison of CFA for Kodak database,  $n_h = n_w = 10$  for all CFAs. Refer to Figure 3-14 for the CFAs

| CFA      | Kodak        |              |              |              |                |             |               |
|----------|--------------|--------------|--------------|--------------|----------------|-------------|---------------|
|          | $\mu$        | $\mu R$      | $\mu G$      | $\mu B$      | $\sigma_{rgb}$ | $\sigma$    | SSIM          |
| Bayer    | 38.90        | 38.57        | <b>41.53</b> | 37.59        | 4.64           | 6.57        | 0.9911        |
| 2x2#1    | 39.51        | 39.41        | 40.36        | 38.99        | 1.01           | 7.21        | 0.9923        |
| 2x2#2    | 39.10        | 38.71        | 39.39        | 39.32        | 0.53           | 7.64        | 0.9917        |
| 2x2#3    | 39.12        | 38.27        | 40.27        | 39.22        | 1.78           | <b>5.01</b> | 0.9916        |
| 3x3#1    | 37.16        | 37.40        | 37.13        | 37.02        | <b>0.22</b>    | 6.34        | 0.9881        |
| 3x3#2    | 38.96        | 38.70        | 39.48        | 38.78        | 0.39           | 6.81        | 0.9913        |
| 3x3#3    | 36.36        | 37.40        | 35.81        | 36.13        | 1.11           | 5.41        | 0.9859        |
| 4x4#1    | 40.26        | 40.51        | 40.32        | 40.05        | 0.40           | 6.44        | 0.9933        |
| 4x4#2    | <b>40.40</b> | <b>41.00</b> | 39.82        | <b>40.56</b> | 0.76           | 6.39        | <b>0.9936</b> |
| Yamanaka | 38.73        | 37.82        | 40.95        | 38.19        | 3.46           | 6.81        | 0.9910        |
| Lukac    | 39.35        | 38.70        | 41.47        | 38.57        | 3.13           | 6.31        | 0.9918        |
| Holladay | 38.57        | 39.23        | 39.62        | 37.30        | 1.87           | 6.05        | 0.9908        |
| CNRS     | 39.78        | 40.01        | 40.02        | 39.39        | 0.40           | 6.42        | 0.9927        |
| Fuji     | 38.83        | 37.88        | 41.21        | 38.20        | 3.76           | 6.21        | 0.9910        |
| Condat   | 40.05        | 39.74        | 40.63        | 39.86        | 0.42           | 4.92        | 0.9924        |

4x4 its 41MB. However it is possible to do it with the constant neighborhood, where the size of correlation matrix is only 33.5MB. This shows the huge advantage gained from using constant neighborhood even with a large super-pixel size.

For the Kodak database we find the best CFA for  $\sigma_{rgb}$  is 3x3 #1 and for  $\sigma$  is 2x2 #3. This is understandable as this was the criteria for selecting these CFAs. We choose 4x4 #1 as it gives a low  $\sigma_{rgb}$  along with a high  $\mu$ . Among other CFAs, CNRS and Condat were found to have good performance for  $\sigma_{rgb}$ , i.e. all color channels are equally well reconstructed. We find Condat CFA to be a good performer. In terms of  $\mu$ , the 4x4 #2 was the best performer. For the McM database, the Condat CFA has a good performance for several parameters. Although 4x4 #2 has a slightly lower  $\mu$  than Condat, it has a better SSIM performance.

Table 3.3: Comparison of CFA for McM database,  $n_h = n_w = 10$  for all CFAs. Refer to Figure 3-14 for the CFAs

| CFA      | McM          |              |              |              |                |             |               |
|----------|--------------|--------------|--------------|--------------|----------------|-------------|---------------|
|          | $\mu$        | $\mu R$      | $\mu G$      | $\mu B$      | $\sigma_{rgb}$ | $\sigma$    | SSIM          |
| Bayer    | 35.61        | 35.54        | <b>39.13</b> | 33.86        | 8.08           | 9.19        | 0.9827        |
| 2x2#1    | 35.13        | 35.47        | 36.82        | 33.81        | 3.03           | 9.36        | 0.9818        |
| 2x2#2    | 34.74        | 34.49        | 34.55        | 35.49        | <b>1.40</b>    | 9.60        | 0.9818        |
| 2x2#3    | 35.37        | 35.68        | 37.38        | 33.92        | 3.82           | 9.54        | 0.9824        |
| 3x3#1    | 34.58        | 35.98        | 34.18        | 34.02        | 1.95           | 9.26        | 0.9805        |
| 3x3#2    | 35.42        | 36.18        | 35.90        | 34.56        | 1.63           | 9.18        | 0.983         |
| 3x3#3    | 34.08        | 36.85        | 33.87        | 32.72        | 5.57           | 9.66        | 0.9785        |
| 4x4#1    | 35.44        | 36.96        | 35.89        | 34.14        | 2.83           | 9.32        | 0.9832        |
| 4x4#2    | 35.96        | 37.25        | 35.55        | <b>35.49</b> | 1.84           | 9.33        | <b>0.9851</b> |
| Yamanaka | 35.08        | 35.44        | 36.75        | 33.75        | 3.05           | 9.40        | 0.9815        |
| Lukac    | 35.60        | 35.81        | 38.18        | 33.96        | 5.24           | 9.06        | 0.9828        |
| Holladay | 35.27        | <b>37.39</b> | 37.21        | 32.96        | 7.24           | 9.15        | 0.9818        |
| CNRS     | 35.39        | 36.64        | 35.46        | 34.54        | 1.98           | 9.15        | 0.9832        |
| Fuji     | 35.14        | 35.11        | 38.70        | 33.35        | 8.39           | 9.13        | 0.9809        |
| Condat   | <b>36.05</b> | 36.90        | 36.54        | 35.09        | 1.57           | <b>6.75</b> | 0.9833        |

### 3.4.5 Effect of gaussian noise on LMMSE performance

In this section we demonstrate the effect of adding gaussian noise to the CFA images simulated from the Kodak and McM databases. The  $D$  operator was trained on noise-free images and used to demosaic noisy images. For the simulation we add gaussian noise of standard deviation five. Table 3.4 and 3.5 shows the result for different metrics tested for different CFAs. For the Kodak database and McM database we see that 4x4 # 2 has the overall best performance for the various metrics. For comparison with the state of the art, ACUDE<sup>14</sup> compared their approach with other algorithms in the literature for testing this particular noise and showed favorable results. Therefore we choose to compare with them.

Table 3.4: Comparison of CFA for Kodak database,  $n_h = n_w = 10$  for all CFAs, with additive gaussian noise of standard deviation=5. Refer to Figure 3-14 for the CFAs

| CFA      | Kodak        |              |              |              |                |             |               |
|----------|--------------|--------------|--------------|--------------|----------------|-------------|---------------|
|          | $\mu$        | $\mu R$      | $\mu G$      | $\mu B$      | $\sigma_{rgb}$ | $\sigma$    | SSIM          |
| Bayer    | 32.14        | 31.62        | 33.85        | 31.35        | 1.92           | 0.41        | 0.9208        |
| 2x2 #1   | 32.05        | 31.54        | 33.53        | 31.41        | 1.45           | 0.33        | 0.9181        |
| 2x2 #2   | 32.06        | 31.46        | 31.69        | <b>33.25</b> | 1.00           | 0.41        | 0.9238        |
| 2x2 #3   | 32.00        | 31.29        | 33.53        | 31.51        | 1.56           | 0.22        | 0.9171        |
| 3x3 #1   | 31.51        | 32.19        | 31.01        | 31.41        | 0.40           | 0.59        | 0.9196        |
| 3x3 #2   | 32.20        | 32.12        | 32.38        | 32.11        | <b>0.05</b>    | 0.41        | 0.9246        |
| 3x3 #3   | 31.31        | 32.26        | 31.21        | 30.63        | 0.75           | 0.63        | 0.9162        |
| 4x4 #1   | 32.30        | 32.60        | 32.27        | 32.06        | 0.10           | 0.21        | 0.9241        |
| 4x4 #2   | <b>32.38</b> | <b>32.76</b> | 31.86        | 32.58        | 0.25           | <b>0.20</b> | <b>0.9263</b> |
| yamanaka | 32.05        | 31.36        | 33.73        | 31.45        | 1.86           | 0.43        | 0.9198        |
| lukac    | 32.16        | 31.57        | 33.81        | 31.47        | 1.78           | 0.29        | 0.9199        |
| holladay | 32.02        | 32.59        | 32.72        | 30.98        | 0.98           | 0.39        | 0.9214        |
| cnrs     | 32.23        | 32.46        | 32.14        | 32.10        | 0.07           | 0.26        | 0.9240        |
| fuji     | 32.02        | 31.22        | <b>34.16</b> | 31.29        | 2.85           | 0.35        | 0.9180        |
| condat   | 31.98        | 31.12        | 33.13        | 31.92        | 1.06           | 0.29        | 0.9183        |

For instance with LMMSE, for Bayer CFA on the Kodak database we get a  $\mu$  of 32.14 dB, while ACUDE reports 33.49 dB. Similarly with McM with LMMSE, for Bayer CFA we get a  $\mu$  of 31.24 dB, while ACUDE reports 32.41 dB. For sure ACUDE has a higher performance than LMMSE, but one needs to consider that LMMSE take less than half second for Kodak images while ACUDE requires more than one and a

Table 3.5: Comparison of CFA for McM database,  $n_h = n_w = 10$  for all CFAs, with additive gaussian noise of standard deviation=5. Refer to Figure 3-14 for the CFAs

| CFA      | McM          |              |              |              |                |             |               |
|----------|--------------|--------------|--------------|--------------|----------------|-------------|---------------|
|          | $\mu$        | $\mu R$      | $\mu G$      | $\mu B$      | $\sigma_{rgb}$ | $\sigma$    | SSIM          |
| Bayer    | 31.24        | 30.72        | 33.44        | 30.24        | 3.17           | 1.67        | 0.9304        |
| 2x2 #1   | 30.99        | 30.59        | 32.69        | 30.12        | 2.06           | 1.86        | 0.9283        |
| 2x2 #2   | 30.91        | 30.41        | 30.41        | <b>32.22</b> | 1.42           | 2.20        | 0.9335        |
| 2x2 #3   | 31.03        | 30.65        | 32.85        | 30.10        | 2.32           | 1.72        | 0.9280        |
| 3x3 #1   | 30.57        | 31.57        | 29.88        | 30.47        | 0.92           | 1.92        | 0.9279        |
| 3x3 #2   | 31.31        | 31.57        | 31.46        | 30.98        | <b>0.31</b>    | 1.85        | 0.9350        |
| 3x3 #3   | 30.39        | 31.84        | 30.39        | 29.37        | 1.89           | 2.18        | 0.9249        |
| 4x4 #1   | 31.16        | 31.89        | 31.18        | 30.57        | 0.63           | 1.77        | 0.9329        |
| 4x4 #2   | <b>31.41</b> | <b>32.08</b> | 30.82        | 31.48        | 0.58           | <b>1.59</b> | <b>0.9363</b> |
| yamanaka | 31.05        | 30.73        | 32.69        | 30.16        | 1.97           | 1.95        | 0.9294        |
| lukac    | 31.20        | 30.78        | 33.11        | 30.23        | 2.49           | 1.64        | 0.9297        |
| holladay | 30.91        | 31.95        | 31.82        | 29.48        | 2.16           | 1.73        | 0.9276        |
| cnrs     | 31.18        | 31.76        | 31.00        | 30.87        | 0.45           | 1.80        | 0.9336        |
| fuji     | 30.97        | 30.42        | <b>33.55</b> | 29.83        | 4.20           | 1.79        | 0.9268        |
| condat   | 31.08        | 30.69        | 32.34        | 30.49        | 1.21           | 1.76        | 0.9299        |

half hour to process a single image, with the code publicly available. We don't lose much  $\mu$  compared to ACUDE while being less computationally complex.

### 3.4.6 Comparison with other methods on Bayer

We compare the best 4x4 using a neighborhood of 10 with the state of the art methods applying on the Bayer CFA using both the Kodak and McM database. The following Table 3.6 and 3.7 shows the evaluation parameters as well as the computation time on Matlab. The code for the algorithms is found on web site<sup>55</sup>.

Table 3.6: Comparison between the best 4x4 with LMMSE on Kodak database and other methods for noiseless case. LPA-ICI<sup>56</sup>, LIAN<sup>57</sup>, DA<sup>36</sup>, HD<sup>58</sup>, SA<sup>59</sup>, DFPD<sup>60</sup>, DLMMSE<sup>61</sup>, AP<sup>62</sup>, LI<sup>26</sup>, HA<sup>34</sup>, LLSC<sup>27</sup>, LDI NAT<sup>16</sup>, LDI NLM<sup>16</sup>, WECD<sup>63</sup>, LU<sup>64</sup>, ACUDE<sup>14</sup> are tested with a border of 15. 4x4#1 and 4x4#2 reported with a border of 10.

| CFA      | Kodak        |              |              |              |                |             |               |             |
|----------|--------------|--------------|--------------|--------------|----------------|-------------|---------------|-------------|
|          | $\mu$        | $\mu R$      | $\mu G$      | $\mu B$      | $\sigma_{rgb}$ | $\sigma$    | SSIM          | Time(s)     |
| 4x4#1    | 40.26        | 40.51        | 40.32        | 40.05        | <b>0.40</b>    | 6.44        | 0.9933        | 0.64        |
| 4x4#2    | 40.40        | <b>41.00</b> | 39.82        | 40.60        | 0.76           | 6.39        | 0.9936        | 0.65        |
| LPA-ICI  | 40.52        | 39.63        | 43.00        | 39.91        | 4.45           | 7.08        | 0.9874        | 1.15        |
| LIAN     | 39.53        | 38.59        | 42.13        | 38.86        | 4.63           | 7.39        | 0.9855        | 0.31        |
| DA       | 37.82        | 37.38        | 40.66        | 36.54        | 5.27           | <b>5.82</b> | 0.9829        | 0.09        |
| HD       | 37.72        | 36.94        | 39.59        | 37.26        | 2.82           | 8.64        | 0.9799        | 28.11       |
| SA       | 39.01        | 37.92        | 41.56        | 38.53        | 4.57           | 6.37        | 0.985         | 1.30        |
| DFPD     | 39.17        | 38.32        | 41.23        | 38.70        | 3.25           | 7.83        | 0.9837        | 1.55        |
| DLMMSE   | 40.05        | 39.12        | 42.58        | 39.53        | 4.79           | 6.92        | 0.9866        | 28.84       |
| AP       | 39.25        | 38.29        | 41.73        | 38.70        | 4.40           | 6.07        | 0.9849        | 1.61        |
| LI       | 35.66        | 35.16        | 38.83        | 34.25        | 6.05           | 10.24       | 0.9729        | <b>0.02</b> |
| HA       | 36.87        | 36.75        | 38.05        | 36.08        | 1.20           | 10.40       | 0.9769        | 0.08        |
| Bilinear | 30.19        | 29.25        | 33.07        | 29.26        | 4.94           | 10.94       | 0.916         | 0.04        |
| LLSC     | <b>41.46</b> | 40.53        | <b>44.36</b> | <b>40.67</b> | 5.52           | 6.70        | <b>0.9939</b> | 336.00      |
| LDI NAT  | 37.61        | 36.91        | 39.37        | 37.05        | 2.24           | 9.75        | 0.9799        | 925.61      |
| LDI NLM  | 37.66        | 37.09        | 39.43        | 36.96        | 2.27           | 7.43        | 0.9727        | 202.83      |
| WECD     | 39.16        | 38.06        | 41.92        | 38.69        | 5.42           | 6.02        | 0.9854        | 1.39        |
| LU       | 38.93        | 38.29        | 40.73        | 38.39        | 2.68           | 8.03        | 0.9839        | 980.31      |
| ACUDE    | 40.71        | 39.77        | 43.46        | 40.02        | 5.20           | 6.72        | 0.9934        | 5872        |

All the algorithms other than LLSC<sup>27</sup> proposed by Mairal et al. were tested on a Windows Xeon e5 1603 (4 cores @ 2.8GHz) with 16GB RAM. Matlab codes or Matlab executables as provided by authors was used. For LLSC the algorithm was tested on a Linux Xeon e5 1620 (4 cores @ 3.6GHz) with 8GB RAM, as the source code used executables compiled for Linux, giving it a potential advantage (as code is compiled + faster processor). However, this code makes use of a single core only

Table 3.7: Comparison between the best 4x4 with LMMSE on McM database and other methods for noiseless case. LPA-ICI<sup>56</sup>, LIAN<sup>57</sup>, DA<sup>36</sup>, HD<sup>58</sup>, SA<sup>59</sup>, DFPD<sup>60</sup>, DLMMSE<sup>61</sup>, AP<sup>62</sup>, LI<sup>26</sup>, HA<sup>34</sup>, LLSC<sup>27</sup>, LDI NAT<sup>16</sup>, LDI NLM<sup>16</sup>, WECD<sup>63</sup> and LU<sup>64</sup>, are tested with a border of 15. 4x4#1 and 4x4#2 reported with a border of 10.

|          | McM          |              |              |              |                |             |               |             |
|----------|--------------|--------------|--------------|--------------|----------------|-------------|---------------|-------------|
| CFA      | $\mu$        | $\mu R$      | $\mu G$      | $\mu B$      | $\sigma_{rgb}$ | $\sigma$    | SSIM          | Time(s)     |
| 4x4#1    | 35.44        | 36.96        | 35.89        | 34.14        | 2.83           | 9.32        | 0.9832        | 0.39        |
| 4x4#2    | 35.96        | <b>37.30</b> | 35.55        | <b>35.40</b> | <b>1.84</b>    | 9.33        | 0.9851        | 0.40        |
| LPA-ICI  | 34.70        | 34.32        | 37.88        | 33.29        | 6.62           | 13.20       | 0.9655        | 0.64        |
| LIAN     | 34.91        | 34.55        | 37.95        | 33.52        | 6.13           | 10.40       | 0.9673        | 0.17        |
| DA       | 32.22        | 31.82        | 34.69        | 31.07        | 4.16           | 13.33       | 0.9528        | 0.06        |
| HD       | 33.46        | 32.94        | 36.96        | 32.15        | 7.94           | 11.20       | 0.9576        | 18.50       |
| SA       | 32.69        | 32.56        | 34.42        | 31.70        | 2.71           | 17.88       | 0.9529        | 0.87        |
| DFPD     | 34.22        | 33.74        | 37.17        | 32.96        | 5.64           | 10.77       | 0.9624        | 0.99        |
| DLMMSE   | 34.43        | 33.97        | 37.90        | 33.02        | 7.93           | 11.21       | 0.9647        | 18.27       |
| AP       | 33.14        | 32.79        | 35.13        | 32.17        | 3.01           | 12.47       | 0.9567        | 1.03        |
| LI       | 34.39        | 33.94        | 37.52        | 32.97        | 6.03           | <b>8.75</b> | 0.9645        | <b>0.02</b> |
| HA       | 34.79        | 34.58        | 37.90        | 33.27        | 6.87           | 9.96        | 0.9654        | 0.05        |
| Bilinear | 32.29        | 31.65        | 35.38        | 31.20        | 6.59           | 11.58       | 0.9487        | 0.03        |
| LLSC     | 36.14        | 35.99        | 38.82        | 34.71        | 5.02           | 11.11       | <b>0.9858</b> | 219.00      |
| LDI NAT  | <b>36.19</b> | 36.26        | <b>39.78</b> | 34.39        | 8.68           | 9.19        | 0.9719        | 549.69      |
| LDI NLM  | 36.10        | 36.15        | 39.52        | 34.36        | 8.05           | 9.26        | 0.9692        | 129.10      |
| WECD     | 32.12        | 31.57        | 34.37        | 31.24        | 3.78           | 19.34       | 0.9477        | 0.88        |
| LU       | 35.58        | 35.13        | 39.30        | 34.18        | 8.99           | 9.12        | 0.9695        | 579.84      |

as it doesn't support multi-threading as per author. The execution time are average for the database and should be taken as a trend and not as absolute indicator of an algorithm performance as Matlab code is interpreted and not compiled and all the codes might benefit from optimization. For LMMSE 4x4#1 and 4x4#2 we show the performance for the sliding neighborhood, with the constant neighborhood we are at 0.3s for the Kodak database.

For the Kodak database LLSC has the best performance in  $\mu$  of 41.46, while our proposed 4x4#2 peaks at 40.40. However we are not clipping the values between [0 1] for LMMSE. If we were to do so we are at 40.68dB for 4x4#2. We are comparable in performance for SSIM. However our algorithm is blazing fast at 0.64s (sliding) compared to 336s for the LLSC. Our algorithm can be scaled easily, for instance for implementation on embedded systems where there are timing budgets. For a

neighborhood of 7, 4x4#2 has a  $\mu$  of 40.31, a loss of 0.09db for an execution time of 0.27s, i.e. twice as fast as earlier. Also our algorithm reconstructs all 3 channels equally well resulting in a lower  $\sigma_{rgb}$ . Next comparable algorithm is LPA-ICI at 40.52 but again we are almost twice as fast and have a better SSIM. ACUDE at 40.71dB also has a very good performance, however despite having high PSNR, it still exhibits false colors in the high frequency regions. Also we found it to be very slow. The authors claim that the code publicly provided is not optimized and they can do better. Appendix B Figure B-1 shows some examples of reconstruction based on particular area on images.

For the McM databases, we couldn't verify the results provided by ACUDE, therefore we don't present in the Table 3.7 as we don't have access to all the metrics from there paper. Overall for the McM database, the results get little murkier. The best performing algorithm is LDI NAT (which specifically promoted the database as better than Kodak) for  $\mu$  equal to 36.19. However at 550s it is very slow and it doesn't have a good performance for the Kodak database. Despite having a higher PSNR its SSIM score is lower than others. Visually results are good. Second best algorithm is LLSC at 36.14, but again it is slow and visually the result is not good (see Appendix B, Figure B-2). The best compromise is 4x4#2 which has a good  $\mu$ , good SSIM at a fast speed with visually pleasing result. Even though Condat CFA with LMMSE has slightly higher  $\mu$ , but visually the result is not so good. We also provide results for another CFA 4x4#1 which despite having lower  $\mu$  than 4x4#2 gives visually better results with less color noise. For sure the extent of the size of the super-pixel provides a better encoding of the scene as shown by the objective criteria such the PSNR. But in our simulations, we also see that the random arrangement of colors in the CFA also reduces the visibility of the noise generated by the mosaicking/demosaicing process because noise becomes less structured.

### 3.4.7 Effect of database on learning

The above results are demonstrated for the case when the entire database of Kodak and McM was used to learn the demosaicing operator  $D$ . Now we test the effect of

leaving one image out when learning on the database and calculating the difference with when the same image was also learnt (original case). Figure 3-16 shows then difference in PSNR when a particular image is not learnt. For  $4 \times 4$  for the Kodak database for neighborhood of 10 we see an average loss of 0.11dB while it is 0.15dB for the McM database, therefore the effect of leaving one image out can be considered negligible. However certain images are more susceptible to loss, notably image number 8 of the McM database, which has very little color compared to other images in the database. In general we have a smaller loss for Kodak than McM database. One possible explanation could be that McM has only 18 images of ( $500 \times 500$  pixels) while Kodak has 24 images ( $512 \times 768$  pixels), almost 2.1 times more data to learn from. Therefore there is more data to learn the statistics of images and it is more resilient. Some examples of the leave one out are presented in the Appendix B, Figure B-5 and B-6.



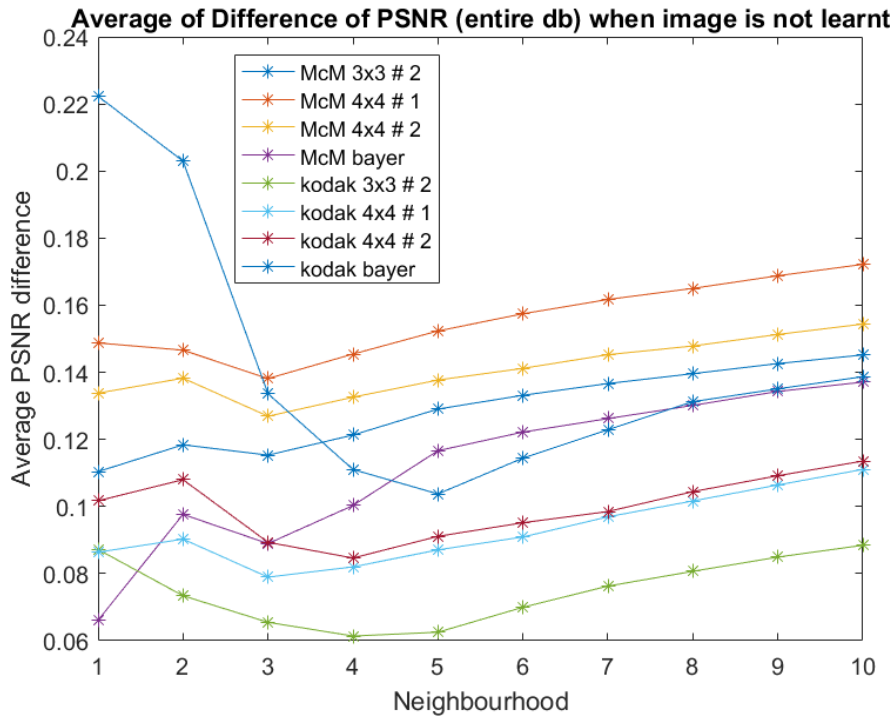
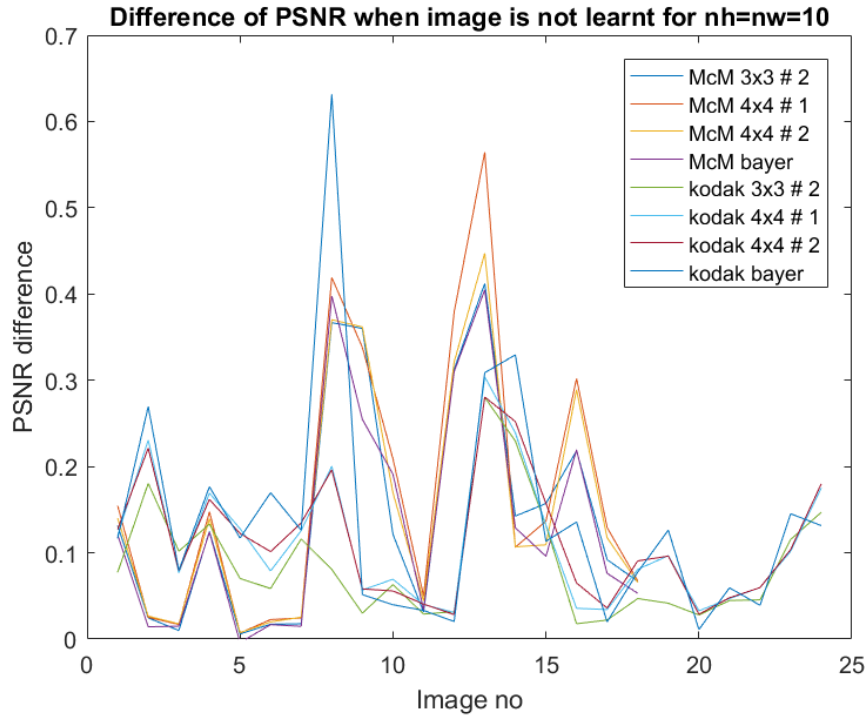


Figure 3-16: Top: Difference in PSNR when a particular image is not learnt. Bottom: Average of Difference of PSNR (entire database) along with neighborhood

### 3.5 Simulating totally random CFAs

We consider a totally random CFA, i.e. CFA with the same size of the image,  $H \times W$ . For example for Kodak images  $H=512$ ,  $W=768$ , (Figure 3-17). Now for demosaicing for this particular pattern, it is not possible to calculate  $D$  for such a big size due to memory limitations. Therefore we first divide the  $X$  into small 'cfasubsizes' and calculate  $M_1$  and  $S_1$  for each cfasubsize and demosaic it separately. Figure 3-18 for instance shows a cfasubsize of  $32 \times 32$  pixels. We need to rewrite  $M_1$  and  $S_1$  as the implementation from earlier cannot be reused in this case, as the assumption of block shift invariance is no longer true. We need to form the neighborhood by considering pixels from the neighboring cfasubsize block. For calculating  $D$  for all the blocks, we are looking at approximately 127s and 2.85s on average for actually demosaicing an image. We found a cfasubsize of 16 was the best compromise for efficiency. For 1000 random CFAs we got an average PSNR of 38.16 dB for the Kodak database for a neighborhood of 7. Figure 3-19 shows the histogram of average PSNR for 1000 different random CFA patterns. Equivalent value for Bayer for neighborhood of 7 is 38.62dB. Still, we have less false colors in the high frequency areas of the demosaiced image with totally random CFA pattern.

In literature, demosaicing using Kronecker Compressive Sensing (KCS) and Group-sparse reconstruction<sup>7</sup> have been proposed where they claim that these two yield good result for totally random sampling. We found that, using KCS, we found an average PSNR of 27.69dB in 181s per image. Using Group-sparse method we could get an average PSNR of 27.95 dB in 199s for a given random CFA sample. This is not good as be seen in the Figure 3-21 where we see output image is very noisy. They get good results in their paper because for 3-channel or 4-channel reconstruction they take the first 3 or 4 images of the Cave hyperspectral database<sup>65</sup> where the correlation between the images is very high (400nm, 410nm, 420nm, 430nm) which is not the case for RGB filters which are usually broadband and centered apart.



Figure 3-17: Random CFA super-pixel of size  $512 \times 768$  applied on the Lighthouse image

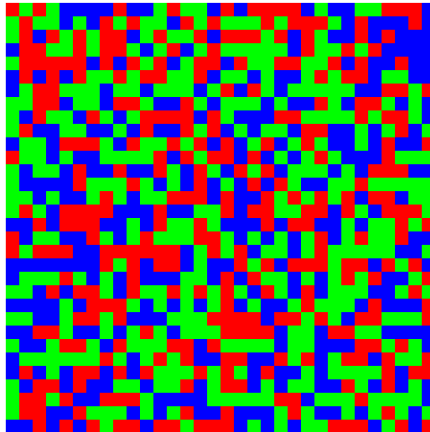


Figure 3-18: Crop of CFA super-pixel, showing  $32 \times 32$  pixels here

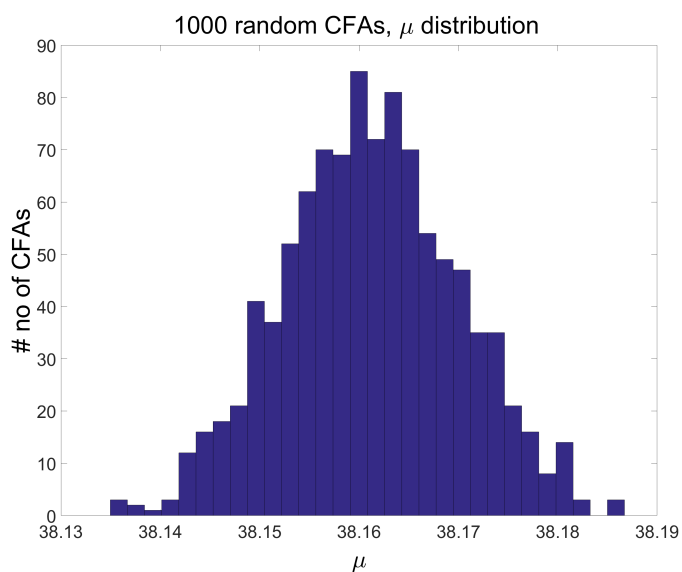


Figure 3-19: Histogram of average PSNR  $\mu$  for Kodak image database for 1000 totally random CFAs. We consider a neighborhood of 7 with cfasubsize of 16



Figure 3-20: Left: Demosaiced Lighthouse image for the Bayer CFA using bilinear interpolation. Right: Demosaiced image using our approach considering totally random CFA. We see that we have no false colors in the fence region in the random case.



Figure 3-21: Left: Demosaiced Lighthouse image using KCS<sup>7</sup>. Right: Demosaiced using Group Sparse<sup>7</sup>

### 3.6 Discussion

Evaluating the performance of demosaicing algorithm is not a straightforward task. There are several issues at large here. First, there is the question of the test database being used, as can be seen in the results section that all algorithms don't behave equally on each database. Even simple bilinear interpolation which is the weakest contender for average PSNR has visually better result for the McM database (see Appendix B, Figure B-2) as other sophisticated algorithms create artifacts at the edges.

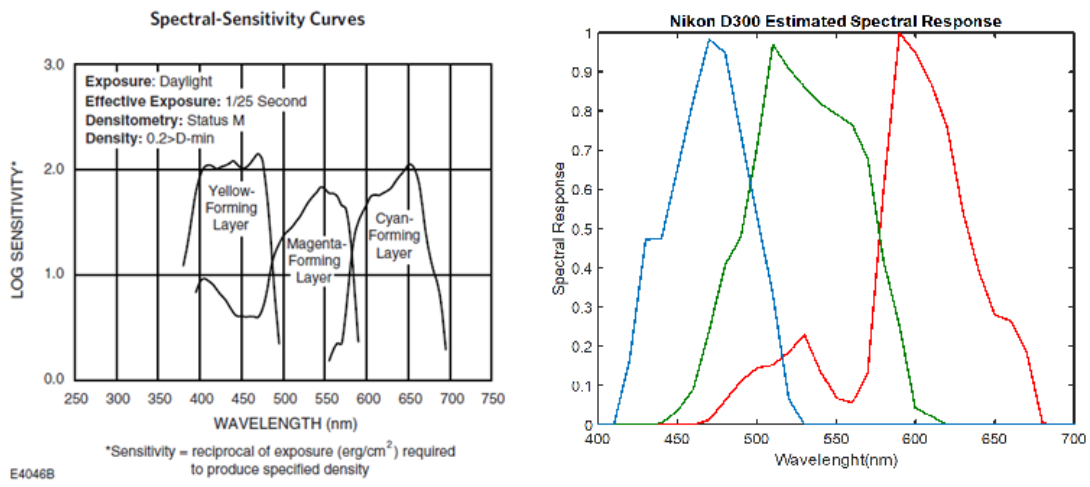


Figure 3-22: Left: Spectral Sensitivity Curves of Kodak Film 2273,3273/ESTAR Base<sup>8</sup> Right: Spectral Sensitivity curves of a Nikon D300 camera color filter

Regarding the Kodak database, these images were originally shot on 35mm Kodak film (different kinds, Ektar 25, VR 100, Gold 100, Ektachrome) and probably scanned to obtain true color images. Figure 3-22 compares the spectral sensitivity curves of a Kodak Ektar 100 film with a modern Nikon D300 camera. It is clear that they are very different. Little is known about the scanning process which was employed in scanning these images. The scanner lamp will have its own emission spectrum curve which will finally have an effect on the effective spectral sensitivity of the images recorded. The demosaicing algorithms have been developed keeping this database in mind and most of them are optimized for reducing the artifacts for this database.

Which becomes clear when the same algorithms are applied on McM database,

they don't perform very well. SA, DPF and WECD perform well for the Kodak but poorly for McM. On the other hand we have LDI NAT, NLM which performs well for McM but performs poorly for Kodak database. In terms of average PSNR only our proposed algorithm and LLSC seems to perform equally well for both Kodak and McM. However, despite LLSC having higher PSNR for McM database, visually its results are very poor compared to ours (see Appendix B).

In the literature now there is a trend<sup>13,66</sup> to recommend new optimal CFAs based on frequency selection which use more than 3 colors where the new colors are linear combinations of Red, Green and Blue Filters. However, it has not clear whether such color filters are physically possible to make. Secondly regarding metrics for performance evaluation, PSNR is considered the holy grail and there seems to be a race to reach to the top. However, we found it to be not always true. We have several examples where an algorithm with lower PSNR has less color noise, visual artifacts then one with high PSNR. Most glaring example is bilinear with 32.29dB compared to several others (LPA-ICI, LLSC) which have 36 dB for McM. Even with our proposed algorithm 4x4#1 despite having lower PSNR than 4x4#2 performs better in visual terms. Let's conclude that PSNR is important but it is not everything.

There are hyperspectral databases such as Cave or Finlayson<sup>17</sup> which can be used to generate true color images using any light source and color filters with specified spectral sensitivity transmittance function. However they generally are photographs of still subjects, due to high exposure times required and therefore don't contain high frequency elements and thus have their own limitations to evaluating demosaicing algorithms.

Another metric used to compare performance is SSIM which is a measure of structural similarity and to my understanding was developed for gray-scale images. If we leave bilinear interpolation out from comparison there is maximum difference of 0.0012 in SSIM, a mere 0.12% difference on average between algorithms. With such a small difference it is difficult to conclude anything from looking at this metric. The reason for this small difference is probably because we are averaging across entire images which tends to equalize the differences in SSIM values. It might be more appropriate

to test SSIM on small regions of the image where artifacts are more probable. Also it is difficult to draw any judgments based on visual appearance of false colors. A point to remember here is that not all monitors are equal and it is important to properly calibrate monitors when evaluating algorithms for artifacts which touch 40dB for the Kodak database. There are other algorithms available in state of art which claim good performance however we couldn't compare them here because the executable codes are not available. Comparing directly with the results produced in papers is not always possible, as sometimes there is lack of clarity regarding images tested. Some authors choose to only test 23 images from Kodak database, other might chose to do it on only 12 images. Some mix both the Kodak and McM database and present an average of some images. We believe there is a need to streamline image databases and testing methodology so that decisions regarding performance can be made objectively.

### 3.7 Conclusion

In this chapter we provide a flexible, fast and accurate linear minimum mean squared error demosaicing using the redundancy given by the neighborhood of the sampled image. The method is quite fast and allows us to systematically compare the performance of 2x2 and 3x3 CFA's super-pixels and most of the 4x4 CFAs. Compared to frequency selection approaches used today for optimizing CFAs, our method does not guess the frequency spectrum of the sampled image by the CFA. Rather, it uses a learning procedure that computes optimal reconstruction filters. Even when the aliasing between luminance and chrominance is strong (as arising for random pattern), the method finds good linear reconstruction filters. In this chapter we showed that random CFAs performed better than regular ones by doing a systematic evaluation and also evaluating our algorithm on totally random CFAs. We found that the best CFAs are the ones with non-periodic arrangement of colors (some are symmetric along the  $\pm 45^\circ$  line) and almost equal number of RGB. We demonstrated that for totally random CFAs the color noise was not present in the higher frequency regions rather spread out across the entire image, making it visually less disturbing compared



to what we would have with Bayer CFA. This proves our hypothesis that randomness as present in biological systems like the human retina has an advantage on the regular sampling patterns traditionally used in digital systems. It is a question of developing algorithms which can account for this randomness which we do by training on a database of natural images, thereby learning a model of image formation on a random mosaic. We showed that with our LMMSE algorithm we can get comparable performance to the state of the art. We also performed an analysis with images corrupted with gaussian noise and showed that our algorithm fares favorably with the state of the art. The best known methods for demosaicing are thought to be those with edge directed interpolation and post processing. We show that the 4x4 best CFAs give reconstruction results equivalent to the best nonlinear algorithms applied on Bayer. It even provides less variability among colors and particular image in the database. Moreover it gives better SSIM evaluation for a less computation time. This result suggest linear demosaicing for being favorably used in the embedded camera devices. Statistics of natural images are probably random because a contour or a particular object's color could potentially appear anywhere on the images. That is probably why random CFA perform better than periodic one for encoding the spatial and chromatic structure of natural images.

# Chapter 4

## Multicolor CFA's

### 4.1 Introduction

Historically, demosaicing methods was designed without any knowledge of the spectral transmittance of the filters. One just considered R,G and B to be independent to each other and reversed the problem of mosaicing as three interpolation problems. Nethertheless, correlation between color channel has been shown today as a requisite for good spatial and color quality of the reconstruction. If we take into consideration color transform from the camera color space to the RGB space we can evaluate multicolor CFA with RGB CFA. We restrict the study to multicolor being linear combination matrix  $A$  of RGB. Since the performance of multicolor CFA is given by the parameter on  $A$ , we can optimize these parameters for improving the reconstruction. In the state of art several optimized CFAs have been proposed considering this method<sup>13,14,35,67,68</sup>. Therefore, we believe it provides a good comparison of our demosaicing approach with them. We extend the LMMSE framework presented in the previous chapter by considering a color mixing matrix  $A$  in between  $M$  and  $y$ , which allows to compare with state of art demosaicing algorithms. We also present an approach for optimization by evaluating average MSE across an image database without the need of explicitly demosaicing and evaluating every single image. We express average MSE in term of correlation matrix  $R_y$  and other matrices ( $M_1$ ,  $S_1$ ,  $D$  and others) related to the CFA in question. Therefore we avoid the need for a

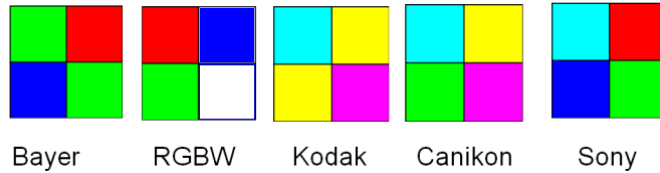


Figure 4-1: CFAs with RGB and multicolor CFAs<sup>3,9,10</sup>

systematic evaluation of every images in the database.

In the late 1990s and early 2000s cameras did make use of Cyan, Yellow, Magenta or even Emerald colors (CYME) in the CFAs. Kodak is known for using CYYM<sup>3</sup>, Canon and Nikon for CYGM<sup>9</sup> and Sony for RRGBE<sup>10</sup> color filters in their cameras (Figure 4-1). Recently White (absence of any color filter) has also made its appearance as it theoretically helps to recover more dynamic range and therefore has applications in low light photography. Lately, there has been a renewal in designing optimal CFAs considering more than three primary (RGB) colors. Because in the general case, multicolor CFA could not be displayed, an additional step after demosaicing is required to convert the image to RGB format for display.

If we perform a DFT (Discrete Fourier Transform) of a CFA image we see that the luminance and chrominance components are heavily multiplexed, however for periodic CFAs like the Bayer they are localized separately. Luminance in the low frequency regions and chrominance in the higher<sup>36,37</sup>. To demosaic successfully we need to design filters to separate them as cleanly as possible<sup>38,57</sup>. The chrominance component is sub-sampled, so we need to interpolate it and add back the luminance to get the final color image. The design philosophy for CFAs is to move the chrominance components away from the horizontal and vertical axis where luminance has its maximum intensity<sup>13,35,66-69</sup>. See Figure 4-2 for average of DFT for Kodak database images<sup>15</sup> for several CFAs. We see that CFAs like Yamanaka<sup>11</sup>, Holladay<sup>12</sup>, Bai<sup>13</sup>, Hao 40 and Hao 50 (to some extent)<sup>14</sup> fulfill this criteria. Then, we need to design demosaicing low pass and high pass filters which can separate luminance and chrominance. Generally this approach works for periodic CFA patterns and cannot work for random CFAs as chrominance components are present across the entire frequency band. See Appendix B.4.

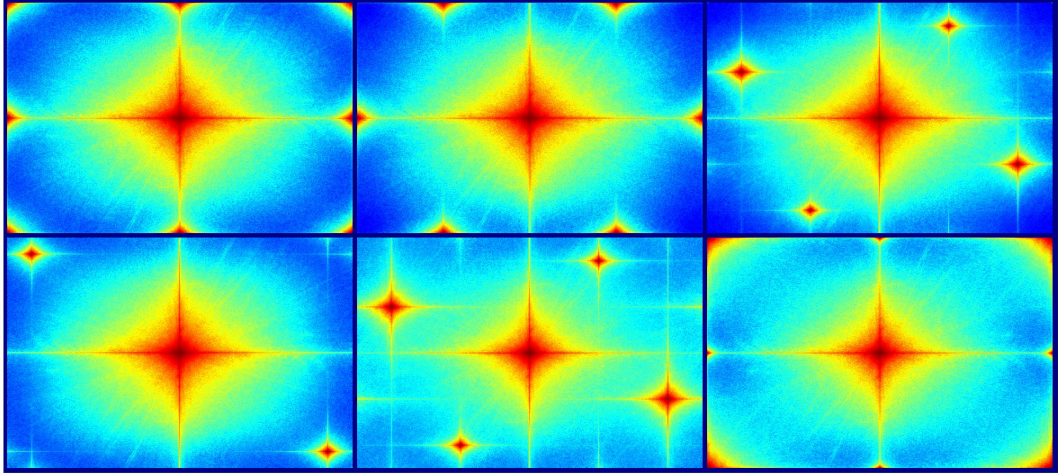


Figure 4-2: Average DFT for Kodak database. First row (LtoR): Bayer<sup>3</sup>, Yamaguchi<sup>11</sup>, Holladay<sup>12</sup> Second row(LtoR): Bai<sup>13</sup>, Hao40, Hao50<sup>14</sup>

For random or periodic CFAs we can consider demosaicing as an inverse problem of estimating the missing colors from the sampled ones. In the previous chapter we described our Linear Minimum Mean Square Error (LMMSE) based approach for demosaicing any CFA. In the next section we explain the matrix model of our LMMSE based solution adapted for the 'multicolor CFAs'. Then we use a gradient descent method to find the optimal CFA pattern which gives the best performance.

## 4.2 Matrix Model for Demosaicing for multicolor CFAs

In a vector space model of color vision, any general color can be considered as a linear combination of three primaries Red, Green and Blue. Let us define  $C$  to be an 'artificial color' image having multicolor channels.  $C = [\alpha_i R + \beta_i G + \gamma_i B]$  is a  $H \times W \times P_C$  image containing several color channels build from a linear combination of R, G and B images of size  $H \times W$ . We can consider as many colors  $P_C$  as the size of basis pattern  $h \times w$ ,  $P_C \leq hw$ . Here, white is a case where  $\alpha = \beta = \gamma = 1$  and black where  $\alpha = \beta = \gamma = 0$ . We can express  $c$  of size  $P_C hw \times HW/(hw)$  which is

unfolded multicolor image  $C$  of size  $H \times W \times P_C$ . As  $C$  has  $P_C$  color channels.

$$c = Ay \quad (4.1)$$

where  $A$  is matrix of  $[\alpha_i, \beta_i, \gamma_i]$  of size  $P_C hw \times Phw$  which apply on each element in  $y$ . For example, for a RGBW CFA as defined in Figure 4-1, the matrix  $A$  will be

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

where  $\otimes$  is the kron operator

Now  $x$  is the CFA image's matrix in multi colors which is  $M$  projection of  $c$ .  $M$  is  $hw \times P_C hw$  matrix that transforms  $c$  full multicolor image vector into  $x$  mosaicked image vector by choosing selectively according to spatial arrangement of colors.

$$x = Mc$$

$$x = MAy$$

See Figure 4-3 for the visual representation of the same. For the purpose of this study we have access to RGB images ( $Y$ ) and not images in the  $C$  domain. Therefore we convert RGB images into  $C$  colors images, sub-sample into CFA image and perform demosaicing on them.

As discussed in previous chapter, it is straightforward to write the demosaicing in case of multicolor CFA using the LMMSE approach with neighborhood. Let  $A_1$ , be  $A$  which applies on vectors build on a color image with neighborhood. We can write our demosaicing operator as follows:

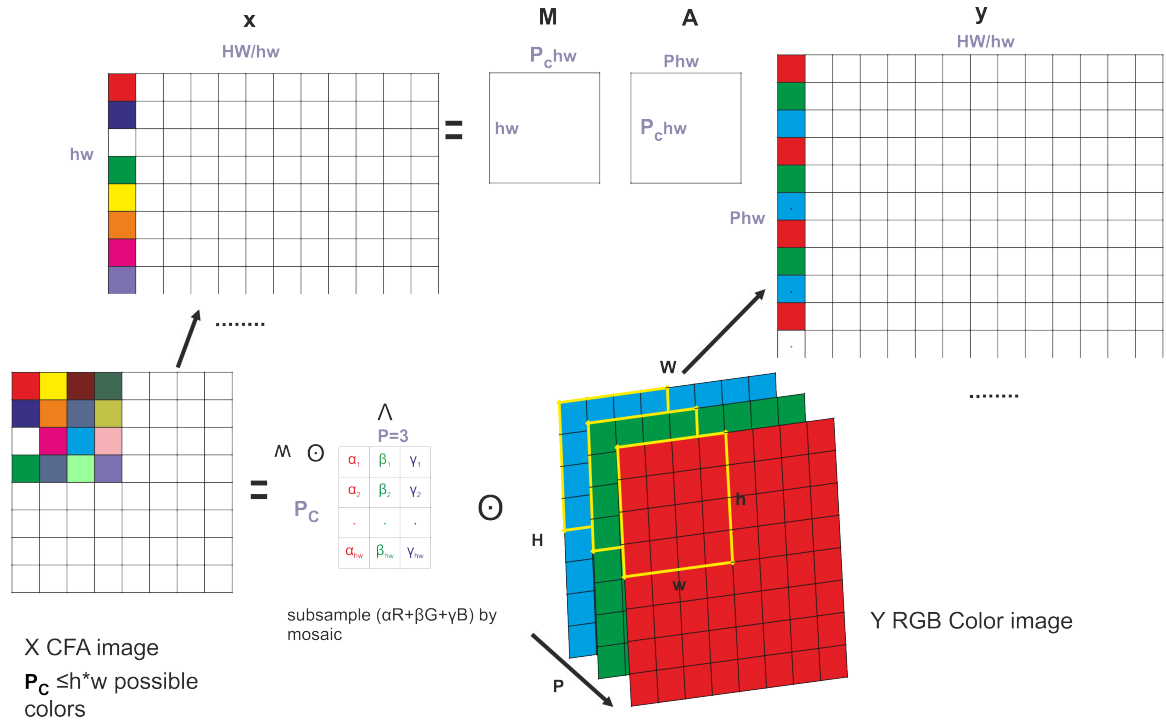


Figure 4-3: Matrix model of the multicolor CFA image formation without neighborhood

$$\begin{aligned}
 \hat{y} &= Dx_1 & (4.2) \\
 x_1 &= M_1 A_1 y_1 \text{ and } y = S_1 y_1 \\
 D &= S_1 R_{y_1} A_1^t M_1^t (M_1 A_1 R_{y_1} A_1^t M_1^t)^{-1}
 \end{aligned}$$

So starting from CFA image  $x$  in the artificial color domain  $C$  we can do demosaicing and directly recover a full RGB image. This is an advantage of considering  $C$  to be linear combination of RGB and not an arbitrary color.

### 4.3 Finding Optimum CFA arrangement

It is easy to imagine that a systematic evaluation<sup>70</sup> like in previous chapter is not possible due to sheer number of possible arrangements and also to evaluate  $MSE$

for each image in database is time consuming. However our model being linear we can express average  $MSE$  as a trace of matrix multiplication<sup>12</sup>. Considering  $y_i^{(k)}$  the  $k^{th}$  realization of the random color vector and  $\tilde{y}_i^{(k)}$  the corresponding vector in the reconstructed image, we can write average  $MSE$  as follows:

$$\begin{aligned}
MSE &= \frac{1}{KP_{hw}} \sum_{i=1..P_{hw}} \sum_{k=1..K} \left( \tilde{y}_i^{(k)} - y_i^{(k)} \right)^2 & (4.3) \\
&= \frac{1}{KP_{hw}} \sum_i \sum_k \tilde{y}_i^{(k)} (\tilde{y}_i^{(k)})^t - \tilde{y}_i^{(k)} (y_i^{(k)})^t - y_i^{(k)} (\tilde{y}_i^{(k)})^t + y_i^{(k)} (y_i^{(k)})^t \\
&= \frac{1}{P_{hw}} Tr(DM_1 A_1 R_{y1} A_1^t M_1^t D^t - DM_1 A_1 R_{y1} S_1^t - S_1 R_{y1} A_1^t M_1^t D^t + S_1 R_{y1} S_1^t)
\end{aligned}$$

where  $K = \frac{HW}{hw} N_{DB}$ .

The above term is independent of CFA image  $x$  and gives us an indicator of the whole performance directly from the cross correlation matrix  $R_{y1}$ . Therefore by evaluating this equation once we directly compute the  $MSE$  and therefore it is very fast compared to averaging explicitly on every images contained in the database. For reference, in Kodak database there are twenty four images. We used the Matlab's `fmincon` function using 'active-set' algorithm<sup>71-73</sup> to find the matrix  $A$  which gives the minimum of Average  $MSE$  for a given CFA size. One thing to understand is that unlike a system evaluation, the CFAs we find by using this approach (solving the optimization problem) are not the best CFAs but rather one of the better CFAs as the optimization process may converge to a local minima.

The above methodology was also used to validate the systematic evaluation of all RGB  $4 \times 4$  CFAs. In previous chapter for  $4 \times 4$  systematic evaluation only 'half-sized' Lighthouse image was used, where it took 9 days to compute. Now using the method of MSE by correlation  $R_{y1}$ , we could evaluate on all full sized 24 images of Kodak in 4 days computation time. We found the same  $4 \times 4$  CFA as from systematic evaluation as the top performer.

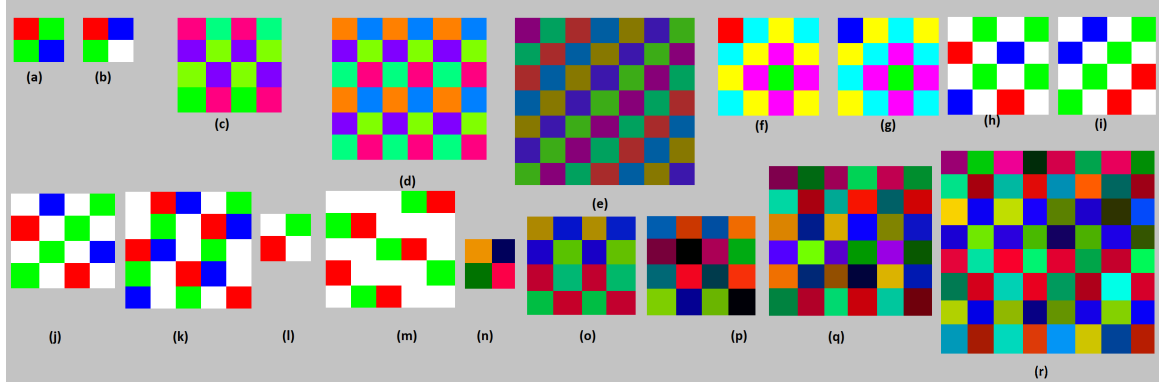


Figure 4-4: All CFAs. (a) Bayer, (b) RGBW, (c) HIRAKAWA<sup>35</sup>, (d) CONDAT<sup>66</sup>, (e) BAI<sup>13</sup>, (f) HAO 4a, (g) HAO 4b<sup>68</sup>, (h) YAMAGAMI<sup>74</sup>, (i) KODAK 2.0, (j) SONY RGBW, (k) HAO40, (l) HAO50, (m) HAO60<sup>14</sup>, (n)  $2 \times 2m$ , (o)  $4 \times 4m_1$ , (p)  $4 \times 4m_2$ , (q)  $6 \times 6m$ , (r)  $8 \times 8m$ .

### 4.3.1 Results

We used the above methodology to find the optimum multicolor CFA, see Figure 4-4 labeled as CFAs  $h \times w_m$ . We used our algorithm to test some state of the art multicolor CFAs. We also compared results of our algorithm to those reported by other authors for their respective CFAs<sup>13,35,67,68</sup>. The metrics we tested our algorithm for is Average PSNR  $\mu$ , defined as

$$\begin{aligned} \text{Average PSNR } \mu &= \frac{\sum PSNR}{N_{DB}} \\ &= (\sum (10 \log_{10} (\frac{1}{MSE}))) / N_{DB} \end{aligned} \quad (4.4)$$

We use a neighborhood of 10 for evaluating our algorithm. We leave a border equal to neighborhood size, i.e. 10 here. Also, we clip the images between [0 1]. Clipping has a real effect on PSNR. For instance  $\mu$  for Bayer CFA is 38.90dB for unclipped and 39.13dB clipped. We also tested color SSIM (average SSIM over RGB channel)<sup>51</sup>,  $\Delta E$ , variance of PSNR over RGB channel (lower value indicates all color channels are well reconstructed)  $\sigma_{rgb}$  and variance of PSNR across database  $\sigma$ .

In the Table 4.1, the first sub-part shows the comparison of the state of art CFAs with our LMMSE algorithm compared with the best state of art algorithms. For Hi-



Table 4.1: LMMSE for Kodak database. Other represents the value from the best state of the art algorithms known to us. <sup>1</sup> is LLSC<sup>27</sup>. <sup>2</sup> is LS Condat. <sup>3</sup> is Bai<sup>13</sup>. <sup>4</sup> is ACUDE<sup>14</sup>. Refer to Figure 4-4 for the CFAs

| CFA       | LMMSE        |               |             |                |             |      | Other                     |
|-----------|--------------|---------------|-------------|----------------|-------------|------|---------------------------|
|           | $\mu$        | SSIM          | $\Delta E$  | $\sigma_{rgb}$ | $\sigma$    | time | $\mu$                     |
| bayer     | 39.13        | 0.9913        | <b>1.40</b> | 4.85           | 6.22        | 0.33 | <b>41.46</b> <sup>1</sup> |
| hirakawa  | <b>40.45</b> | 0.9933        | 1.49        | 2.97           | 5.72        | 0.14 | 40.36 <sup>2</sup>        |
| condat    | <b>40.58</b> | 0.9938        | 1.49        | 1.18           | 6.23        | 0.10 | 40.11 <sup>2</sup>        |
| bai       | <b>40.77</b> | 0.9939        | 1.50        | 1.76           | 6.11        | 0.10 | 40.38 <sup>3</sup>        |
| hao4b     | <b>40.75</b> | 0.9938        | 1.52        | 1.47           | 5.78        | 0.14 | 40.73 <sup>4</sup>        |
| hao4a     | 40.49        | 0.9938        | 1.50        | 1.23           | 5.97        | 0.14 |                           |
| kodak2.0  | 38.43        | 0.9902        | 1.80        | 2.21           | 5.84        | 0.21 | <b>38.70</b> <sup>4</sup> |
| sonyrgbw  | 37.38        | 0.9882        | 1.95        | 3.54           | 5.66        | 0.13 | <b>38.10</b> <sup>4</sup> |
| hao40     | 38.66        | 0.9911        | 1.71        | <b>0.70</b>    | 5.64        | 0.11 | <b>38.93</b> <sup>4</sup> |
| hao50     | 39.07        | 0.9917        | 1.69        | 2.23           | 5.86        | 0.33 | <b>40.61</b> <sup>4</sup> |
| hao60     | 37.45        | 0.9884        | 2.17        | 7.67           | <b>5.32</b> | 0.10 | <b>37.51</b> <sup>4</sup> |
| RGBW      | 39.74        | 0.9926        | 1.59        | 1.89           | 5.69        | 0.33 |                           |
| yamagami  | 37.14        | 0.9874        | 1.99        | 3.96           | 5.93        | 0.13 |                           |
| 2x2 $m$   | 40.08        | 0.9930        | 1.54        | 1.68           | 6.40        | 0.32 |                           |
| 4x4 $m_1$ | 41.11        | <b>0.9944</b> | <b>1.44</b> | <b>0.72</b>    | 5.95        | 0.19 |                           |
| 4x4 $m_2$ | <b>41.12</b> | 0.9943        | <b>1.44</b> | 0.81           | 5.94        | 0.19 |                           |
| 6x6 $m$   | 41.09        | 0.9943        | <b>1.44</b> | 0.83           | 5.88        | 0.27 |                           |
| 8x8 $m$   | 41.09        | 0.9943        | 1.46        | 0.76           | 5.86        | 0.41 |                           |
| 10x10 $m$ | 40.51        | 0.9936        | 1.46        | 0.79           | <b>5.66</b> | 0.61 |                           |

rakawa, Condat, Bai and Hao 4b CFAs our method outperforms others. This shows the strength of LMMSE, which despite being generic it outperforms algorithms which were specifically designed for these CFAs. Then for CFAs with white pixels like Hao40, Hao50, Hao60, Sony RGBW and Kodak 2.0, ACUDE<sup>14</sup> is the best performer. We earlier mentioned the limitation of Bayer CFAs and requirement of computationally expensive algorithms to overcome that. Therefore we recommend multicolor CFA in the lower subpart of above table which show the best CFAs we found for size 2 to 10. We are able to achieve 41.12dB for the Kodak database, in less than 0.2s on Intel i7 6700. In terms of computational time required, 2x2 is slower than bigger CFAs, this is due to time required to divide the CFA image into more smaller pieces.

Table 4.2 shows the results for our algorithm on the McM database<sup>16</sup>. Figure 4-5 shows the crop of the fencing region of the Lighthouse image from the Kodak

Table 4.2: LMMSE for McM database. In Other best results were from the ACUDE<sup>14</sup>.

| CFA               | LMMSE        |               |             |                |             |      | Other                     |
|-------------------|--------------|---------------|-------------|----------------|-------------|------|---------------------------|
|                   | $\mu$        | SSIM          | $\Delta E$  | $\sigma_{rgb}$ | $\sigma$    | time | $\mu$                     |
| bayer             | 35.70        | 0.9830        | <b>3.35</b> | 7.96           | <b>8.99</b> | 0.20 | <b>36.38</b> <sup>4</sup> |
| hirakawa          | 35.22        | 0.9821        | 4.19        | 3.41           | 9.49        | 0.08 | 34.2 <sup>4</sup>         |
| condat            | <b>36.04</b> | 0.9851        | 3.99        | 0.94           | 9.23        | 0.06 | 35.42 <sup>4</sup>        |
| bai               | 35.24        | 0.9831        | 4.43        | 0.89           | 9.62        | 0.06 |                           |
| hao4b             | 35.63        | 0.9838        | 4.25        | <b>0.70</b>    | 9.62        | 0.08 | <b>35.64</b> <sup>4</sup> |
| hao4a             | 35.84        | 0.9845        | 4.14        | 1.36           | 9.13        | 0.08 |                           |
| kodak2.0          | 34.74        | 0.9803        | 4.38        | 1.60           | 9.27        | 0.12 | <b>35.15</b> <sup>4</sup> |
| sonyrgbw          | 34.46        | 0.9788        | 4.47        | 1.84           | 9.29        | 0.08 | <b>34.87</b> <sup>4</sup> |
| hao40             | 35.50        | 0.9832        | 3.96        | 1.40           | 9.06        | 0.07 | <b>36.21</b> <sup>4</sup> |
| hao50             | <b>35.72</b> | 0.9831        | 4.18        | 3.81           | 9.51        | 0.20 | 36.71 <sup>4</sup>        |
| hao60             | 34.64        | 0.9796        | 4.83        | 7.00           | 9.58        | 0.06 | <b>35.31</b> <sup>4</sup> |
| RGBW              | 35.86        | 0.9842        | 3.76        | 2.75           | 9.42        | 0.21 |                           |
| yamagami          | 34.55        | 0.9789        | 4.32        | 3.13           | 9.31        | 0.08 |                           |
| 2x2m              | 35.91        | 0.9845        | 3.77        | 4.63           | 9.47        | 0.20 |                           |
| 4x4m <sub>1</sub> | 35.90        | 0.9849        | 4.08        | 1.81           | 9.46        | 0.11 |                           |
| 4x4m <sub>2</sub> | 36.00        | <b>0.9852</b> | 4.03        | 2.27           | 9.40        | 0.11 |                           |
| 6x6m              | 35.71        | 0.9845        | 4.18        | 1.65           | 9.37        | 0.16 |                           |
| 8x8m              | 35.91        | 0.9849        | 4.14        | 1.86           | 9.27        | 0.24 |                           |
| 10x10m            | 35.64        | 0.9839        | 4.25        | 1.20           | 8.93        | 0.38 |                           |

database. For CFA size 4x4 and higher it is color noise free.

### 4.3.2 DFTs of proposed CFAs

Figure 4-6 shows the DFT (Discrete Fourier Transform) of the CFA image for the various CFAs proposed. It can be seen from that for CFAs proposed from size 4 to 10, it won't be possible to use frequency selection method to separate luminance and chrominance as they are heavily multiplexed. Still LMMSE provides good results with these CFAs.

## 4.4 Discussion

Some of the proposed CFAs have something like a dark pixel, a pixel with a very low sensitivity. Actually, with the LMMSE model the final value of a pixel depends not

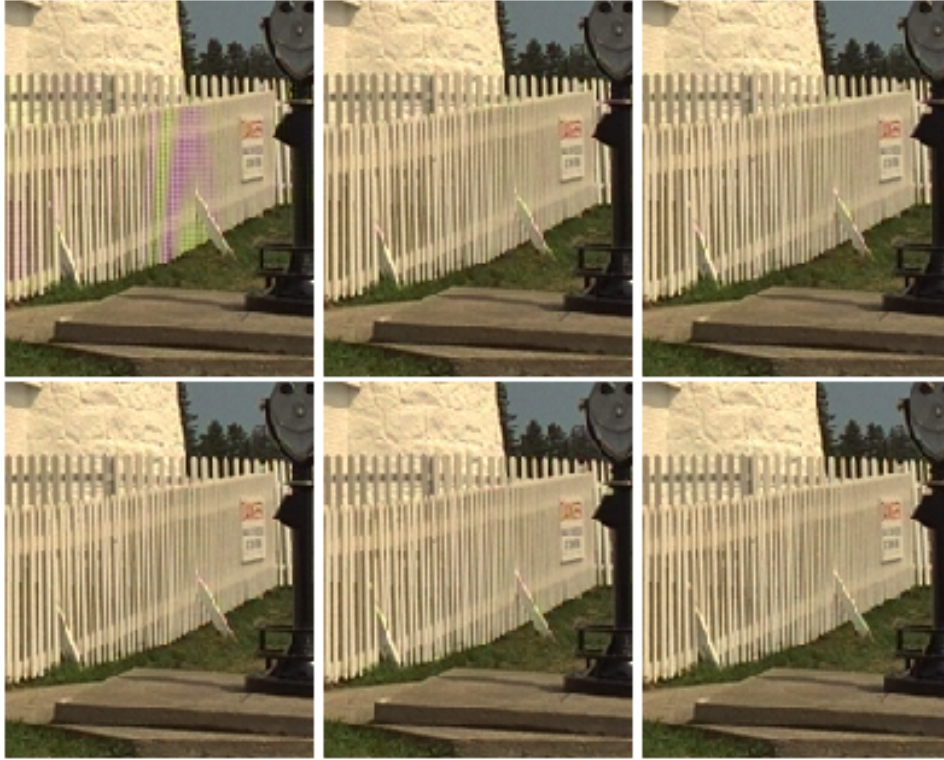


Figure 4-5: Crop of Lighthouse image for proposed  $2 \times 2m$ ,  $4 \times 4m_1$ ,  $4 \times 4m_2$ ,  $6 \times 6m$ ,  $8 \times 8m$ , and  $10 \times 10m$  CFAs

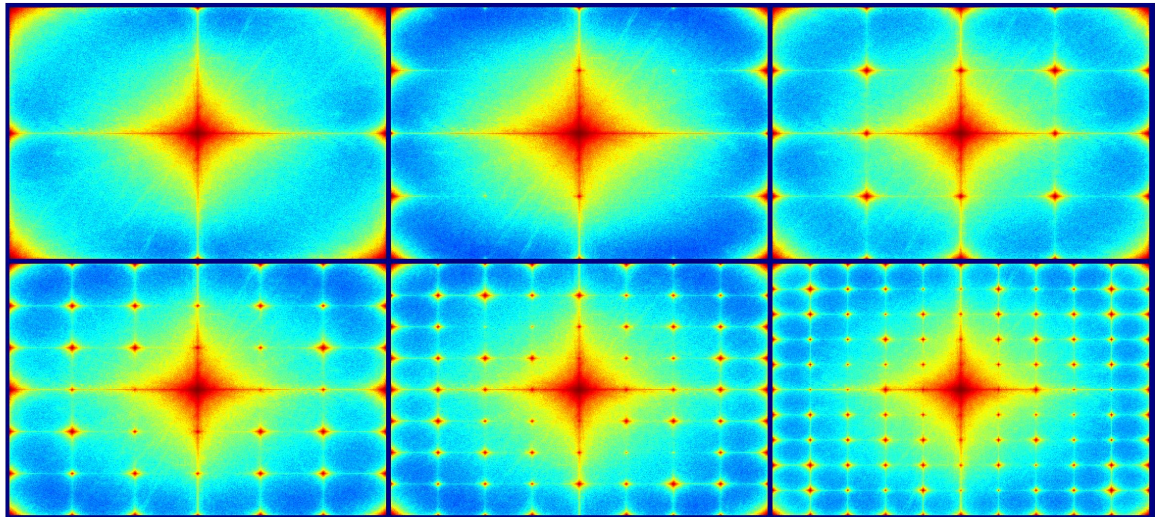


Figure 4-6: Frequency Response of proposed  $2 \times 2m$ ,  $4 \times 4m_1$ ,  $4 \times 4m_2$ ,  $6 \times 6m$ ,  $8 \times 8m$ , and  $10 \times 10m$  CFAs calculated using Discrete Fourier Transform

only on its own but also on its neighboring pixel. For instance for  $2 \times 2m$ , the pixel(1,2) is light blue having  $\gamma$  of 0.357. We can make it pure blue at 1 and we still get the

same  $\mu$ . Similarly for  $4 \times 4 m_2$ , we have two dark pixels. We can make pixel(2,2) green and pixel(4,4) as blue and we still have same  $\mu$ . The proposed CFAs are optimized ones; they are not necessarily the best ones. We start with a random CFA pattern and stop after a set number of iterations. We may continue the optimization process or choose a different random seed and get another random CFA which has equally good performance.

## 4.5 Conclusion

We presented a method to evaluate average MSE by directly evaluating correlation term on the image database along with matrices like  $M_1$ ,  $S_1$  particular to a given CFA. This is quite fast as compared to a complete evaluation involving individually demosaicing each image. It allows us to find the best filters by solving a gradient-descent optimization problem thus avoiding the need for a lengthy systematic evaluation. The proposed algorithm has the best performance to computational complexity of all the algorithms tested. The algorithm is generic and can be used for any random CFA unlike algorithms based on frequency selection or edge aware algorithms which are tuned to particular CFAs. The proposed CFAs have performance higher than 41.1dB which is amongst the best results in state of art. The proposed CFAs cannot be demosaiced by frequency selection method, therefore for random CFAs, LMMSE despite being linear is a good solution. The neighborhood compensates the sub-sampling by the mosaic by adding redundancy and improves the color reconstruction. We show that for CFAs like Hirakawa<sup>35</sup>, Condat<sup>66</sup>, Bai<sup>13</sup> and Hao4b<sup>68</sup> our method performed better than even frequency selection method which was in the first place used to propose these CFAs. It shows that LMMSE with neighborhood and learning on an image database is a powerful and generic tool which outperforms even customized solutions.



# Chapter 5

## Demosaicing for Spectral Filter Arrays (SFAs)

### 5.1 Introduction

In previous chapters we demonstrated solutions for demosaicing for color filter arrays (CFAs) and multicolor CFAs. By CFAs we refer to mosaic of Red, Green and Blue filters only and the term multicolor CFAs refers to color filters which are linear combination of these Red, Green and Blue filters. However, it is now possible to have filters with customisable spectral response, one can choose the bandwidth and peak wavelength according to application. It is also possible to implant these on a mosaic. We refer to these mosaics as Spectral Filter Arrays (SFAs). It is a more generic term and would also encompass CFAs. We use the term SFA instead of CFA when we have data for the filters spectral sensitivity available. CFAs were developed for retrieving color information from a scene whereas SFAs are no longer limited to that. For example, we have SFAs with Near-Infrared (NIR) channel.

SFA<sup>75</sup> provides, similarly to color filter arrays (CFA), a spatio-spectral, sparse description of an image of the scene. Spectral reconstruction addresses the recovery of the spectral information from multispectral data<sup>76</sup>, and will be dealt with in the next chapter. We focus on the reconstruction of the spatial information for each sensor spectral sensitivity, which is referred to as demosaicing.

SFA, such as most spectral imaging techniques, may be applied to different problems: high quality color imaging<sup>77</sup>, spectral reconstruction imaging<sup>76</sup> and generic computer vision<sup>78</sup>. Sensor design and data processing may be different depending on the application. According to our knowledge, the first work on filter array imaging dedicated to spectral imaging has been conducted by Ramanath *et al.*<sup>79,80</sup>. Since then, several works have been conducted to develop this concept<sup>29,81–85</sup>. Physical realization of SFA sensors are yet few. Although a limited number of companies started to commercialize the technology<sup>33,86,87</sup>, we identify only two prototypes embedded in actual cameras in the academic: One comes from the Le2i at Université de Bourgogne, Franche-Comté<sup>88</sup>, the other comes from Ukotomi & Tanaka Laboratory at the Tokyo Institute of Technology<sup>21</sup>. Interest of these realizations is that each of them comes with a database freely available online for research, SFA database spanning the visible and NIR<sup>89</sup> and TokyoTech Multispectral Image dataset<sup>1</sup>. SFA demosaicing is somewhat a new problem which potentially opened new application for color demosaicing such as designing new camera color space depending on the application. The drawback is that it is more complicated to manage. Relaxing the constraint of spectral density of the individual pixel increase the dimension of the problem. Hopefully, because many solution should give the same performance, approximation should be possible.

Most method of SFA demosaicing are dependent on the SFA pattern, which impairs their generality. Amongst the others, most require heavy processing or iteration, which breaks the interesting potential of SFA for real-time robotic applications. The LMMSE formulation provides a potentially very good candidate for real-time applications since, after training, it could be embedded into the camera hardware and perform real-time without losing the generality required by the different layouts present in the market.

Also, most of the works above have been evaluated on simulation. The challenge with real data is that there is no available ground truth to compare reconstruction with. Although in the case of color images, and potentially on spectral images in the

---

<sup>1</sup><http://www.ok.sc.e.titech.ac.jp/res/MSI/MSIdata.html>

visible, dedicated to accurate colorimetric rendering, we could consider psycho-visual ratings and rankings. When near infra-red (NIR) data are present, it is more difficult to evaluate the result and color images makes no complete sense.

One way to evaluate demosaicing would be to assess the quality of the data in a general case. We propose to use **no-reference image quality metrics** to evaluate the demosaiced image quality, and to couple the analysis with a usual simulation on hyperspectral radiance reference data set. No-reference image quality metrics have already been used in this context for color images by Gasparini *et al.*<sup>90</sup>, but according to our knowledge, not on spectral data. As a first step toward this methodology, we use gray-level, general quality metrics applied by band in this work.

In next Section, we develop the N-LMMSE demosaicing for spectral images. We then develop our experiment and evaluation procedure, which is based on real SFA images that span the visible and the NIR coming from prototype camera developed at University of Bourgogne, see Figure 5-6 which we shall refer as 'JB camera'. Results demonstrate that the N-LMMSE method permits to reconstruct better fine details and in particular text and high frequency, however, it exhibits more zipping effect. Another observation is that energy balance<sup>91</sup> plays a role in the learning/reconstruction process of the LMMSE, which impairs in particular the NIR channel due to illumination shift between learning and training. This suggests that a white balance must be performed before demosaicing. No-reference metrics seems not well adapted for the NIR information and show different behavior depending on the image content and demosaicing. Further works are required to evaluate the quality of this evaluation process.

## 5.2 Linear model for SFA demosaicing

### 5.2.1 Model of linear SFA image formation

In the previous chapter the image formation model presented required the availability of full resolution images for all color filters. However this is generally not true in the



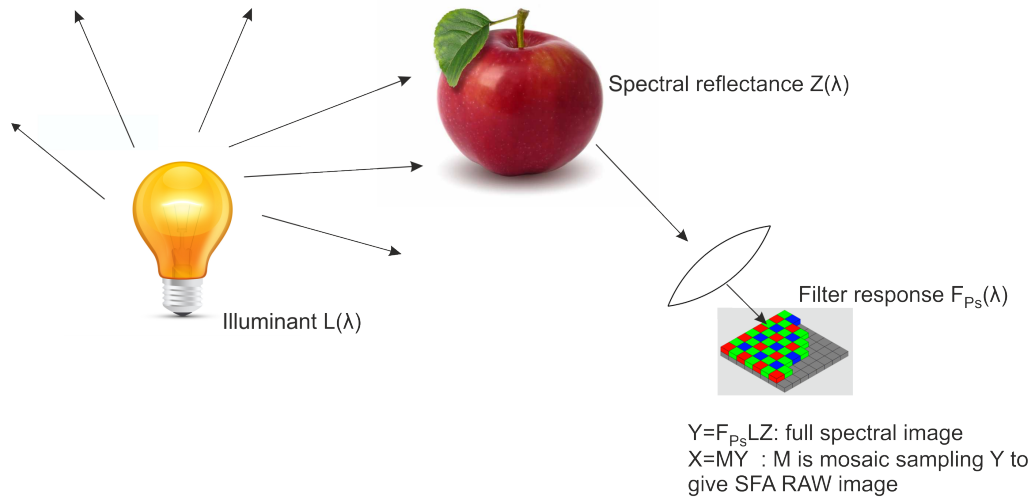


Figure 5-1: Image formation model from reflectance objects.

case of SFAs and neither is it practical to develop color-wheel kind of camera for every SFA. Therefore, we need a different model of image formation in the case of SFAs which model the same but from the physical attributes. A reflectance model of image formation, see (Figure 5-1) considers a light source that has a spectral power distribution  $L(\lambda)$ , which illuminates a scene that has a spectral reflectance  $Z(\lambda)$ . The reflected spectra then passes through a camera lens, is acted upon by the spectral response  $F_i(\lambda)$  of each spectral filter, where  $i = 1..P_s$  are number of different spectral filters overlaid on the sensor. The signal is integrated and for a pixel covered by filter  $F_i(\lambda)$ , gives  $\rho_i = \int F_i(\lambda)L(\lambda)Z(\lambda)d\lambda$ . If each spectral filter covered the entire surface of the sensor we would get a full resolution filtered image  $Y$ . However, in the case of SFA,  $Y$  is sampled by the filter mosaic  $M$  before hitting the silicon sensor to form the SFA raw image  $X$ . Figure 5-2 shows the entire process in the form of images. The goal of demosaicing algorithm is to estimate  $Y$  from  $X$ . Demosaicing algorithms exploit spatial and spectral correlations to guess the missing colors. Algorithms in the spatial domain adapted to Bayer-like layout estimate the *Green* channel first and then interpolate the other channels, by estimating the edges in horizontal or vertical direction. Alternatively in the frequency domain, luminance and spectral (alternatively chrominance in CFA) information are localized separately and it is a question of designing demosaicing filters to separate them<sup>37</sup>. However, for SFAs we

need to consider a generic spectral model as the spatial and spectral correlations for SFAs can be different than those for visible RGB camera. For RGB cameras filters are generally wideband and overlapping, spatial correlation is present in most of the bands. The SFAs may have narrow band spectral response and there maybe no overlap present, therefore the spatial/spectral correlation is different. Also, between visible and NIR filters, the spatial correlation is quite different due to difference in reflectance properties of objects for NIR. Therefore, we need a different class of algorithms to solve this problem.

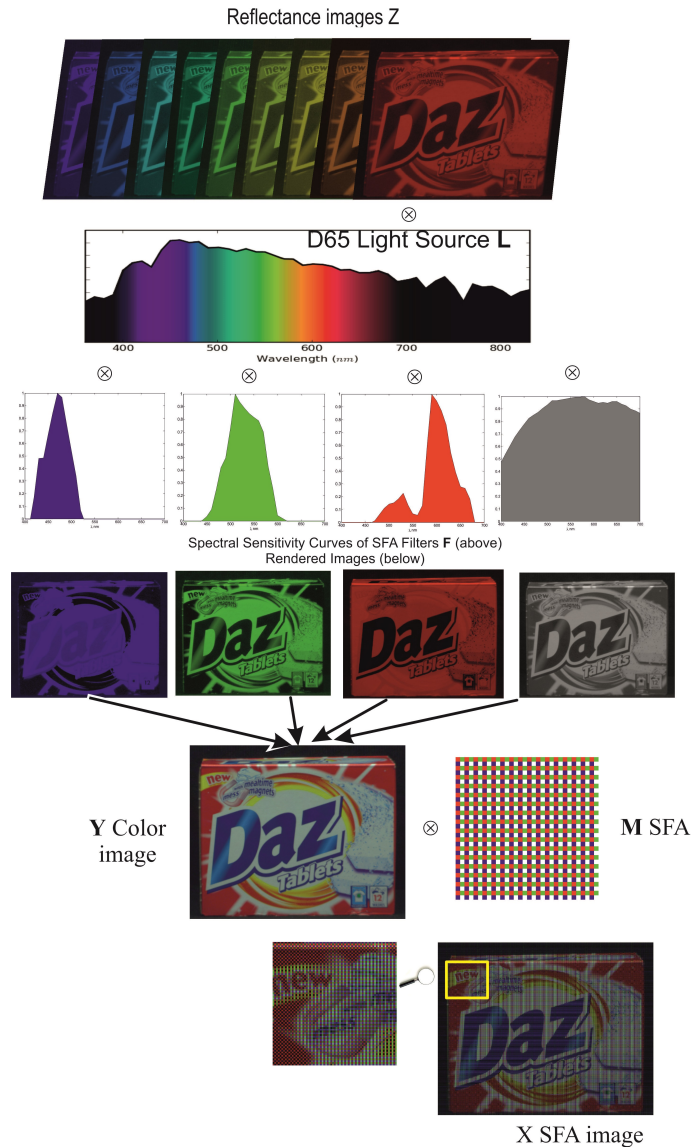


Figure 5-2: General image formation model illustrating the steps involved in getting the SFA image  $X$  from the reflectance images  $Z$ .  $\otimes$  is used to symbolize a mathematical operation for illustration purpose only.  $Z$  is of size  $H \times W \times P_\lambda$ , where  $P_\lambda$  varies along the wavelength  $\lambda$ . Reflectance images don't have any color, they are artificially colored here along  $\lambda$  for comprehension.  $L$  is a light source;  $F$  is spectral sensitivity curves of each color filter.  $L$  and  $F$  are defined along wavelength, not necessarily the same as for  $Z$ , so we may need to interpolate them to be of same dimension as  $P_\lambda$ , with same step size. Now  $L$  is of size  $P_\lambda \times 1$  and  $F$  is of size  $4 \times P_\lambda$  as we consider 4 color filters (Red, Green, Blue and White here). Now  $Z$  is multiplied by  $L$  then by  $F_r, F_g, F_b, F_w$  for filters corresponding to red, green, blue and white respectively, to obtain 4 rendered images. These 4 channels constitute  $Y$  color image which when operated upon by the mosaic  $M$ , gives  $X$  as SFA image.

In the previous chapters we showed that demosaicing can be considered as an inverse problem and we can choose a solution on the criteria of minimizing the mean square error between the  $Y$  and an estimate  $\hat{Y}$  derived from  $X$ . We showed that by considering a linear solution with neighborhood pixels (N-LMMSE) and training on a database of images one could get result comparable to non-linear solution in a more efficient manner. However earlier  $Y$  was limited to  $P = 3$  color channels but now it is now  $P_s$  spectral channels. Also, we need to account for the spectral image formation model described earlier which leads to the spectral N-LMMSE formulation described below.

### 5.2.2 N-LMMSE formulation

Previously we showed that one needs to unfold image matrices into column vectors to allow linear algebraic matrix multiplications. Now, we also need to unfold illuminant and spectral filter response as they are now part of the spectral image formation model. As earlier we choose to do so in a block-shift invariant manner as we need the same demosaicing operation on each basis pattern of filter arrangement. A discrete physical measurement leads to a discrete representation of image reflectance  $Z$ , and scene illumination  $L$  over the spatial resolution of the sensor and its spectral sensitivity  $F$  by bands. This product  $Y$  is then subsampled by the mosaic to simulate the SFA image  $X$ .

$X$  which is the SFA image is of size  $H \times W$  where  $H$  and  $W$  are the number of rows and columns in the sensor. Similarly  $Y$  which is full resolution spectral image of size  $HW P_s$  where  $P_s$  is the number of spectral filters present in the mosaic. If  $P_s$  spectral filters form the mosaic basis pattern of size  $h \times w$ , we can have utmost  $P_s \leq hw$ . Now this basis pattern is repeated across the surface of the sensor to form the SFA.

$Z$  is reflectance data typically defined for several wavelength bands  $P_\lambda$ ,  $Z$  has a size of  $HW P_\lambda$ .  $F$  contains the transmittance for the  $P_s$  filters defined over a range of wavelength,  $F$  is of size  $P_s \times P_\lambda$ .  $F$  is typically specified by the camera manufacturers or can be measured using a monochromator and spectro-photometer (Appendix C).

$L$  is the SPD of the light source defined for  $P_\lambda$  wavelength bands. We diagonalize this matrix, make it of size  $P_\lambda \times P_\lambda$  to enable matrix product. See (Figure 5-3).

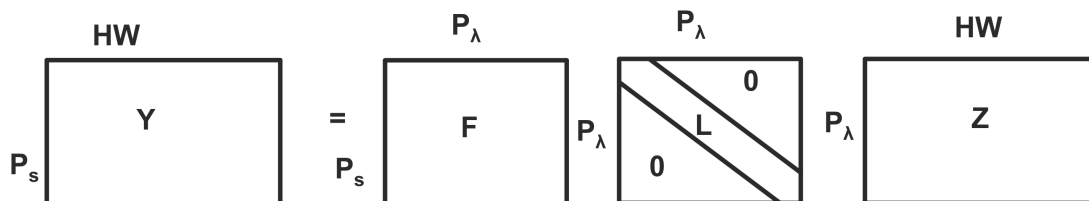


Figure 5-3: Illustration of image formation model on reflectance images. For sake of matrix multiplication we express 3D images  $Y$  and  $Z$  by unfolding them along  $HW$ . Applying mosaicing projection by SFA to  $Y$  will give us SFA image  $X$  of size  $1 \times HW$

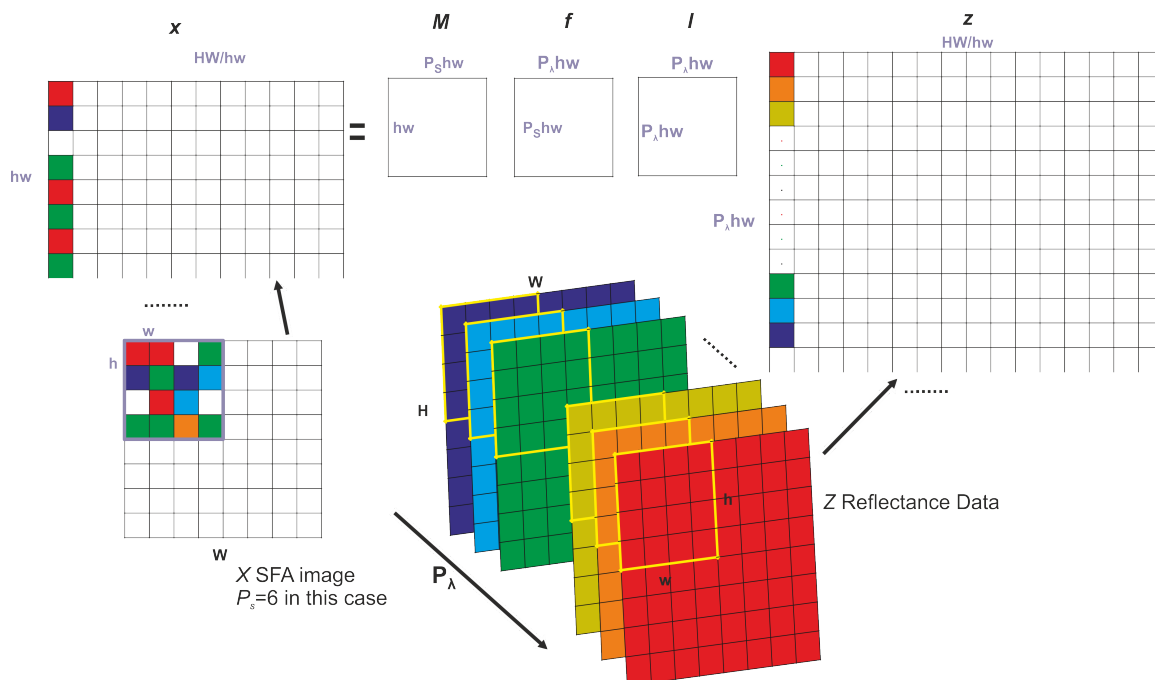


Figure 5-4: Linear matrix model of the SFA image formation without neighborhood. Reflectance images are not colored. Here they are artificially done to help the understanding of matrix unfolding.

The relation between full resolution  $Y$  and  $X$  is defined by multiplication by a projection matrix  $M$ . We consider the block shift invariant property of the mosaic and unfold accordingly. See (Figure 5-4).

Since the mosaic is composed of basis pattern of size  $hw$  we can unfold each basis pattern into a column vector. So  $Z$  can be unfolded into a matrix  $z$  of size

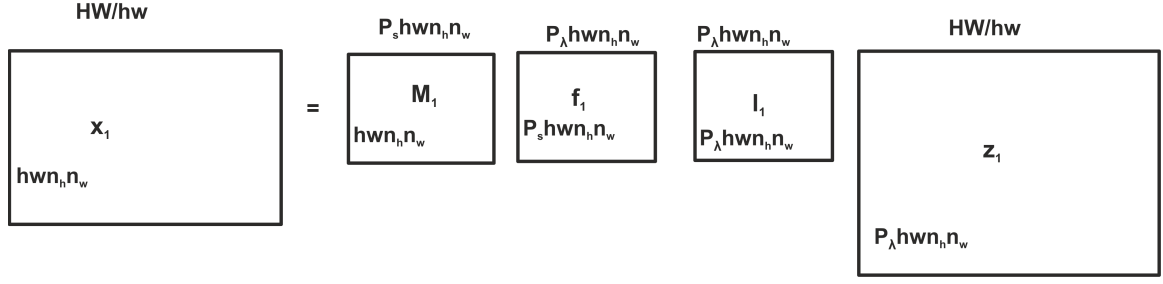


Figure 5-5: Illustration of linear matrix model of SFA image formation incorporating sliding neighborhood.

$P_\lambda hw \times HW/(hw)$ . Similarly for  $X$  which is unfolded to matrix  $x$  of size  $hw \times HW/(hw)$ . So  $y$  is unfolded color image in  $P_s$  channels of size  $P_s hw \times HW/(hw)$ .  $L$  is also replicated across  $hw$  to a matrix  $l$  of size  $P_\lambda hw \times P_\lambda hw$ .  $F$  is replicated to a matrix  $f$  of size  $P_s hw \times P_\lambda hw$ . Finally  $M$  is the projection matrix of size  $hw \times P_s hw$ .

Then we write:

$$\begin{aligned}
 y &= flz & (5.1) \\
 x &= My \\
 x &= Mflz
 \end{aligned}$$

In the description below we present matrices sizes considering sliding neighborhood. Let  $z_1$  be the matrix  $z$  considering a neighborhood window of size  $n_h \times n_w$  pixels around each pixel in the basis pattern. So  $z_1$  will be of size  $P_\lambda hwn_h n_w \times HW/(hw)$ . Similarly  $x_1$  is of size  $hwn_h n_w \times HW/(hw)$  for the sliding neighborhood. We also need to expand  $f_1$ ,  $l_1$  and  $M_1$  in similar fashion to incorporate neighborhood. See (Figure 5-5). It is also possible to design a matrix  $S_1$  which selects the central pattern in  $z_1$ . Then,

$$\begin{aligned}
\tilde{y} &= Dx_1 & (5.2) \\
x_1 &= M_1 f_1 l_1 z_1 \\
z &= S_1 z_1 \\
D &= flS_1 R_{z_1} l_1^t f_1^t M_1^t (M_1 f_1 l_1 R_{z_1} l_1^t f_1^t M_1^t)^{-1} \\
R_{z_1} &= \frac{1}{K} z_1 z_1^t
\end{aligned}$$

Above equation implies that for a given database of reflectance images we need to learn the correlation matrix  $R_{z_1}$  only once. We can then construct  $M_1$  and  $S_1$  for any given SFA arrangement, construct  $f_1$  for filter SPDs and  $l_1$  for any reference light to find the corresponding  $D$  matrix for demosaicing to recover the full resolution  $\hat{Y}$  image.

In the above description we used the sliding neighborhood to describe the unfolding. One can also use the constant methodology to achieve the same result by replacing  $hwn_h n_w$  with  $(h + n_h - 1)(w + w_h - 1)$  in the sizes of matrix sizes above for the constant method as described in the previous chapter.

## 5.3 JB Camera

### 5.3.1 Quantitative analysis in simulation

We simulate acquisition on the **SCIEN**<sup>2</sup> radiance image database<sup>19</sup>, following the model described above. We simulate acquisition by a real sensor, see (Figure 5-6) that spans visible and NIR<sup>88,92</sup>, where spatial layout follows Miao binary-tree<sup>29,93,94</sup>. The benchmark demosaicing, Miao binary-tree<sup>83</sup>, is applied on the raw data for reconstruction. Algorithms using guided filter and adaptive kernel<sup>21</sup> are tuned to SFAs having quincunx sampling of green so it would be non-efficient and impossible to compare with our instance of SFA. Also we considered the compressive sensing (Kronecker

---

<sup>2</sup><https://scien.stanford.edu/index.php/hyperspectral-image-data/>

and Group-Sparse)<sup>7</sup> approach, see Appendix D. The authors themselves state that this approach is more suited for random sampling of filters and not for uniform ones so it is not really fair to compare their method with our algorithm. Indeed, in physical implementations we usually do not have a totally random arrangement (entire sensor size) of SFAs. In testing on SCIEN images we found the performance of such methods (sPSNR ( $\mu_Y$ )) was worse than Miao binary-tree for our SFA instantiation. Also this

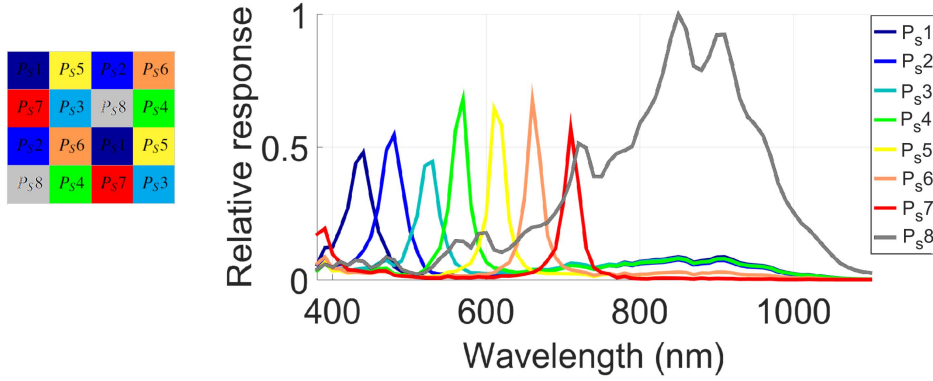


Figure 5-6: Spectral Filter Array(SFA) mosaic arrangement of filters & their spectral response for the JB camera

algorithm was 1400 times slower than N-LMMSE. For computer vision applications for which SFAs are particularly well suited and calculations on embedded systems, real time performance is paramount for which we find sparse based solutions are not suited. Therefore we choose not to present results from their method.

The N-LMMSE is trained and applied in a leave-one-out (LOO) manner on each of the images, while trained on the three others sequentially. We used a neighborhood of size 10 as it gives the best trade off between performance and computational complexity<sup>95</sup>. sPSNR ( $\mu_Y$ ) and sSSIM<sup>96</sup> are used to compare the reconstructed image to the full resolution image. Results of PSNR and SSIM for the benchmark are reported on Table 5.1, while results for the N-LMMSE are reported on Table 5.2. The D operator when applied to SFA images, might give us negative or greater than



1 output values. We do not apply any non negative constraints as of now and simply clip output values between 0 and 1. We observe a large gain of quality according to both PSNR and SSIM using N-LMMSE. We find that with Binary Tree we get an average sPSNR ( $\mu_Y$ ) value of 46.41dB while N-LMMSE gives us 53.74db, a gain of 7.33dB. The output images are also sharper. Similar average sSSIM goes up from 0.9959 to 0.9995.

In anticipation on the evaluation of real data, we also apply no-reference image quality metrics on each channel, considering that each channel is compliant with gray-level natural image statistics. The no reference metrics that we use on each band are supposedly representative of perceived visual quality. This is the most compact mean to perform the cumbersome task of evaluating perceived image quality on each channel. We selected arbitrarily (ad hoc choice after discussions with image quality experts) two metrics that appear to provide reasonably good estimate of quality in the general cases. The metrics are BRISQUE<sup>97,98</sup> and BLIINDS-II<sup>99,100</sup>. We used the implementations available at [LIVE](http://live.ece.utexas.edu/research/Quality/index.htm)<sup>3</sup>. The only no-reference metric dedicated to demosaicing<sup>90</sup> focuses on color images, and thus, unfortunately, was not meeting our purpose of spectral image evaluation. Results are reported on Table 5.1 and 5.2. Both metrics show better quality with smaller values. For both BLIIND-II and BRISQUE, N-LMMSE provides better results with BRISQUE following better the PSNR and SSIM trend.

If we look at the correlation coefficient for the binary-tree algorithm we see that both BRISQUE and BLIINDS-II metrics are highly correlated with PSNR and SSIM for all the spectral channels other than the NIR. This is counter intuitive as in general we expect a negative correlation, higher PSNR means lower BRISQUE for example. The NIR channel not following the same trends indicates that these metrics are not suitable for the same. For the N-LMMSE method the correlation coefficient we see negative or weak correlation for most of the channels which is more as per expectation.

---

<sup>3</sup><http://live.ece.utexas.edu/research/Quality/index.htm>

Table 5.1: Result of PSNR, SSIM, BLIIND-II and BRISQUE for reconstruction based on the benchmark binary tree demosaicing applied to the simulated images from SCIEN database. Column  $S_i, i = 1..8$  indicates the spectral channels the camera delivers.

| Scene name    | Average       | S1     | S2     | S3     | S4     | S5     | S6     | S7     | S8     |
|---------------|---------------|--------|--------|--------|--------|--------|--------|--------|--------|
| BLIIND-II     |               |        |        |        |        |        |        |        |        |
| SanFrancisco  | 43.50         | 42.50  | 44.50  | 46.00  | 32.50  | 49.00  | 47.50  | 48.50  | 37.50  |
| StanfordDish  | 38.38         | 39.50  | 42.00  | 40.50  | 29.00  | 40.00  | 36.50  | 46.50  | 33.00  |
| StanfordMem.  | 62.00         | 73.00  | 66.50  | 61.50  | 49.50  | 64.00  | 64.50  | 84.50  | 32.50  |
| StanfordTower | 33.13         | 33.00  | 34.00  | 39.00  | 24.50  | 32.50  | 31.50  | 36.00  | 34.50  |
| Average       | <b>44.25</b>  | 47.00  | 46.75  | 46.75  | 33.88  | 46.38  | 45.00  | 53.88  | 34.38  |
| BRISQUE       |               |        |        |        |        |        |        |        |        |
| SanFrancisco  | 74.08         | 74.95  | 75.23  | 71.24  | 70.51  | 84.62  | 82.80  | 81.11  | 52.17  |
| StanfordDish  | 67.04         | 67.29  | 67.24  | 65.55  | 64.07  | 73.14  | 72.63  | 74.34  | 52.05  |
| StanfordMem.  | 83.55         | 88.45  | 87.04  | 84.54  | 81.39  | 89.07  | 87.78  | 94.58  | 55.57  |
| StanfordTower | 66.91         | 67.49  | 67.59  | 63.32  | 62.66  | 74.59  | 73.73  | 71.64  | 54.22  |
| Average       | <b>72.89</b>  | 74.54  | 74.28  | 71.16  | 69.66  | 80.36  | 79.23  | 80.42  | 53.50  |
| PSNR          |               |        |        |        |        |        |        |        |        |
| SanFrancisco  | 45.74         | 54.77  | 53.75  | 52.89  | 51.88  | 54.64  | 54.80  | 57.39  | 37.31  |
| StanfordDish  | 44.30         | 53.75  | 53.19  | 52.55  | 51.97  | 55.83  | 55.99  | 57.88  | 35.69  |
| StanfordMem.  | 52.80         | 58.84  | 57.36  | 56.90  | 55.97  | 58.65  | 58.91  | 61.93  | 45.09  |
| StanfordTower | 42.79         | 51.87  | 50.92  | 49.77  | 48.71  | 51.82  | 52.01  | 54.28  | 34.37  |
| Average       | <b>46.41</b>  | 54.81  | 53.80  | 53.03  | 52.13  | 55.24  | 55.43  | 57.87  | 38.11  |
| SSIM          |               |        |        |        |        |        |        |        |        |
| SanFrancisco  | 0.9958        | 0.9970 | 0.9965 | 0.9957 | 0.9950 | 0.9971 | 0.9971 | 0.9981 | 0.9290 |
| StanfordDish  | 0.9954        | 0.9961 | 0.9959 | 0.9951 | 0.9945 | 0.9976 | 0.9976 | 0.9982 | 0.9229 |
| StanfordMem.  | 0.9985        | 0.9991 | 0.9989 | 0.9988 | 0.9985 | 0.9990 | 0.9990 | 0.9994 | 0.9881 |
| StanfordTower | 0.9938        | 0.9945 | 0.9940 | 0.9925 | 0.9911 | 0.9950 | 0.9951 | 0.9964 | 0.8932 |
| Average       | <b>0.9959</b> | 0.9967 | 0.9963 | 0.9955 | 0.9948 | 0.9972 | 0.9972 | 0.9980 | 0.9333 |

### 5.3.2 Analysis on real images

Real raw images from the same sensor are provided by the SFA<sup>4</sup> database<sup>89</sup>. We applied on the raw data the binary-tree benchmark demosaicing method and the N-LMMSE method trained, this time, on the four images of the SCIEN database. Reconstruction example on one image is respectively provided on Figures 5-7 and 5-8. We observe visually that we gain spatial resolution, images are sharper with less false colors, text is better reconstructed by the N-LMMSE method as compared with the binary-tree, however there are some zipper artifacts in visible channels and NIR channel has lots of artifacts. There we observe one of the limitation of the

<sup>4</sup><http://chic.u-bourgogne.fr/>

Table 5.2: Result of PSNR, SSIM, BLIIND-II and BRISQUE for reconstruction based on the N-LMMSE demosaicing applied to the simulated images from SCIEN database. We used a LOO method.

| Scene name    | Average       | S1     | S2     | S3     | S4     | S5     | S6     | S7     | S8     |
|---------------|---------------|--------|--------|--------|--------|--------|--------|--------|--------|
| BLIIND-II     |               |        |        |        |        |        |        |        |        |
| SanFrancisco  | 40.56         | 41.00  | 41.00  | 43.00  | 40.50  | 43.00  | 43.00  | 42.50  | 30.50  |
| StanfordDish  | 35.00         | 39.50  | 37.50  | 35.00  | 32.00  | 35.50  | 32.50  | 40.00  | 28.00  |
| StanfordMem.  | 51.50         | 58.50  | 52.50  | 50.00  | 30.50  | 61.00  | 59.50  | 72.00  | 28.00  |
| StanfordTower | 33.75         | 37.50  | 35.50  | 34.00  | 34.50  | 31.50  | 30.50  | 35.00  | 31.50  |
| Average       | <b>40.20</b>  | 44.13  | 41.63  | 40.50  | 34.38  | 42.75  | 41.38  | 47.38  | 29.50  |
| BRISQUE       |               |        |        |        |        |        |        |        |        |
| SanFrancisco  | 60.70         | 61.69  | 60.91  | 58.17  | 57.49  | 69.85  | 69.92  | 71.31  | 36.28  |
| StanfordDish  | 51.74         | 54.23  | 52.26  | 50.10  | 47.92  | 57.78  | 58.17  | 61.99  | 31.46  |
| StanfordMem.  | 64.84         | 72.77  | 68.88  | 65.72  | 62.26  | 70.06  | 70.20  | 79.91  | 28.94  |
| StanfordTower | 53.23         | 53.67  | 52.55  | 50.60  | 49.67  | 57.62  | 57.72  | 60.68  | 43.35  |
| Average       | <b>57.63</b>  | 60.59  | 58.65  | 56.15  | 54.34  | 63.83  | 64.00  | 68.47  | 35.01  |
| PSNR          |               |        |        |        |        |        |        |        |        |
| SanFrancisco  | 55.30         | 68.43  | 68.11  | 67.67  | 66.01  | 67.38  | 66.35  | 69.96  | 46.50  |
| StanfordDish  | 53.08         | 65.95  | 64.99  | 64.95  | 64.22  | 66.27  | 65.27  | 67.72  | 44.27  |
| StanfordMem.  | 54.63         | 64.71  | 61.92  | 62.05  | 60.40  | 63.31  | 64.21  | 67.56  | 46.20  |
| StanfordTower | 51.96         | 65.05  | 64.46  | 64.22  | 62.16  | 64.23  | 63.13  | 66.34  | 43.17  |
| Average       | <b>53.74</b>  | 66.04  | 64.87  | 64.72  | 63.20  | 65.30  | 64.74  | 67.89  | 45.04  |
| SSIM          |               |        |        |        |        |        |        |        |        |
| SanFrancisco  | 0.9996        | 0.9999 | 0.9999 | 0.9999 | 0.9998 | 0.9998 | 0.9998 | 0.9999 | 0.9915 |
| StanfordDish  | 0.9995        | 0.9998 | 0.9998 | 0.9998 | 0.9997 | 0.9998 | 0.9997 | 0.9998 | 0.9891 |
| StanfordMem.  | 0.9995        | 0.9998 | 0.9997 | 0.9996 | 0.9995 | 0.9997 | 0.9997 | 0.9999 | 0.9925 |
| StanfordTower | 0.9994        | 0.9997 | 0.9997 | 0.9997 | 0.9996 | 0.9997 | 0.9996 | 0.9998 | 0.9871 |
| Average       | <b>0.9995</b> | 0.9998 | 0.9998 | 0.9997 | 0.9997 | 0.9998 | 0.9997 | 0.9998 | 0.9901 |

methods that needs to learn compared to the interpolation methods in the sense that the changes of conditions and content between training and application could impact strongly the results. For purpose of training here we have only 4 images available, the training would be more general if we have bigger dataset of differing conditions available. Also, there is a difference in illumination of the test images (SCIEN Radiance data) which the model learns and the illumination in real images shot under a D65 simulator. Here, our result suggests that energy balance or spectral white balance<sup>101</sup> should be applied prior to demosaicing in the case of N-LMMSE method.

Worse results of N-LMMSE to the reconstruction of the NIR for the real images are due to statistics of real images which are artificial materials shot indoor not being



Figure 5-7: Binary tree results on the 8 spectral channels.



Figure 5-8: N-LMMSE results on the 8 spectral channels.

similar to those used for learning from the SCIEN database. If we look at the spectral response curves of the filters it is clear that the NIR filter higher sensitivity compared to other filters. If we simulate on SCIEN it translates into on average pixel values 10 times more compared to the 'deep red' filter and similarly for other filters. This is what the  $D$  operator learns. Whereas in the real images captured by the camera the NIR channel is only 5.8 times more sensitive so the demosaiced result is not good. Other channel maintain similar ratios in the simulated and real data so demosaicing works well for them.

Also, there is possibly another reason for the degrading NIR performance. Thomas

et al.<sup>88</sup> Appendix A, the authors have provided the spectral response of all the pixels on the sensor. We see a rather large variance in the NIR channel compared to other filters. What our model learns is an average response of these pixels, therefore there is a difference between what the model predicts and actual RAW images from the camera. For the zipper artifacts we have two issues, one is that the spatial resolution allowed by the SFA pattern is less which could limit the performance of our demosaicing operator. We also see there is variability in spectral response for even the color filters, which could also create artifacts even in visible channels. Generally for RGB cameras available commercially we don't have such intra channel variability and therefore we do not have such artifacts with N-LMMSE. Again these are hypothetical explanations.

Other demosaiced images show similar tendency and are available at [this link](#)<sup>5</sup>.

We apply the two no-reference metrics BLIIND-II and BRISQUE on each of the reconstructed channels, which are considered as gray-level images. Results are shown in Table 5.3 for the binary-tree and in Table 5.4 for the N-LMMSE. With the N-LMMSE approach the values for NIR channel are either zero as negative, which it is not clear how to interpret. However we know that these metrics may not be suitable for NIR channel. Similarly for the BLIIND-II some of the other spectral channels report zero value which is again unclear as to the relation with other channels. In general the values for both the metrics are lower for N-LMMSE compared to binary tree which implies that it is better at reconstructing the spectral channels. This can be collaborated with visual inspection of image channels. However at the moment the relation between reference and no-reference metrics is not so clear with respect to our evaluation on SCIEN images and more work is required to clearly identify their relation.

### 5.3.3 Visualization of color images

In order to visualize images and provide digest qualitative visual results, we visualize a colorimetric version of the images. Let us denote  $Q$  to be the XYZ color matching

---

<sup>5</sup><http://chic.u-bourgogne.fr/>

function. Let us denote XYZ to sRGB transform matrix as  $T_X$ . So

$$T_X = \begin{bmatrix} 3.2406 & -1.5372 & -0.4986 \\ -0.9689 & 1.8758 & 0.0415 \\ 0.0557 & -0.2040 & 1.0570 \end{bmatrix} \quad (5.3)$$

Let  $T_F$  be the transform that convert from Filter space to sRGB. The idea is that first we calculate a transform from FtoXYZ using pseudo inverse as  $QF(FF^t)^{-1}$ . This when multiplied with  $T_X$  gives us  $T_F$ . So matrix product of  $T_F$  with  $Y$  gives the  $Y_{sRGB}$ . Now gamma has to be applied to accordingly to the sRGB standards.

$$T_F = QF(FF^t)^{-1}T_X \quad (5.4)$$

Color versions of the database are shown in Figure 5-9 for binary-tree and in Figure 5-10 for the N-LMMSE. Note also that the color version of these images are different from<sup>89</sup> because a different color transform was used. Lapray et al.<sup>89</sup> fitted a linear transform based on reflectance measurement of the Gretag Macbeth Color Checker.

On these images, we first observe, that the N-LMMSE exhibits some spectral noise, graininess, which is more prominent than in the case of the binary-tree, which provides smoother images. We observe again that high frequencies are better reconstructed by the N-LMMSE. This is particularly clear on any of the texts and on the the ruler graduations. Color artifacts are also quite well avoided by the N-LMMSE, but closer look show zipping effect while binary-tree is preserved thanks to its directional interpolation aspect. Although the zipping effect is problematic however we have gained spatial resolution. Still it is important to remember that this is a sensor with 8 different filters with different relative sensitivities and in addition intra-pixel variability in spectral response.

Color visualization limits the influence of the NIR component, which is not very well reconstructed by the N-LMMSE, but still provide a general overview of the

performance.



Figure 5-9: Visualization of SFA images demosaiced by the binary-tree and rendered in sRGB space.

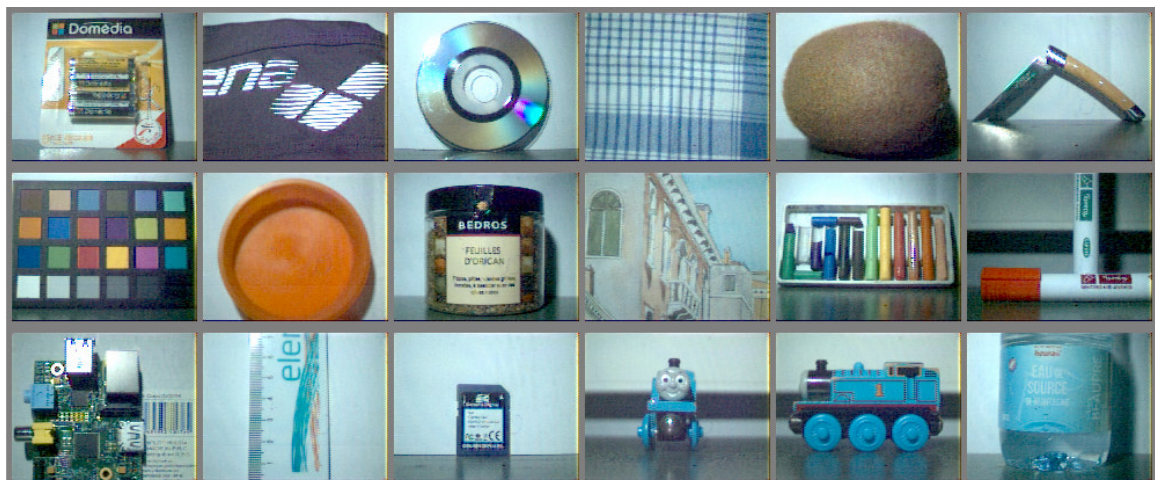


Figure 5-10: Visualization of SFA images demosaiced by N-LMMSE and rendered in sRGB space.

## 5.4 Conclusion

In conclusion, we have extended the N-LMMSE framework from the RGB color to the spectral domain. We found that N-LMMSE provides good results in terms of PSNR & SSIM for such SFA layout which we demonstrate on image simulated from SCIEN hyperspectral database and also validated on real RAW images. In particular,

it provides good reconstruction of text and high frequency components, but shows limited performance in the object edge, whereas the binary-tree provide a blurred but unzipped image. Looking at sPSNR ( $\mu_Y$ ) and sSSIM numbers of simulated SCIEN images we find a significant gain compared to binary-tree approach. Also N-LMMSE being a linear method is computationally simple compared to compressive sensing/sparse solutions and therefore is suitable for real time applications. We used no-reference metrics on each band which are supposedly representative of perceived visual quality of individual bands. It is yet difficult to understand the results of the no-reference metrics, which seem to depend greatly on the image content, in particular BLIIND-II, which may not be suitable for this evaluation.



Table 5.3: Result of reconstruction for the benchmark binary-tree applied to the real SFA images.

| Scene name     | Average      | S1    | S2    | S3    | S4    | S5    | S6    | S7    | S8    |
|----------------|--------------|-------|-------|-------|-------|-------|-------|-------|-------|
| BLIIND-II      |              |       |       |       |       |       |       |       |       |
| Battery        | 36.44        | 39.00 | 43.00 | 35.50 | 37.00 | 34.50 | 30.50 | 29.00 | 43.00 |
| Black Swimsuit | 33.50        | 36.50 | 36.00 | 34.50 | 32.00 | 31.50 | 33.50 | 20.50 | 43.50 |
| CD             | 31.63        | 33.00 | 39.50 | 40.50 | 34.00 | 30.00 | 25.00 | 13.50 | 37.50 |
| Kerchief       | 44.13        | 44.50 | 49.50 | 48.00 | 50.00 | 48.00 | 45.50 | 31.00 | 36.50 |
| Kiwi           | 29.88        | 26.50 | 29.50 | 28.00 | 30.50 | 32.50 | 38.00 | 15.00 | 39.00 |
| Knife          | 33.31        | 36.00 | 38.00 | 40.00 | 34.00 | 30.50 | 26.50 | 13.50 | 48.00 |
| Macbeth        | 32.44        | 34.00 | 33.50 | 34.00 | 29.50 | 28.00 | 32.50 | 25.00 | 43.00 |
| Orange         | 21.00        | 18.00 | 18.50 | 20.50 | 22.00 | 27.00 | 22.00 | 13.50 | 26.50 |
| Origan         | 33.31        | 31.00 | 33.00 | 35.00 | 32.00 | 30.00 | 35.00 | 31.50 | 39.00 |
| Painting       | 25.25        | 30.00 | 26.00 | 26.00 | 30.50 | 28.50 | 24.50 | 11.00 | 25.50 |
| Pastel         | 48.31        | 42.00 | 44.00 | 61.00 | 60.50 | 51.50 | 40.50 | 34.50 | 52.50 |
| Pens           | 50.19        | 45.50 | 47.50 | 50.00 | 48.00 | 54.50 | 51.00 | 37.50 | 67.50 |
| Raspberry      | 36.56        | 34.00 | 42.50 | 30.00 | 33.50 | 36.00 | 40.00 | 32.50 | 44.00 |
| Ruler          | 31.94        | 29.50 | 44.00 | 41.50 | 35.50 | 32.50 | 23.00 | 13.00 | 36.50 |
| SD             | 24.00        | 22.50 | 24.50 | 26.50 | 27.50 | 25.00 | 25.00 | 15.00 | 26.00 |
| Train Front    | 29.69        | 27.50 | 28.00 | 30.50 | 31.50 | 30.50 | 32.00 | 21.50 | 36.00 |
| Train Side     | 24.44        | 28.50 | 27.00 | 27.50 | 30.50 | 25.50 | 21.00 | 11.00 | 24.50 |
| Water          | 37.81        | 43.50 | 50.00 | 43.00 | 36.00 | 33.00 | 26.50 | 24.50 | 46.00 |
| Average        | <b>33.55</b> | 33.42 | 36.33 | 36.22 | 35.25 | 33.83 | 31.78 | 21.83 | 39.69 |
| Median         | <b>32.88</b> | 33.50 | 37.00 | 34.75 | 32.75 | 31.00 | 31.25 | 21.00 | 39.00 |
| BRISQUE        |              |       |       |       |       |       |       |       |       |
| Battery        | 57.54        | 59.50 | 60.29 | 59.63 | 56.06 | 56.22 | 55.46 | 61.14 | 52.06 |
| Black Swimsuit | 83.18        | 84.21 | 86.31 | 86.04 | 86.99 | 83.33 | 82.85 | 79.10 | 76.64 |
| CD             | 53.10        | 55.71 | 52.87 | 54.25 | 51.47 | 55.12 | 54.76 | 55.67 | 44.93 |
| Kerchief       | 59.65        | 45.00 | 61.96 | 82.39 | 76.24 | 55.49 | 47.47 | 64.06 | 44.56 |
| Kiwi           | 52.85        | 62.96 | 57.56 | 53.67 | 51.58 | 49.90 | 50.36 | 60.23 | 36.53 |
| Knife          | 45.86        | 51.43 | 47.69 | 46.18 | 50.95 | 50.40 | 38.93 | 40.29 | 41.05 |
| Macbeth        | 65.65        | 64.84 | 63.00 | 63.05 | 63.29 | 66.75 | 68.59 | 71.92 | 63.79 |
| Orange         | 64.00        | 65.67 | 67.81 | 66.51 | 65.90 | 64.74 | 66.33 | 65.36 | 49.73 |
| Origan         | 52.95        | 54.97 | 56.21 | 49.32 | 55.99 | 52.16 | 52.99 | 56.18 | 45.78 |
| Painting       | 46.95        | 52.76 | 47.64 | 50.26 | 48.71 | 40.80 | 48.55 | 51.72 | 35.13 |
| Pastel         | 71.07        | 70.75 | 70.64 | 73.38 | 71.62 | 72.65 | 73.56 | 68.04 | 67.94 |
| Pens           | 66.62        | 65.28 | 65.86 | 63.96 | 63.56 | 68.72 | 71.97 | 70.27 | 63.32 |
| Raspberry      | 60.96        | 63.29 | 65.18 | 61.14 | 62.82 | 58.70 | 57.51 | 57.30 | 61.78 |
| Ruler          | 53.24        | 52.77 | 50.07 | 52.51 | 52.69 | 55.08 | 58.70 | 61.21 | 42.85 |
| SD             | 52.00        | 52.53 | 50.04 | 53.80 | 52.50 | 50.88 | 53.86 | 57.46 | 44.92 |
| Train Front    | 64.58        | 66.37 | 62.80 | 63.65 | 63.94 | 66.66 | 70.23 | 70.52 | 52.50 |
| Train Side     | 70.23        | 72.06 | 71.79 | 69.10 | 66.36 | 72.37 | 73.84 | 74.01 | 62.29 |
| Water          | 57.22        | 61.28 | 57.25 | 58.41 | 55.92 | 55.83 | 59.95 | 61.15 | 47.99 |
| Average        | <b>59.87</b> | 61.19 | 60.83 | 61.51 | 60.92 | 59.77 | 60.33 | 62.53 | 51.88 |
| Median         | <b>58.60</b> | 62.12 | 61.12 | 60.38 | 59.44 | 56.02 | 58.11 | 61.18 | 48.86 |

Table 5.4: Result of reconstruction for the N-LMMSE applied to the real SFA images.

| Scene name     | Average      | S1     | S2     | S3     | S4     | S5     | S6     | S7     | S8     |
|----------------|--------------|--------|--------|--------|--------|--------|--------|--------|--------|
|                | BLIIND-II    |        |        |        |        |        |        |        |        |
| Battery        | 16.19        | 16.00  | 22.50  | 21.00  | 19.50  | 16.50  | 15.50  | 18.50  | 0.00   |
| Black Swimsuit | 20.56        | 12.50  | 29.50  | 17.50  | 22.50  | 37.50  | 27.00  | 17.50  | 0.50   |
| CD             | 17.38        | 18.50  | 28.00  | 25.50  | 20.50  | 19.00  | 15.50  | 12.00  | 0.00   |
| Kerchief       | 25.00        | 22.50  | 32.00  | 27.00  | 27.50  | 31.50  | 30.00  | 29.50  | 0.00   |
| Kiwi           | 14.88        | 3.00   | 13.00  | 8.50   | 12.50  | 25.00  | 28.00  | 29.00  | 0.00   |
| Knife          | 16.25        | 15.50  | 23.50  | 20.00  | 17.50  | 26.50  | 14.00  | 13.00  | 0.00   |
| Macbeth        | 8.00         | 2.00   | 11.50  | 9.00   | 1.50   | 13.50  | 9.50   | 17.00  | 0.00   |
| Orange         | 6.75         | 0.00   | 0.00   | 0.00   | 0.00   | 14.50  | 17.00  | 22.50  | 0.00   |
| Origan         | 11.56        | 9.00   | 16.50  | 14.50  | 10.50  | 9.00   | 13.50  | 19.50  | 0.00   |
| Painting       | 11.88        | 5.00   | 12.50  | 6.50   | 14.50  | 18.00  | 18.00  | 20.50  | 0.00   |
| Pastel         | 18.06        | 7.00   | 0.00   | 27.50  | 31.00  | 30.50  | 22.00  | 26.50  | 0.00   |
| Pens           | 19.00        | 6.50   | 24.50  | 18.00  | 18.00  | 29.50  | 27.00  | 28.50  | 0.00   |
| Raspberry      | 13.00        | 8.00   | 14.50  | 14.50  | 13.50  | 23.00  | 17.00  | 13.50  | 0.00   |
| Ruler          | 15.81        | 13.50  | 24.00  | 21.00  | 14.50  | 17.00  | 15.50  | 21.00  | 0.00   |
| SD             | 8.63         | 0.00   | 8.00   | 1.00   | 0.00   | 20.50  | 20.00  | 19.50  | 0.00   |
| Train Front    | 6.31         | 0.00   | 13.00  | 0.00   | 0.00   | 9.50   | 16.50  | 11.50  | 0.00   |
| Train Side     | 7.38         | 2.00   | 10.50  | 5.00   | 0.00   | 9.00   | 20.50  | 12.00  | 0.00   |
| Water          | 25.06        | 26.50  | 36.00  | 31.50  | 26.50  | 30.50  | 22.50  | 27.00  | 0.001  |
| Average        | <b>14.54</b> | 9.31   | 17.75  | 14.89  | 13.89  | 21.14  | 19.39  | 19.92  | 0.03   |
| Median         | <b>15.34</b> | 7.50   | 15.50  | 16.00  | 14.50  | 19.75  | 17.50  | 19.50  | 0.00   |
|                | BRISQUE      |        |        |        |        |        |        |        |        |
| Battery        | 12.37        | 19.42  | 20.52  | 16.95  | 14.89  | 14.48  | 10.37  | 13.86  | -11.50 |
| Black Swimsuit | 43.26        | 47.77  | 47.39  | 47.09  | 44.70  | 50.83  | 50.58  | 49.19  | 8.50   |
| CD             | 14.06        | 16.72  | 18.49  | 18.00  | 18.52  | 15.01  | 13.64  | 20.08  | -7.97  |
| Kerchief       | 45.85        | 44.04  | 65.06  | 64.98  | 73.07  | 52.84  | 44.73  | 43.89  | -21.79 |
| Kiwi           | 17.63        | 23.47  | 19.29  | 18.08  | 16.78  | 21.46  | 23.55  | 31.67  | -13.28 |
| Knife          | 15.56        | 17.24  | 21.29  | 19.05  | 16.26  | 18.35  | 18.80  | 18.92  | -5.44  |
| Macbeth        | 40.72        | 44.98  | 46.91  | 47.00  | 47.15  | 44.07  | 43.81  | 47.01  | 4.84   |
| Orange         | 139.23       | 138.36 | 156.53 | 157.11 | 157.34 | 157.14 | 151.30 | 121.47 | 74.59  |
| Origan         | 8.77         | 11.32  | 9.37   | 7.92   | 6.12   | 11.11  | 12.16  | 20.81  | -8.66  |
| Painting       | 21.43        | 27.17  | 32.33  | 31.29  | 30.56  | 26.62  | 23.03  | 20.59  | -20.13 |
| Pastel         | 32.97        | 39.40  | 43.47  | 39.19  | 36.56  | 33.58  | 31.91  | 34.11  | 5.51   |
| Pens           | 24.25        | 28.57  | 24.46  | 22.16  | 19.55  | 27.29  | 28.84  | 43.87  | -0.73  |
| Raspberry      | 14.69        | 17.63  | 17.57  | 17.28  | 17.61  | 18.00  | 18.12  | 22.56  | -11.29 |
| Ruler          | 14.30        | 18.15  | 22.54  | 23.93  | 25.07  | 19.57  | 16.79  | 19.08  | -30.73 |
| SD             | 23.09        | 21.72  | 29.28  | 30.45  | 36.06  | 30.17  | 29.09  | 21.04  | -13.09 |
| Train Front    | 29.61        | 35.44  | 35.06  | 34.82  | 34.19  | 31.86  | 31.45  | 40.45  | -6.39  |
| Train Side     | 18.18        | 30.04  | 26.49  | 23.06  | 18.16  | 12.82  | 18.41  | 32.48  | -15.99 |
| Water          | 20.44        | 22.05  | 23.15  | 24.06  | 24.86  | 26.76  | 25.69  | 27.51  | -10.61 |
| Average        | <b>29.80</b> | 33.53  | 36.62  | 35.69  | 35.41  | 34.00  | 32.90  | 34.92  | -4.68  |
| Median         | <b>20.93</b> | 25.32  | 25.48  | 24.00  | 24.97  | 26.69  | 24.62  | 29.59  | -9.63  |



# Chapter 6

## Spatio-Spectral Optimization of spectral filters transmission in the SFAs

### 6.1 Introduction

In the last chapter, we provide the general framework for demosaicing for any spectral filter array (SFA). We also demonstrated the results with an academic SFA solution. In this chapter, we will extend the framework by providing a methodology to optimize filter spectral sensitivity of transmission based on different applications, like demosaicing, spectral reflectance recovery, selection of filters for color-wheel, etc. For the purpose of learning our demosaicing operator we consider hyperspectral image databases like Finlayson (A-3), Cave (A-4), and SCIEN (A-5). Finlayson and Cave have data from 400nm to 700nm, visible range. While SCIEN has data in the NIR range also. Finlayson image database is particularly interesting because it has the Macbeth ColorChecker Chart in good resolution which is useful to check color rendering. Finlayson and Cave have scenes of mostly objects shot in a studio setting. SCIEN has both landscape images and studio portraits. We considered only landscape images shot without the polarizing filter as they are representative of natural

scenes. Previously we limited our discussion of demosaicing to recovering the spectral channels only. Arad et al. presented a method based on sparse dictionary<sup>102</sup>, used to estimate hyperspectral images from a single RGB image as input. In this chapter, we will investigate if we could do the same directly from the SFA image using LMMSE.

First, we will provide trends on various performance metrics based on gaussian shaped filters and SFA size. Finally, we provide a set of optimized filters.

## 6.2 Image formation workflow and quality metrics

To define the image quality metrics one needs to first describe the various pathways possible to determine sRGB images from hyperspectral images and its rendering through spectral filters of a camera. Figure 6-1 shows the image formation model from reflectance images to the final sRGB images. We recall from previous chapter that a light source  $L$  illuminates an object of reflectance  $Z$  to give Radiance image  $RI$ . Now, for the sake of defining the performance metrics we can define three pathways to obtain final sRGB images. We consider sRGB color space as digital images are primarily for display purposes and we need to select a particular color space. It could even be AdobeRGB, ProPhoto RGB or any other however sRGB is most commonly used. There are three possible pathways.

1. **Radiance to sRGB:** We need to multiply  $RI$  radiance image with the XYZ color matching function defined as  $Q$  to get the corresponding XYZ coordinate image  $RI_{XYZ}$ . This we transform from XYZ to sRGB by matrix  $T_X$  (equation 5.3) which gives us the  $RI_{sRGBlinear}$ , i.e. linear sRGB values from Radiance image. We need to apply gamma to get final sRGB values, i.e.  $RI_{sRGB}$ . The following equation defines the gamma transformation as specified in the sRGB standard for an image  $I_{sRGBlinear}$ .

$$\begin{aligned} I_{sRGB} &= 12.92I_{sRGBlinear}, I_{sRGBlinear} \leq 0.0031308 \\ &= (1 + 0.055)I_{sRGBlinear}^{\frac{1}{2.4}} - 0.055, I_{sRGBlinear} > 0.0031308 \end{aligned} \quad (6.1)$$

We used the above equation to transform linear values in sRGB to reach final sRGB values.

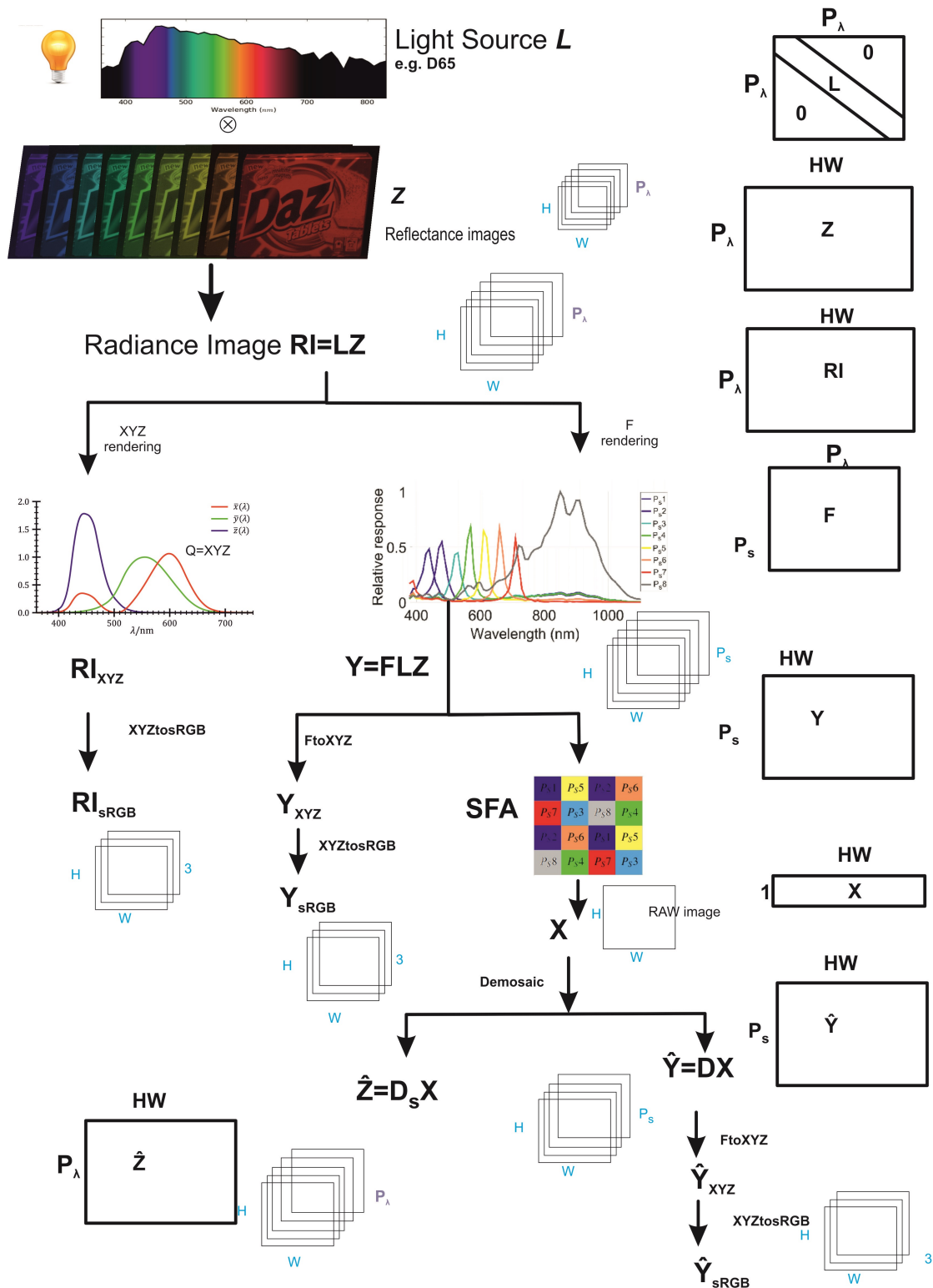


Figure 6-1: Image formation model from reflectance objects with the workflow to get sRGB images

2. **Fully sampled Filter space to sRGB:** Consider an imaging system where all spectral filters are captured in full resolution, for instance a color wheel arrangement or split prism with several sensors. We can also simulate such a capture from hyperspectral database. Let  $F$  be the combined spectral response of imaging pipeline. This when multiplied with the radiance image gives us the full resolution  $Y = FLZ$  spectral image. Let  $T_F$  (equation 5.4) be the transform that convert from Filter space to sRGB color space. Further gamma is applied to get  $Y_{sRGB}$ .

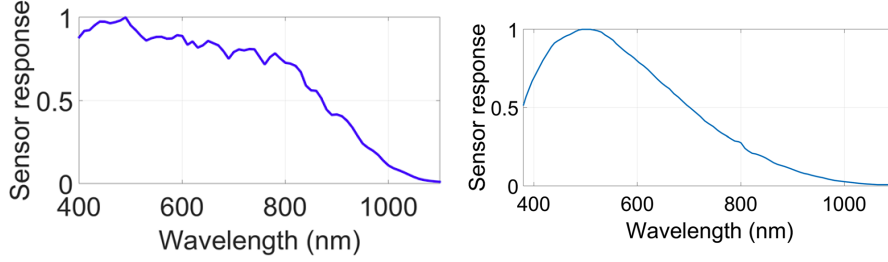
3. **SFA Raw image to sRGB:** This is the RAW image captured by a sensor overlaid with a mosaic  $M$ . Subsampling full spectral image  $Y$  with the SFA gives us the Raw image  $X$ . By multiplying  $X$  with demosaicing operator  $D$  as described in previous chapters we obtain an estimate of  $Y$  as  $\hat{Y}$ . We can further apply  $T_F$  and gamma to get the sRGB version  $\hat{Y}_{sRGB}$ . Similarly we can also calculate another operator  $D_s$  described later which can give an estimate of reflectance  $\hat{Z}$ .

We are talking in terms of images here, however to enable matrix multiplication we need to transform these images into column vectors. Similarly, to ensure good performance of demosaicing we need to incorporate neighborhood in the decomposition and also to account for neighborhood in the constructing of matrices like  $T_F$  and  $Q$ .  $F$  is usually specified in terms of Transmittance,  $L$  in relative SPD (spectral power distribution) which might not be normalized. For sake of calculation we normalize both between 0 and 1. Now  $Y_{XYZ}$  and  $\hat{Y}_{XYZ}$  can be transformed into the Lab domain and to ensure that it matches the range of Lab values from direct transformation of  $RI_{XYZ}$  we need to normalize the  $FtoXYZ$  and eventually  $T_F$ . The idea is to calculate the maximum value of the sum of the  $T_F$  as the factor  $fac$ . If the values  $fac$  is less than 1, e.g. for white filters, than we make it equal to 1. Now we divide  $FtoXYZ$  by this  $fac$ . So simultaneously  $T_F$  is also divided by the same factor.

### 6.2.1 Note about filter response $F$

When we describe  $F$  above we talked in terms of combined response of the entire imaging pipeline (objective, spectral filters transmittance and silicon response). However,

we can also break it in terms of its individual components. Let  $G$  be the spectral response of the silicon sensor. Now the spectral response depends on the material (Silicon / InGaAs), thickness of the material, temperature.<sup>103</sup> Also the kind of technology employed, CCD, CMOS, NMOS, backlit, front illuminated, etc. have an impact on the sensor response. Figure 6-2a, 6-2b shows the sensor response for two particular silicon sensors<sup>91</sup>.



(a) Sensor response for Silios sensor (b) Sensor response for Sony IMX174

Figure 6-2: Sensor spectral response for two different Silicon based sensors

These sensors are overlaid with color filters having a transmittance function  $F_t$  which is then modulated by the silicon response. The camera records a value which is a product  $F_t$  and  $G$ . For the purpose of optimization of filters one has to consider a particular sensor. Due to variability in the sensor response and for the purpose of simplicity here we consider an ideal sensor having a flat ( $=1$ ) spectral response. The idea is to discuss below the methodology for optimization and not to propose the ideal filters. From now on, when we write  $F$  we mean the combined response of sensor, the filters and also the objective. Utilization of Faber-Perot interferometers<sup>91</sup> provides great flexibility in design of spectral filters. Lapray et al.<sup>91</sup> demonstrate that these filters can be approximated by use of a Gaussian model. According to gaussian model the filter can be defined in terms of three parameters  $\lambda_\mu$ , the wavelength of peak sensitivity, the standard deviation  $\lambda_\sigma$ , i.e. the spread of the filter and finally the amplitude  $A_m$ , the intensity factor.

$$F(\lambda) = \frac{A_m}{\lambda_\sigma \sqrt{2\pi}} e^{-\frac{(\lambda - \lambda_\mu)^2}{2(\lambda_\sigma)^2}} \quad (6.2)$$

Before describing the optimization process for filters we will discuss the effect of



these parameters of the gaussian filters in terms of the image quality and demosaicing performance.

## 6.2.2 Performance metrics

According to the description above of the possible pathways to render sRGB images from the hyperspectral database, directly or rendering through spectral filters or demosaicing first and then rendering. Let us define the following performance metrics. All the metrics below except for  $\mu_{z(chart)}$  are averages across the entire image database. Figure 6-3 provide a visualization of the key metrics.

1.  $\mu_Z$ : The PSNR between Reflectance image  $Z$  and its estimate  $\hat{Z}$ . We calculate PSNR by first calculating the MSE between all the pixels for all the wavelengths. Its helps to deduce which filters layout and spectral response are relevant if the application is recovery of spectrum.
2.  $\mu_{Z(chart)}$ : The PSNR between the reflectance and its estimate of the 24 color patches in the Macbeth chart image of the Finlayson database. We take a region of 10x10 pixels on each color patch and average its value. So we get 24 reflectance values and calculate the PSNR for these 24 values. See Figure 6-11 which shows each region of 10x10 pixels on the Macbeth color chart. It is quicker to estimate and uses Macbeth Colorchecker chart which is a standard.
3.  $\mu_Y$ : The PSNR between fully sampled spectral image  $Y$  and its demosaiced version  $\hat{Y}$ . This is relevant for computer vision applications as we can compare all the recovered channels. In this model, we are not limited to just color applications. Filters could well be UV (Ultraviolet) or NIR (Near Infrared) in nature.
4.  $\mu_1$ : The PSNR between the sRGB version derived directly from radiance  $RI_{sRGB}$  and  $Y_{sRGB}$ . Primarily it captures the error due to the limit imposed due to filters and the  $T_F$  transform. Metric of interest if the end application doesn't involve demosaicing, e.g. color wheel camera.

5.  $\mu_2$ : The PSNR between the sRGB version derived from  $Y_{sRGB}$  and its demosaiced version  $\hat{Y}_{sRGB}$ . Primarily it captures the error due to demosaicing. This metric is also of interest if one was to compare different demosaicing algorithms.
6.  $\mu_3$ : The PSNR between the sRGB version derived from Radiance  $RI_{sRGB}$  and demosaiced version  $\hat{Y}_{sRGB}$ . Primarily it captures the error due to both demosaicing and the  $T_F$ , Filter space to sRGB color space transform. In a way its a combination of both  $\mu_1$  and  $\mu_2$ , however the relation is not simply linear. This metric is most useful as it tells us which filter layout and spectral properties gives us the best demosaicing and accurate color rendering.
7.  $SSIM_1, SSIM_2, SSIM_3$ : Similarly to the above three  $\mu_{1,2,3}$ , they defined the SSIM of the sRGB images  $RI_{sRGB}, Y_{sRGB}$  and  $\hat{Y}_{sRGB}$ .
8.  $\Delta E_1, \Delta E_2, \Delta E_3$ : Similarly to the above three  $\mu_{1,2,3}$ , they define the Euclidean distance in Lab version of the sRGB images  $RI_{sRGB}, Y_{sRGB}$  and  $\hat{Y}_{sRGB}$ . Further we use them to denote the average across all images in a given database. Where

$$\Delta E = \sqrt{(L_2 - L_1)^2 + (a_2 - a_1)^2 + (b_2 - b_1)^2} \quad (6.3)$$

where  $(L_1, a_1, b_1)$  and  $(L_2, a_2, b_2)$  are CIE Lab coordinates for two colors. Now to get Lab coordinates one has to first pass through XYZ coordinates as follows

$$\begin{aligned} L &= 116f\left(\frac{Y}{Y_n}\right) - 16 \\ a &= 500\left(f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right)\right) \\ b &= 200\left(f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right)\right) \end{aligned}$$

where  $X_n, Y_n$  and  $Z_n$  are XYZ values of reference whitepoint (6.4)

$$f(t) = \begin{cases} \sqrt[3]{t} & \text{if } t > \delta^3 \\ \frac{t}{3\delta^2} + \frac{4}{29} & \text{otherwise} \end{cases}$$

$$\delta = \frac{6}{29}$$

With reference to above equation  $X, Y, Z$  refer to  $XYZ$  color space values, not to confuse with the image notation used otherwise in this thesis. Similarly for  $L$ .

Ideally, if the application is accurate color imaging, we are looking for high  $\mu_3$  and low  $\Delta E_3$ . If the application is less false colors, artifacts and moire we are looking for higher  $\mu_2$  with a  $\Delta E_3$  within tolerance limits. As determination of above metrics requires an evaluation of each metric individually on each image of the database, demosaicing and processing it is time consuming. Therefore we define an approximation metrics of above metrics (except SSIM), denoted by prefix 'm' (e.g.  $m\mu_Z$  for  $\mu_Z$ ), which can be derived directly from the cross correlation matrix  $R_{z1}$ . In the next section, we determine these equations which are then used for calculating the optimized filters.

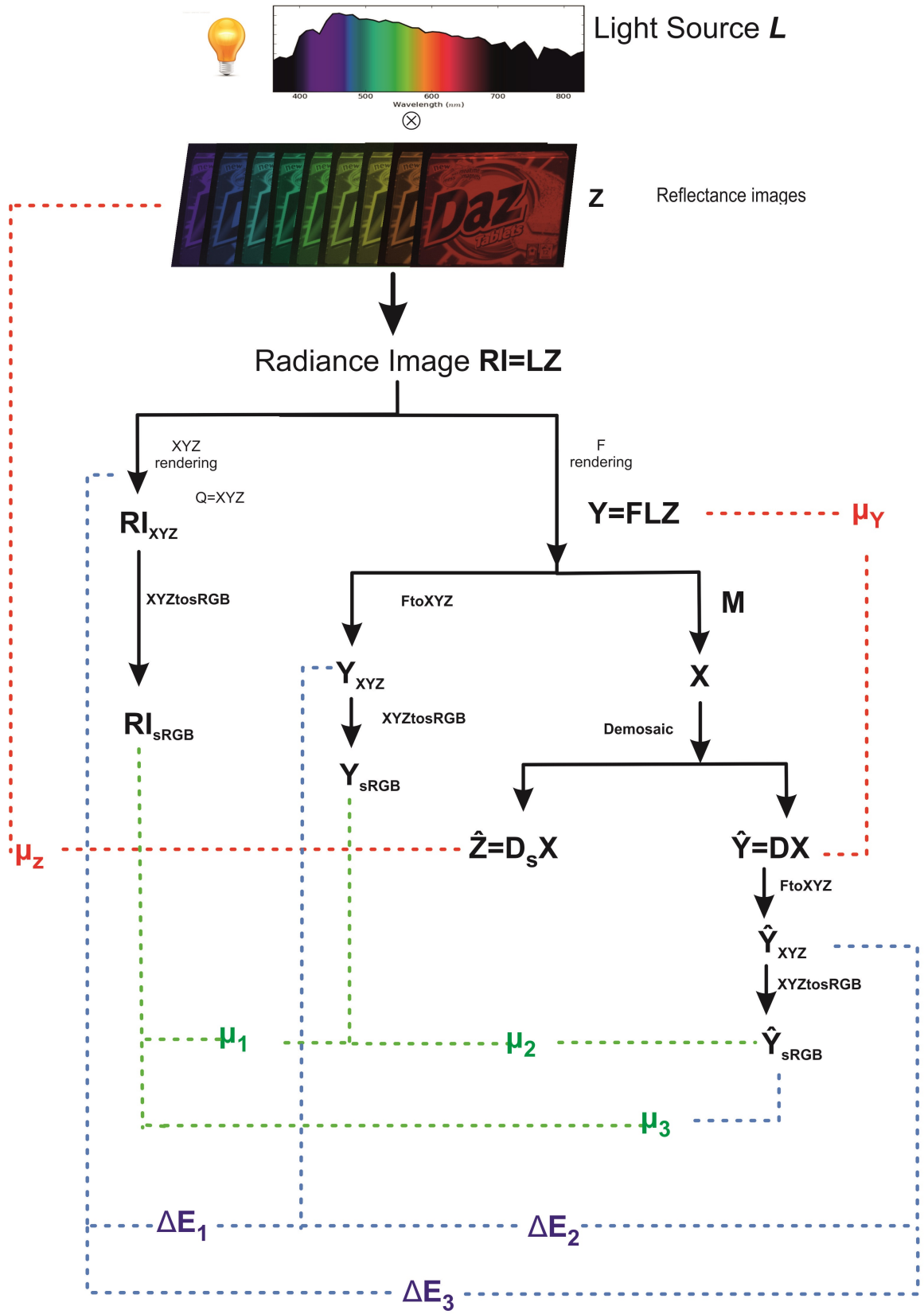


Figure 6-3: Image formation model illustrating the different parameters for optimization

### 6.2.3 Parameter for Optimization by Correlation matrix

Let us recall from last chapter that  $x_1 = M_1 f_1 l_1 z_1$ ,  $z = S_1 z_1$ ,  $y = flz$  that it can be rewritten as  $y = flS_1 z_1$ . Also  $\hat{y} = Dx_1$  where  $D$  is defined as below:

$$D = flS_1 R_{z_1} l_1^t f_1^t M_1^t (M_1 f_1 l_1 R_{z_1} l_1^t f_1^t M_1^t)^{-1} \quad (6.5)$$

$$D = flD_s$$

$$D_s = S_1 R_{z_1} l_1^t f_1^t M_1^t (M_1 f_1 l_1 R_{z_1} l_1^t f_1^t M_1^t)^{-1} \quad (6.6)$$

where  $R_{z_1}$  is defined as  $R_{z_1} = \frac{1}{K} z_1 z_1^t$ .

MSE or PSNR is parameter of choice for determining the image quality of demosaiced image. We consider the average of PSNR for all images in Finlayson database to ensure our result is robust. As shown in the previous chapter it is more judicious to use the average MSE because it could be expressed directly from the parameter of the problem. Following the same procedure as previous chapter we can write:

$$m\mu_Y = \frac{1}{P_s h w} \text{Tr}(D M_1 f_1 l_1 R_{z_1} l_1^t f_1^t M_1^t D^t - D M_1 f_1 l_1 R_{z_1} S_1^t l^t f^t) \quad (6.7)$$

$$- f l S_1 R_{z_1} l_1^t f_1^t M_1^t D^t + f l S_1 R_{z_1} S_1^t l^t f^t$$

$$m\mu_Z = \frac{1}{P_\lambda h w} \text{Tr}(D_s M_1 f_1 l_1 R_{z_1} l_1^t f_1^t M_1^t D_s^t - D_s M_1 f_1 l_1 R_{z_1} S_1^t) \quad (6.8)$$

$$- S_1 R_{z_1} l_1^t f_1^t M_1^t D_s^t + S_1 R_{z_1} S_1^t$$

$$m\mu_3 = \frac{1}{3 h w} \text{Tr}(T_F f l D_s M_1 f_1 l_1 R_{z_1} l_1^t f_1^t M_1^t D_s^t l^t f^t T_F^t) \quad (6.9)$$

$$- T_F f l D_s M_1 f_1 l_1 R_{z_1} S_1^t l^t Q^t T_X^t$$

$$- T_X Q l S_1 R_{z_1} l_1^t f_1^t M_1^t D_s^t l^t f^t T_F^t$$

$$+ T_X Q l S_1 R_{z_1} S_1^t l^t Q^t T_X^t)$$

$$m\mu_2 = \frac{1}{3 h w} \text{Tr}(T_F f l S_1 R_{z_1} S_1^t l^t f^t T_F^t - T_F f l S_1 R_{z_1} S_1^t l^t Q^t T_X^t) \quad (6.10)$$

$$- T_X Q l S_1 R_{z_1} S_1^t l^t f^t T_F^t + T_X Q l S_1 R_{z_1} S_1^t l^t Q^t T_X^t)$$

$$m\mu_1 = \frac{1}{3 h w} \text{Tr}(T_F f l D_s M_1 f_1 l_1 R_{z_1} l_1^t f_1^t M_1^t D_s^t l^t f^t T_F^t) \quad (6.11)$$

$$- T_F f l D_s M_1 f_1 l_1 R_{z_1} S_1^t l^t f^t T_F^t$$

$$- T_F f l S_1 R_{z_1} l_1^t f_1^t M_1^t D_s^t l^t f^t T_F^t$$

$$+ T_F f l S_1 R_{z_1} S_1^t l^t f^t T_F^t)$$

where  $D_s$  is defined as in equation 6.6 above,  $T_F$  and  $T_X$  are defined in equation 5.4,5.3,  $Q$  is the CIE-XYZ color matching functions.

$\Delta E$  is defined as the distance between two colors. Parmar et al.<sup>12</sup> showed that for a small color difference in XYZ space, its corresponding difference in Lab could be approximated by a linear transform  $J$ . Where  $J$  is gradient of the XYZ to Lab transform chosen around a white point. Refer to equation 6.3 for the nonlinear XYZtoLAB conversion. So we can write it in a linear approximation:

$$\Delta E_3 = \left( \frac{\sum J(\hat{Y}_{XYZ} - R I_{XYZ})^2}{3 N_{DB} H W} \right)^{1/2} \quad (6.12)$$

where  $\Delta E_3$  is the average across all images in the database. Also, let us consider an

average of  $\Delta E^2$ , defined as  $\text{Avg}\Delta E^2$ :

$$\text{Avg}\Delta E_3^2 = \frac{1}{3hw} \text{Tr}(JQl(D_s M_1 f_1 l_1 R_{z_1} l_1^t f_1^t M_1^t D_s^t - D_s M_1 f_1 l_1 R_{z_1} S_1^t - S_1 R_{z_1} l_1^t f_1^t M_1^t D_s^t + S_1 R_{z_1} S_1^t) l^t Q^t J^t) \quad (6.13)$$

### 6.3 Evaluation of metrics in terms of Gaussian filters

We will first evaluate the effect of  $\lambda_\sigma$ , whether using narrow band or broad band filters are beneficial for the demosaicing process. Secondly, we will evaluate the effect of increasing the number of filters. Figure 6-4 shows the effect of increasing  $\lambda_\sigma$  on the green filter. Although gaussian filters are continuous in nature, however, the reflectance databases like Finlayson and Cave are discrete with step size of 10 nm, therefore we interpolate gaussian filters to be discrete values. One thing to remember is that, as the  $\lambda_\sigma$  increases we increase the overall sensitivity of the filter, so the product  $flz$ , pixel values in  $Y$  also increases. It could lead to saturation, therefore to ensure equal sensitivity between different comparisons we modify the  $F$  by dividing it by the maximum of sum of product of  $f$  and  $l$  for different filters in the mosaic. This is somewhat analogous to having something like an exposure control.

Figure 6-5 shows the filter for  $\lambda_\sigma$  equal to 4 for SFA sizes from 2x2, 3x3, 4x4 and Bayer layout. We considered as many filters as the super-pixel permits,  $P_s = hw$ . We positioned the peak wavelength of these filters to ensure equal spacing between them. The idea being to first identify the trend of the performance metrics with changes in  $\lambda_\sigma$ . For the simulations to follow we will use  $D65$  for training.

Figure 6-6, 6-7 shows the demosaiced rendered sRGB Macbeth chart from Finlayson database, considering Bayer SFA with gaussian filters for varying  $\lambda_\sigma$  without white balancing and with white balancing respectively. Purpose of this to show that additional post-processing like white balancing could be required to get visually correct color, however it modifies the effect of  $T_F$ . As the sRGB images change with white balancing, this changes metrics like  $\mu_1$ ,  $\mu_2$  and  $\mu_3$ . Therefore, we do not incor-

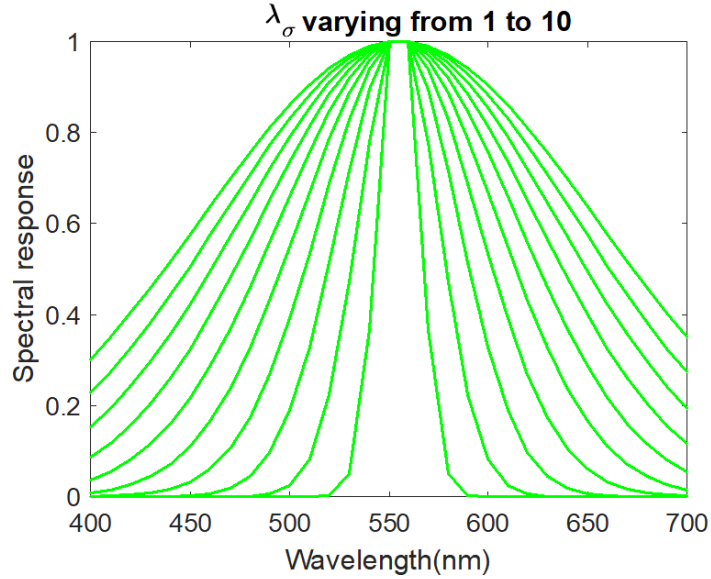


Figure 6-4: With increasing  $\lambda_\sigma$ , filter goes from narrow band to wideband. In the above figure filters with higher  $\lambda_\sigma$  have higher sensitivity than lower ones, therefore we need to normalise them to ensure equal sensitivity.

porate white-balancing in our evaluation procedure. However white balancing would have no effect of  $\mu_Y$  or  $\mu_Z$ . Figure 6-8, 6-9 and 6-10 show the  $\hat{Y}_{sRGB}$ , the demosaiced sRGB rendered Macbeth chart image from the Finlayson database for SFAs 2x2, 3x3, and 4x4 respectively for  $\lambda_\sigma$  varying as [0.25 0.5 1 2 3 ... 10]. We see that for 3x3 and 4x4 SFAs for higher  $\lambda_\sigma$  some of the images are dark. This is because of demosaicing error some of the pixels become saturated (see at the edge of yellow and pink patches) and there is quite a difference in range compared to other pixels. Ideally, in a RAW workflow a white level is set which is generally not equal to 1 as in this case. One could calculate it by first calculating a histogram of all pixels and setting white level equal to say 90% of the cumulative pixel value. This would have allowed us to display these images at the same brightness level. For the purpose of this simulation we don't modify the white level.

If we look at the demosaiced images for 3x3 and 4x4 SFAs we see that there are artifacts present due to demosaicing. This tend to roughly correspond to the  $\mu_Y$  values. It is only at very high sigma values for SFA sizes 3x3 and 4x4 that artifacts tend to disappear however at these values the color difference error is very large. It



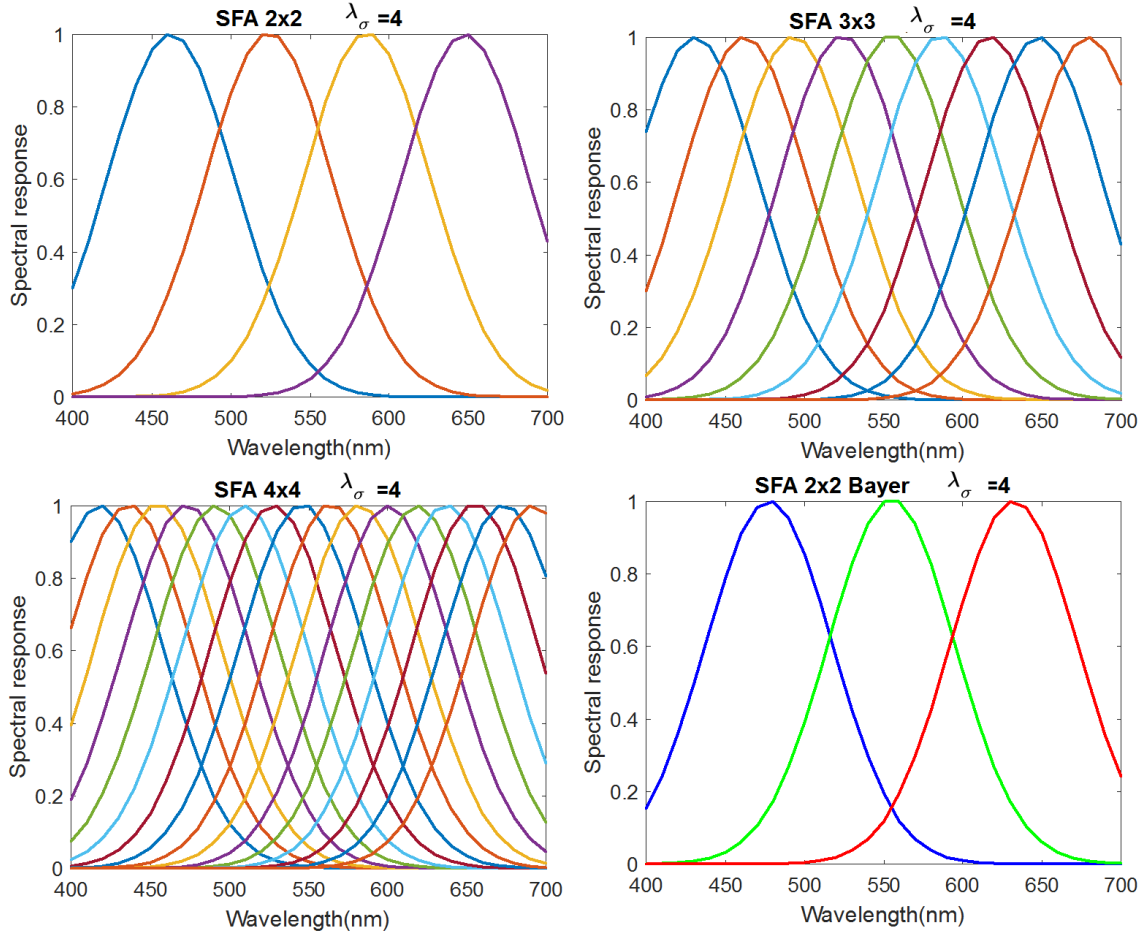


Figure 6-5: Filters for SFAs with  $\lambda_\sigma=4$

is not possible to quantify artifacts to the  $\mu_3$  as this is PSNR in sRGB images and in this case the source of error is twofold, one from demosaicing and secondly from the filter space to sRGB color space transform. One would ideally have liked to have a metric which would correspond to the artifacts due to demosaicing error alone.

Table 6.1 and 6.2 shows the evaluation of the performance metrics for the Finlayson and Cave hyperspectral databases respectively. Output images were not white balanced for this evaluation. If we look at some general trend we see:

$\mu_Y$  in general falls and then increases with increasing  $\lambda_\sigma$  for a given SFA size. The highest PSNR are for very high  $\lambda_\sigma$  which indicates that broadband filters are better for demosaicing, which is expected as the correlations between channels will be high. However this comes at a cost of reduced color accuracy, as  $\Delta E_3$  is low.  $\mu_Y$  decreases

if we increase the SFA size, which indicates that increasing number of filter makes the demosaicing task more complex.

$\Delta E_3$  also falls and then increases very rapidly with increasing  $\lambda_\sigma$ . Therefore, broadband filters are recommended if the goal is to demosaic spectral channels however this comes at the cost decreased color accuracy. So the ideal solution for demosaicing would be somewhere in middle,  $\lambda_\sigma$  of 3.

$\Delta E_1$  and  $\Delta E_2$  also follow similar trend to  $\Delta E_3$  however the point of minima for  $\Delta E_2$  tends to be at smaller value of  $\lambda_\sigma$  than for  $\Delta E_1$ . Also, in absolute terms  $\Delta E_1$  is bigger pointing that the color difference is largely due to choice of spectral sensitivity and filter space to sRGB color space transform and less due to demosaicing error.

$\mu_3$  is more closely aligned to  $\mu_1$  therefore again pointing to contribution of Filter space to sRGB color space and filter choice to PSNR error and less contribution of demosaicing error. *SSIM* follows similar trend to  $\mu_{1,2,3}$ .

$\mu_Z$  first rises with increasing  $\lambda_\sigma$  and then falls down. Having bigger SFAs with more filters is beneficial for recovering spectral radiance information. Which is not the case for demosaicing though.

Figure 6-12 shows the reflectance spectra of the 24 colors in the Macbeth chart image from Finlayson database. Each spectrum is the average of 100 pixels in each color patch. To the left is measurement as recorded by the hyperspectral image capture. In the middle is  $\hat{Z}$ , demosaiced reflectance for Bayer SFA for  $\lambda_\sigma = 4$  and to the right for SFA of size 3x3 for  $\lambda_\sigma = 0.25$ . We choose to present results for such varying  $\lambda_\sigma$  for highlighting the difference in spectral recovery. We see that we are able to recover spectral signature from SFAs to a large extent.



Figure 6-6: Effect of  $\lambda_\sigma$  on demosaicing for the Macbeth chart in Finlayson database for Bayer SFA. Left to Right, then Top to Bottom  $\lambda_\sigma$  goes  $[0.25, 0.5, 1, 2, 3, \dots, 10]$



Figure 6-7: ]  
 Effect of  $\lambda_\sigma$  on demosaicing for the Macbeth chart in Finlayson database for Bayer SFA. Output images are white-balanced by ratio of average Red, Green and Blue pixel values. Left to Right, then Top to Bottom  $\lambda_\sigma$  goes [0.25, 0.5, 1,2,3,...10]



Figure 6-8: Effect of  $\lambda_\sigma$  on demosaicing for the Macbeth chart in Finlayson database for 2x2 SFA size, 4 filters. Left to Right, then Top to Bottom  $\lambda_\sigma$  goes [0.25, 0.5, 1, 2, 3, ..., 10]



Figure 6-9: Effect of  $\lambda_\sigma$  on demosaicing for the Macbeth chart in Finlayson database for 3x3 SFA size, 9 filters. Left to Right, then Top to Bottom  $\lambda_\sigma$  goes [0.25, 0.5, 1,2,3,...10]



Figure 6-10: Effect of  $\lambda_\sigma$  on demosaicing for the Macbeth chart in Finlayson database for 4x4 SFA size, 16 filters. Left to Right, then Top to Bottom  $\lambda_\sigma$  goes [0.25, 0.5, 1,2,3,...10]

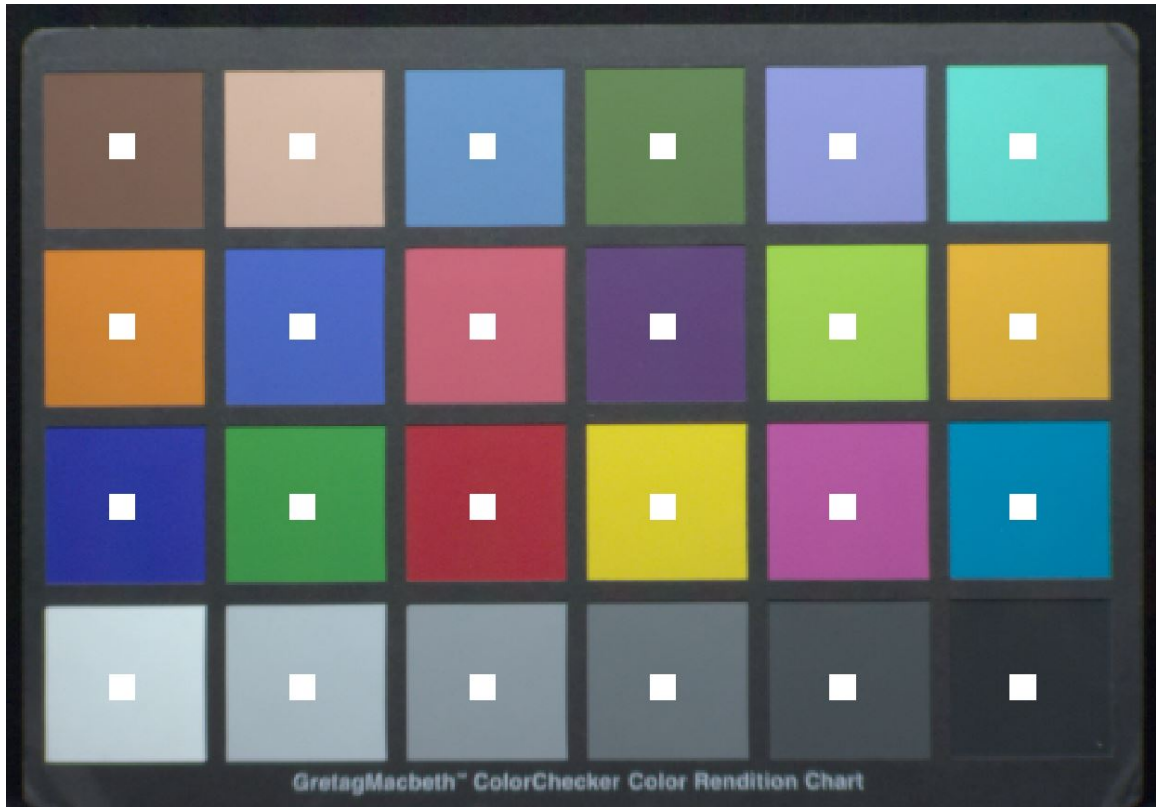


Figure 6-11: Macbeth chart image from the Finlayson database is overlaid with white patches of  $10 \times 10$  pixels. We average pixels from each patch to get an average reflectance value for each color. This average reflectance value is used to determine the PSNR of reflectance  $\mu_{z(\text{chart})}$  between measured (in Finlayson database) and estimated from demosaicing operator  $D_s$

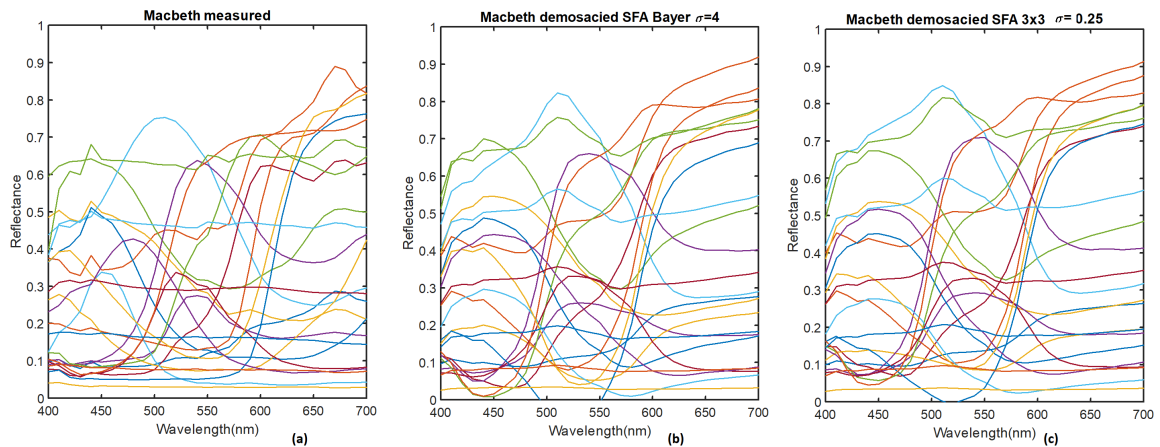


Figure 6-12: Reflectance values for the 24 colors in the Macbeth chart image from the Finlayson database. (a) Reflectance as measured by Finlayson image. (b) Estimated reflectance  $\hat{Z}$  from the  $X$  SFA image for 2x2 Bayer layout of 3 gaussian filters with  $\lambda_\sigma = 4$ . (c) Estimated reflectance  $\hat{Z}$  from the  $X$  SFA image for 3x3 Bayer layout of 9 gaussian filters with  $\lambda_\sigma = 0.25$ .



Table 6.1: Result of varying  $\lambda_\sigma$  for different SFA size. All values are averages across images of the Finlayson database, except  $\mu_{Z(\text{chart})}$  which is average across 24 colors in the Macbeth chart

|              | $\lambda_\sigma$ | $\mu_1$      | $\mu_2$      | $\mu_3$      | $\Delta E_1$ | $\Delta E_2$ | $\Delta E_3$ | $\mu_Y$ | $\mu_{Z(\text{chart})}$ | $\mu_Z$      | $SSIM_1$      | $SSIM_2$      | $SSIM_3$      |
|--------------|------------------|--------------|--------------|--------------|--------------|--------------|--------------|---------|-------------------------|--------------|---------------|---------------|---------------|
| 2x2 gaussian | 0.25             | 26.83        | 32.59        | 26.08        | 9.33         | 3.72         | 10.04        | 39.54   | 26.45                   | 31.38        | 0.9180        | 0.9576        | 0.8960        |
|              | 0.5              | 26.96        | 32.99        | 26.17        | 8.92         | 3.45         | 9.67         | 39.98   | 26.28                   | 31.34        | 0.9206        | 0.9628        | 0.8968        |
|              | 1                | <b>29.22</b> | 33.48        | <b>27.85</b> | 5.44         | 3.08         | 6.57         | 39.86   | 26.83                   | 31.49        | 0.9669        | 0.9675        | 0.9394        |
|              | 2                | 28.29        | <b>34.13</b> | 27.12        | 5.31         | <b>2.81</b>  | 6.56         | 40.37   | 27.46                   | 31.73        | 0.9747        | <b>0.9725</b> | 0.9460        |
|              | 3                | 28.39        | 33.43        | 27.10        | <b>4.33</b>  | 3.32         | <b>5.99</b>  | 40.81   | 27.73                   | <b>31.86</b> | <b>0.9844</b> | 0.9700        | <b>0.9541</b> |
|              | 4                | 27.51        | 32.57        | 26.26        | 5.56         | 4.00         | 7.31         | 41.60   | <b>27.78</b>            | 31.80        | 0.9764        | 0.9647        | 0.9439        |
|              | 5                | 27.33        | 31.79        | 25.88        | 6.17         | 4.59         | 8.14         | 42.54   | <b>27.03</b>            | 31.56        | 0.9733        | 0.9579        | 0.9361        |
|              | 6                | 27.78        | 31.57        | 26.06        | 6.30         | 4.84         | 8.38         | 43.42   | 26.28                   | 31.18        | 0.9730        | 0.9539        | 0.9313        |
|              | 7                | 26.29        | 31.83        | 25.02        | 7.30         | 4.83         | 9.23         | 44.13   | 26.03                   | 30.86        | 0.9653        | 0.9524        | 0.9216        |
|              | 8                | 23.86        | 32.19        | 23.11        | 9.07         | 4.77         | 10.80        | 44.71   | 26.04                   | 30.62        | 0.9486        | 0.9513        | 0.9038        |
|              | 9                | 22.09        | 32.37        | 21.57        | 10.72        | 4.80         | 12.38        | 45.20   | 26.14                   | 30.36        | 0.9288        | 0.9493        | 0.8822        |
| 10           | 20.72            | 32.37        | 20.30        | 12.23        | 4.91         | 13.93        | <b>45.63</b> | 26.29   | 30.02                   | 0.9072       | 0.9460        | 0.8577        |               |
| 3x3 gaussian | 0.25             | 19.03        | <b>33.29</b> | 18.84        | 13.80        | <b>3.54</b>  | 14.62        | 40.29   | 35.32                   | 31.74        | 0.8187        | <b>0.9620</b> | 0.7847        |
|              | 0.5              | 22.89        | 32.44        | 22.35        | 8.52         | 3.74         | 9.84         | 38.51   | 35.16                   | 31.77        | 0.9258        | 0.9576        | 0.8850        |
|              | 1                | 28.57        | 31.74        | 26.64        | 4.10         | 4.01         | 6.35         | 37.33   | 35.09                   | 31.86        | 0.9901        | 0.9548        | 0.9442        |
|              | 2                | 29.18        | 31.84        | 27.10        | 4.01         | 4.08         | <b>6.29</b>  | 38.17   | 33.68                   | <b>31.98</b> | 0.9911        | 0.9571        | 0.9477        |
|              | 3                | 29.12        | 31.89        | 27.09        | 3.81         | 4.28         | 6.40         | 39.10   | 31.50                   | 31.84        | 0.9920        | 0.9581        | <b>0.9495</b> |
|              | 4                | 30.01        | 31.44        | 27.50        | 3.34         | 4.95         | 6.62         | 39.88   | 29.88                   | 31.61        | 0.9937        | 0.9536        | 0.9476        |
|              | 5                | <b>40.75</b> | 30.14        | <b>29.76</b> | <b>0.98</b>  | 6.31         | 6.51         | 40.50   | 29.09                   | 31.38        | <b>0.9993</b> | 0.9428        | 0.9427        |
|              | 6                | 30.14        | 28.87        | 26.35        | 3.23         | 7.89         | 9.31         | 41.18   | 28.28                   | 30.99        | 0.9939        | 0.9286        | 0.9239        |
|              | 7                | 23.88        | 27.42        | 22.18        | 6.72         | 14.61        | 18.56        | 41.75   | 27.36                   | 30.54        | 0.9725        | 0.9130        | 0.8899        |
|              | 8                | 12.65        | 29.22        | 12.56        | 26.01        | 15.59        | 38.73        | 42.24   | 26.58                   | 30.13        | 0.5355        | 0.9223        | 0.5028        |
|              | 9                | 8.76         | 32.00        | 8.73         | 42.18        | 6.14         | 47.75        | 42.66   | 26.12                   | 29.73        | 0.0517        | 0.9499        | 0.0494        |
| 10           | 9.79             | 32.19        | 9.76         | 37.68        | 5.52         | 41.65        | <b>43.02</b> | 25.79   | 29.28                   | 0.1702       | 0.9385        | 0.1610        |               |
| 4x4 gaussian | 0.25             | 20.89        | <b>31.62</b> | 20.45        | 11.30        | 4.36         | 12.67        | 38.98   | 34.24                   | 31.78        | 0.8995        | 0.9444        | 0.8450        |
|              | 0.5              | 21.53        | 31.57        | 21.01        | 9.23         | 4.23         | 10.93        | 38.13   | 34.26                   | 31.79        | 0.9431        | 0.9434        | 0.8861        |
|              | 1                | 25.45        | 30.87        | 24.17        | 5.67         | 4.48         | 8.06         | 36.96   | 34.23                   | 32.05        | 0.9811        | 0.9421        | 0.9226        |
|              | 2                | 28.42        | 30.56        | 26.13        | 4.00         | 4.71         | <b>6.95</b>  | 37.18   | 34.85                   | <b>32.05</b> | 0.9908        | 0.9428        | <b>0.9331</b> |
|              | 3                | <b>29.93</b> | 29.61        | <b>26.55</b> | <b>3.37</b>  | 5.81         | 7.55         | 37.81   | 33.11                   | 31.78        | <b>0.9936</b> | 0.9352        | 0.9287        |
|              | 4                | 29.02        | 23.81        | 22.46        | 3.74         | 83.09        | 85.06        | 38.53   | 31.06                   | 31.36        | 0.9920        | 0.8891        | 0.8826        |
|              | 5                | 10.49        | 27.75        | 10.42        | 34.40        | 10.60        | 43.43        | 39.23   | 29.58                   | 30.88        | 0.2628        | 0.9019        | 0.2451        |
|              | 6                | 15.56        | 25.61        | 15.09        | 18.20        | 59.53        | 74.56        | 39.90   | 28.25                   | 30.40        | 0.7713        | 0.8935        | 0.6972        |
|              | 7                | 11.82        | 26.29        | 11.66        | 28.96        | 45.74        | 72.09        | 40.60   | 27.19                   | 29.97        | 0.4369        | 0.8966        | 0.4012        |
|              | 8                | 13.80        | 26.64        | 13.55        | 22.54        | 68.76        | 87.79        | 41.25   | 26.53                   | 29.60        | 0.6454        | 0.8933        | 0.5840        |
|              | 9                | 11.20        | 27.75        | 11.09        | 31.36        | 18.83        | 47.54        | 41.83   | 26.03                   | 29.22        | 0.3579        | 0.9040        | 0.3280        |
| 10           | 8.32             | 31.08        | 8.29         | 43.64        | <b>3.99</b>  | 47.31        | <b>42.35</b> | 25.46   | 28.83                   | 0.0158       | <b>0.9498</b> | 0.0147        |               |
| 2x2 bayer    | 0.25             | 17.60        | <b>36.65</b> | 17.55        | 27.69        | 2.34         | 27.82        | 43.74   | 23.21                   | 30.02        | 0.5913        | <b>0.9874</b> | 0.5810        |
|              | 0.5              | 21.71        | 35.60        | 21.54        | 14.75        | 2.53         | 15.10        | 42.10   | 23.34                   | 30.10        | 0.7965        | 0.9797        | 0.7811        |
|              | 1                | 25.52        | 35.24        | 25.05        | 10.57        | 2.48         | 11.10        | 40.84   | 23.60                   | 30.22        | 0.8705        | 0.9787        | 0.8523        |
|              | 2                | 25.22        | 36.15        | 24.78        | 10.58        | <b>2.10</b>  | 11.11        | 41.36   | 24.54                   | 30.53        | 0.8701        | 0.9822        | 0.8509        |
|              | 3                | 25.46        | 35.77        | 24.98        | 8.99         | 2.27         | 9.68         | 41.80   | 25.08                   | 30.78        | 0.9032        | 0.9808        | 0.8823        |
|              | 4                | 25.84        | 35.19        | 25.30        | <b>7.75</b>  | 2.60         | <b>8.62</b>  | 42.26   | 25.24                   | <b>30.86</b> | 0.9388        | 0.9779        | 0.9165        |
|              | 5                | <b>26.15</b> | 34.64        | <b>25.53</b> | 8.04         | 2.98         | 8.95         | 42.71   | 25.43                   | 30.82        | <b>0.9460</b> | 0.9753        | <b>0.9231</b> |
|              | 6                | 26.14        | 34.14        | 25.44        | 8.47         | 3.34         | 9.44         | 43.32   | 25.59                   | 30.64        | 0.9411        | 0.9724        | 0.9172        |
|              | 7                | 26.04        | 33.66        | 25.26        | 8.56         | 3.70         | 9.69         | 43.97   | 25.77                   | 30.42        | 0.9360        | 0.9687        | 0.9101        |
|              | 8                | 24.77        | 33.37        | 24.08        | 9.31         | 3.98         | 10.59        | 44.52   | 25.95                   | 30.17        | 0.9248        | 0.9646        | 0.8962        |
|              | 9                | 22.86        | 33.26        | 22.35        | 10.76        | 4.21         | 12.15        | 44.98   | 26.14                   | 29.83        | 0.9046        | 0.9604        | 0.8722        |
| 10           | 21.40            | 33.07        | 20.97        | 12.16        | 4.50         | 13.73        | <b>45.40</b> | 26.33   | 29.45                   | 0.8809       | 0.9553        | 0.8435        |               |

Table 6.2: Result of varying  $\lambda_\sigma$  for different SFA size. All values are averages across images of the Cave database.

|              | $\lambda_\sigma$ | $\mu_1$      | $\mu_2$      | $\mu_3$      | $\Delta E_1$ | $\Delta E_2$ | $\Delta E_3$ | $\mu_Y$ | $\mu_Z$      | $SSIM_1$      | $SSIM_2$      | $SSIM_3$      |
|--------------|------------------|--------------|--------------|--------------|--------------|--------------|--------------|---------|--------------|---------------|---------------|---------------|
| 2x2 gaussian | 0.25             | 27.50        | 37.50        | 27.08        | 9.35         | 2.26         | 9.75         | 43.68   | 29.61        | 0.8747        | 0.9726        | 0.8545        |
|              | 0.5              | 27.63        | 37.78        | 27.22        | 9.20         | 2.14         | 9.59         | 44.14   | 29.60        | 0.8745        | 0.9751        | 0.8545        |
|              | 1                | 29.77        | 38.16        | 29.08        | 5.98         | 1.99         | 6.55         | 43.92   | 29.73        | 0.9398        | 0.9768        | 0.9193        |
|              | 2                | 29.40        | <b>38.81</b> | 28.78        | 5.11         | <b>1.88</b>  | 5.80         | 44.45   | 29.93        | 0.9565        | 0.9789        | 0.9360        |
|              | 3                | <b>30.02</b> | 38.28        | <b>29.25</b> | <b>3.60</b>  | 2.18         | <b>4.69</b>  | 45.00   | <b>29.99</b> | <b>0.9798</b> | <b>0.9790</b> | <b>0.9587</b> |
|              | 4                | 29.20        | 37.36        | 28.42        | 5.74         | 2.58         | 6.63         | 45.83   | 29.79        | 0.9576        | 0.9784        | 0.9371        |
|              | 5                | 28.70        | 35.93        | 27.80        | 7.36         | 3.21         | 8.23         | 46.73   | 29.40        | 0.9343        | 0.9744        | 0.9177        |
|              | 6                | 28.68        | 34.46        | 27.63        | 8.22         | 3.98         | 8.96         | 47.68   | 29.15        | 0.9199        | 0.9644        | 0.9105        |
|              | 7                | 27.44        | 33.79        | 26.56        | 8.91         | 4.39         | 9.43         | 48.59   | 29.20        | 0.9124        | 0.9506        | 0.9037        |
|              | 8                | 25.45        | 33.69        | 24.81        | 9.64         | 4.44         | 10.07        | 49.32   | 29.19        | 0.9061        | 0.9393        | 0.8897        |
|              | 9                | 23.88        | 33.62        | 23.37        | 10.30        | 4.44         | 10.76        | 49.90   | 29.05        | 0.8979        | 0.9308        | 0.8717        |
| 10           | 22.61            | 33.56        | 22.17        | 10.93        | 4.48         | 11.52        | <b>50.38</b> | 28.80   | 0.8869       | 0.9242        | 0.8517        |               |
| 3x3 gaussian | 0.25             | 20.97        | <b>37.30</b> | 20.83        | 11.61        | <b>2.33</b>  | 12.16        | 43.68   | 31.13        | 0.7795        | <b>0.9674</b> | 0.7516        |
|              | 0.5              | 24.79        | 36.42        | 24.38        | 7.36         | 2.53         | 8.22         | 41.78   | 31.18        | 0.9028        | 0.9635        | 0.8693        |
|              | 1                | 30.59        | 35.83        | 29.09        | 3.29         | 2.74         | 4.81         | 40.53   | 31.31        | 0.9885        | 0.9643        | 0.9525        |
|              | 2                | 31.35        | 35.92        | 29.70        | 3.37         | 2.88         | <b>4.81</b>  | 41.60   | <b>31.47</b> | 0.9903        | 0.9653        | <b>0.9550</b> |
|              | 3                | 31.25        | 35.73        | 29.65        | 3.04         | 3.27         | 5.06         | 42.88   | 31.23        | 0.9918        | 0.9601        | 0.9522        |
|              | 4                | 32.04        | 34.28        | 29.90        | 2.70         | 4.77         | 5.87         | 43.80   | 30.71        | 0.9922        | 0.9397        | 0.9414        |
|              | 5                | <b>42.57</b> | 31.64        | <b>31.37</b> | <b>0.84</b>  | 7.68         | 7.67         | 44.53   | 30.23        | <b>0.9990</b> | 0.8959        | 0.9027        |
|              | 6                | 32.16        | 29.40        | 27.39        | 2.63         | 11.30        | 11.99        | 45.34   | 29.74        | 0.9923        | 0.8480        | 0.8590        |
|              | 7                | 25.93        | 28.22        | 23.75        | 5.40         | 22.21        | 24.42        | 46.02   | 29.24        | 0.9671        | 0.8282        | 0.8355        |
|              | 8                | 14.76        | 31.17        | 14.69        | 20.13        | 23.53        | 39.85        | 46.55   | 28.84        | 0.4873        | 0.8940        | 0.5140        |
|              | 9                | 11.00        | 34.37        | 10.98        | 31.63        | 9.82         | 40.85        | 46.98   | 28.54        | 0.0540        | 0.9596        | 0.0574        |
| 10           | 11.98            | 34.84        | 11.96        | 28.39        | 6.45         | 33.28        | <b>47.33</b> | 28.22   | 0.1495       | 0.9466        | 0.1611        |               |
| 4x4 gaussian | 0.25             | 22.88        | <b>35.30</b> | 22.55        | 8.61         | 2.95         | 9.59         | 41.97   | 31.63        | 0.9018        | 0.9596        | 0.8623        |
|              | 0.5              | 23.57        | 35.29        | 23.17        | 7.31         | <b>2.91</b>  | 8.39         | 41.15   | 31.99        | 0.9373        | 0.9574        | 0.8965        |
|              | 1                | 27.53        | 34.75        | 26.53        | 4.52         | 3.10         | 6.08         | 40.13   | <b>32.05</b> | 0.9784        | 0.9544        | 0.9338        |
|              | 2                | 30.47        | 34.52        | 28.71        | 3.23         | 3.35         | <b>5.17</b>  | 40.64   | 31.39        | 0.9889        | 0.9527        | <b>0.9438</b> |
|              | 3                | <b>32.01</b> | 32.91        | <b>29.03</b> | <b>2.69</b>  | 5.00         | 6.31         | 41.44   | 30.99        | <b>0.9926</b> | 0.9412        | 0.9373        |
|              | 4                | 31.10        | 26.02        | 24.33        | 2.99         | 130.11       | 131.46       | 42.30   | 30.55        | 0.9909        | 0.8968        | 0.8950        |
|              | 5                | 12.66        | 30.52        | 12.57        | 26.07        | 12.51        | 37.35        | 43.11   | 30.04        | 0.2286        | 0.9276        | 0.2295        |
|              | 6                | 17.65        | 28.24        | 17.16        | 14.34        | 78.94        | 89.77        | 43.82   | 29.38        | 0.7400        | 0.8932        | 0.7134        |
|              | 7                | 13.95        | 29.21        | 13.79        | 22.22        | 64.83        | 84.19        | 44.52   | 28.75        | 0.3901        | 0.9038        | 0.4054        |
|              | 8                | 15.90        | 28.88        | 15.66        | 17.59        | 107.61       | 120.79       | 45.15   | 28.43        | 0.6027        | 0.8739        | 0.6036        |
|              | 9                | 13.34        | 30.22        | 13.24        | 23.92        | 31.61        | 52.64        | 45.71   | 28.30        | 0.3140        | 0.9081        | 0.3338        |
| 10           | 10.58            | 33.90        | 10.54        | 32.69        | 8.12         | 40.51        | <b>46.18</b> | 28.05   | 0.0241       | <b>0.9605</b> | 0.0246        |               |
| 2x2 bayer    | 0.25             | 19.52        | <b>41.57</b> | 19.49        | 23.15        | <b>1.42</b>  | 23.23        | 48.40   | 28.55        | 0.4939        | <b>0.9897</b> | 0.4843        |
|              | 0.5              | 23.34        | 40.27        | 23.24        | 13.51        | 1.57         | 13.71        | 46.57   | 28.63        | 0.7189        | 0.9822        | 0.7044        |
|              | 1                | 26.70        | 39.72        | 26.45        | 10.42        | 1.62         | 10.71        | 45.17   | 28.69        | 0.8037        | 0.9802        | 0.7871        |
|              | 2                | 26.72        | 40.46        | 26.46        | 9.77         | 1.43         | 10.08        | 45.70   | 28.89        | 0.8164        | 0.9829        | 0.8002        |
|              | 3                | 27.24        | 40.33        | 26.93        | 7.69         | 1.54         | 8.12         | 46.28   | 29.12        | 0.8786        | 0.9824        | 0.8616        |
|              | 4                | 27.82        | 39.99        | 27.47        | <b>5.96</b>  | 1.76         | <b>6.62</b>  | 46.91   | 29.29        | 0.9438        | 0.9826        | 0.9264        |
|              | 5                | <b>27.99</b> | 39.63        | <b>27.60</b> | 6.76         | 1.99         | 7.43         | 47.54   | <b>29.34</b> | <b>0.9530</b> | 0.9837        | <b>0.9361</b> |
|              | 6                | 27.71        | 39.22        | 27.31        | 8.08         | 2.20         | 8.70         | 48.31   | 29.31        | 0.9368        | 0.9843        | 0.9203        |
|              | 7                | 27.49        | 38.79        | 27.06        | 8.67         | 2.39         | 9.32         | 49.10   | 29.22        | 0.9229        | 0.9841        | 0.9064        |
|              | 8                | 26.35        | 38.52        | 25.99        | 9.14         | 2.53         | 9.83         | 49.76   | 29.03        | 0.9116        | 0.9834        | 0.8951        |
|              | 9                | 24.61        | 38.35        | 24.34        | 9.83         | 2.64         | 10.54        | 50.32   | 28.78        | 0.8978        | 0.9821        | 0.8813        |
| 10           | 23.24            | 38.07        | 23.02        | 10.53        | 2.83         | 11.28        | <b>50.81</b> | 28.46   | 0.8818       | 0.9797        | 0.8650        |               |

## 6.4 Design of optimal spectral transmission sensitivity of filters

With respect to the optimization metrics as defined earlier we searched for the optimal gaussian shaped filters for different SFA sizes. We used Matlab's `fmincon` to minimize a cost function. The cost function was set to the performance metric in question (for example  $\mu_Y$ ) while modifying all the parameters of the gaussian filter. Table 6.3 shows the various metrics evaluated for the optimized filters. Figure 6-13 , 6-14 and 6-15 shows the filters obtained from the optimization procedure applied on the Finlayson database.

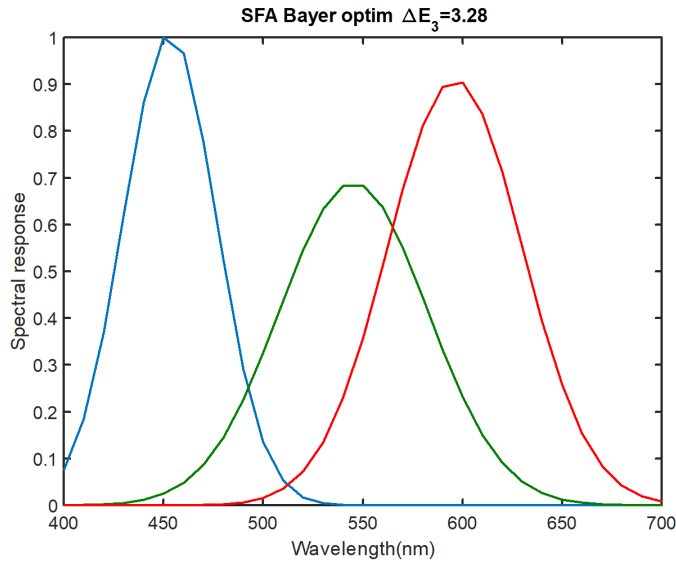


Figure 6-13: Optimized filters with  $\Delta E_3$  as the criteria for Bayer SFA. Optimized filters for neighborhood of 7, optimized on the Finlayson database.

Figure 6-13 shows the optimized filters for  $\Delta E_3$ , optimized for Bayer CFA. We obtain an average  $\Delta E_3$  of 3.28 for all images in the Finlayson database. Correspondingly we get  $\Delta E_3$  of 2.24 for the Cave database. Parmar et al. report 3.2724 for the Cave database<sup>12</sup> using their optimized filters for the Bayer pattern.

Figure 6-14, (a) shows the optimized filter for  $\mu_Y$ , filter most suited for less demosaicing errors. The optimized filter give us PSNR in range of 51-56 dB which is considerably higher than we got from the systematic gaussian evaluation, where the

maximum was 45.63 dB. However, the filters so obtained have very low  $\mu_1$  and  $\mu_3$ .  $\Delta E_3$  is also very high, therefore they are not suitable for color applications.

Figure 6-14, (b) shows the optimized filter for  $\mu_Z$ , filters most suited for recovering reflectance. The highest PSNR obtained is 32.70 dB which is higher than what was obtained earlier using a systematic evaluation with  $\lambda_\sigma$  which was 32.05. What is interesting here is for the 4x4 SFA, the  $\mu_1$  is only 8.21 and  $\Delta E_1$  is 44, so the Filter space to sRGB color space transform totally fails. Even though  $\mu_Y$  at 39.25 is decent for demosaicing, this particular filter combination will not produce good color images. The 2x2 and 3x3 SFA filters are suitable for both color demosaicing and reflectance recovery.

Figure 6-14, (c) shows optimized filters for  $\Delta E_3$ , less color errors in demosaiced images. The best combination give an error of only 3.11 for 2x2 SFA which is coming mainly from demosaicing and not from filter space to sRGB color space transform. These filters are suitable for color demosaicing applications.

Figure 6-14, (d) shows optimized filters for  $\mu_1$ , for applications like color wheel camera where the purpose is to get best possible colors in the sRGB domain. We have a very high PSNR of 71.92 for 16 filters with a very small  $\Delta E_3$ .

Figure 6-15 shows the optimized filters for  $\mu_3$ . We present two cases for the 2x2 SFA, in one case the filters converged to a Bayer arrangement. They have good demosaicing performance with less color errors.

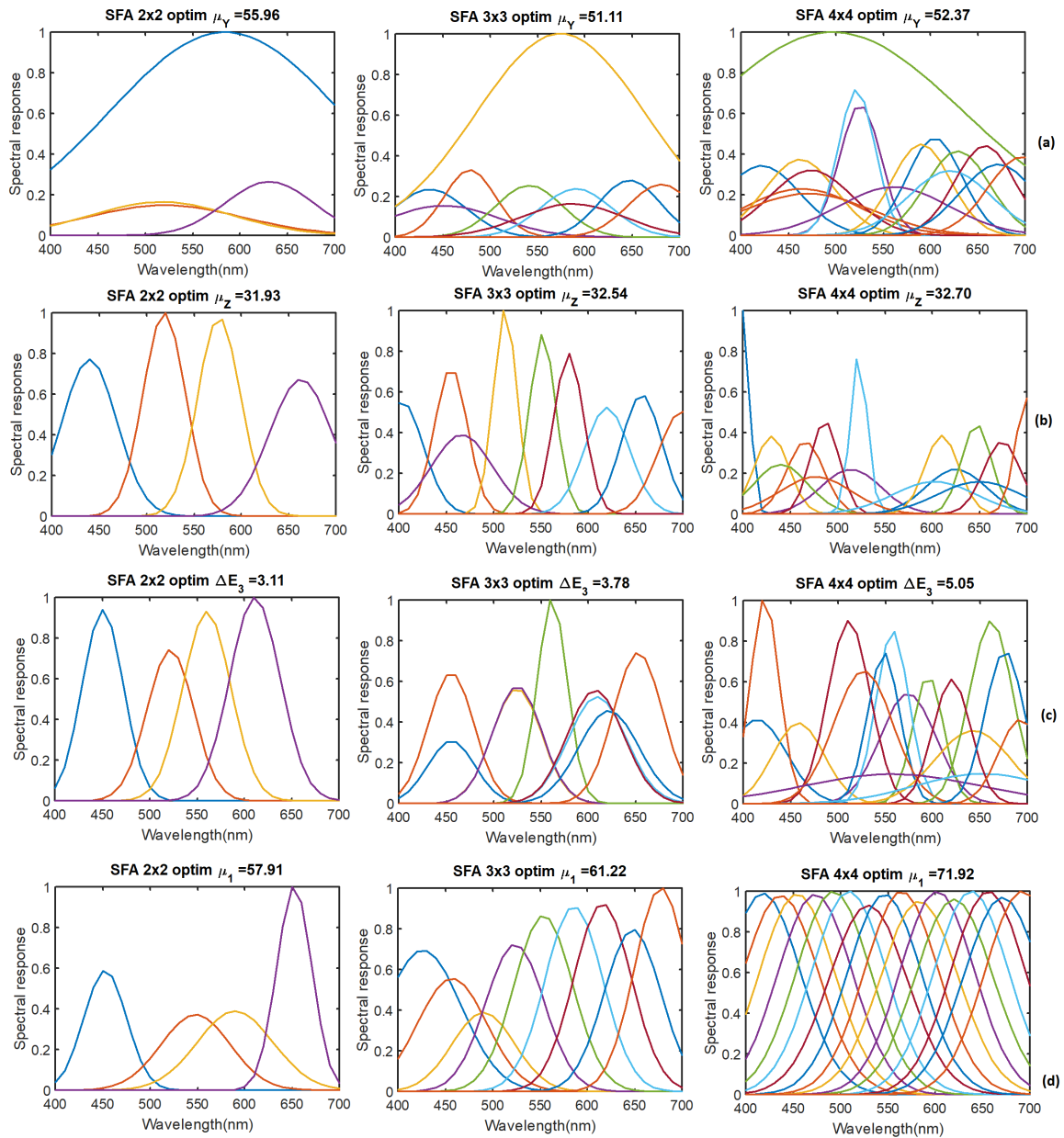


Figure 6-14: Optimized filters for neighborhood of 7, optimized on the Finlayson database. First row:  $\mu_Y$  as the criteria. Here the goal is to minimize demosaicing error in PSNR. Although demosaicing performance is very good however Filter space to sRGB color space transform is very bad and corresponding color difference errors are large. So these filters are not very useful. Second row:  $\mu_Z$  as criteria. Third row:  $\Delta E_3$  as criteria. The most useful filters we obtained, simultaneously other parameters like  $\mu_Y$  and  $\mu_Z$  are also good. Fourth row:  $\mu_1$  as criteria.

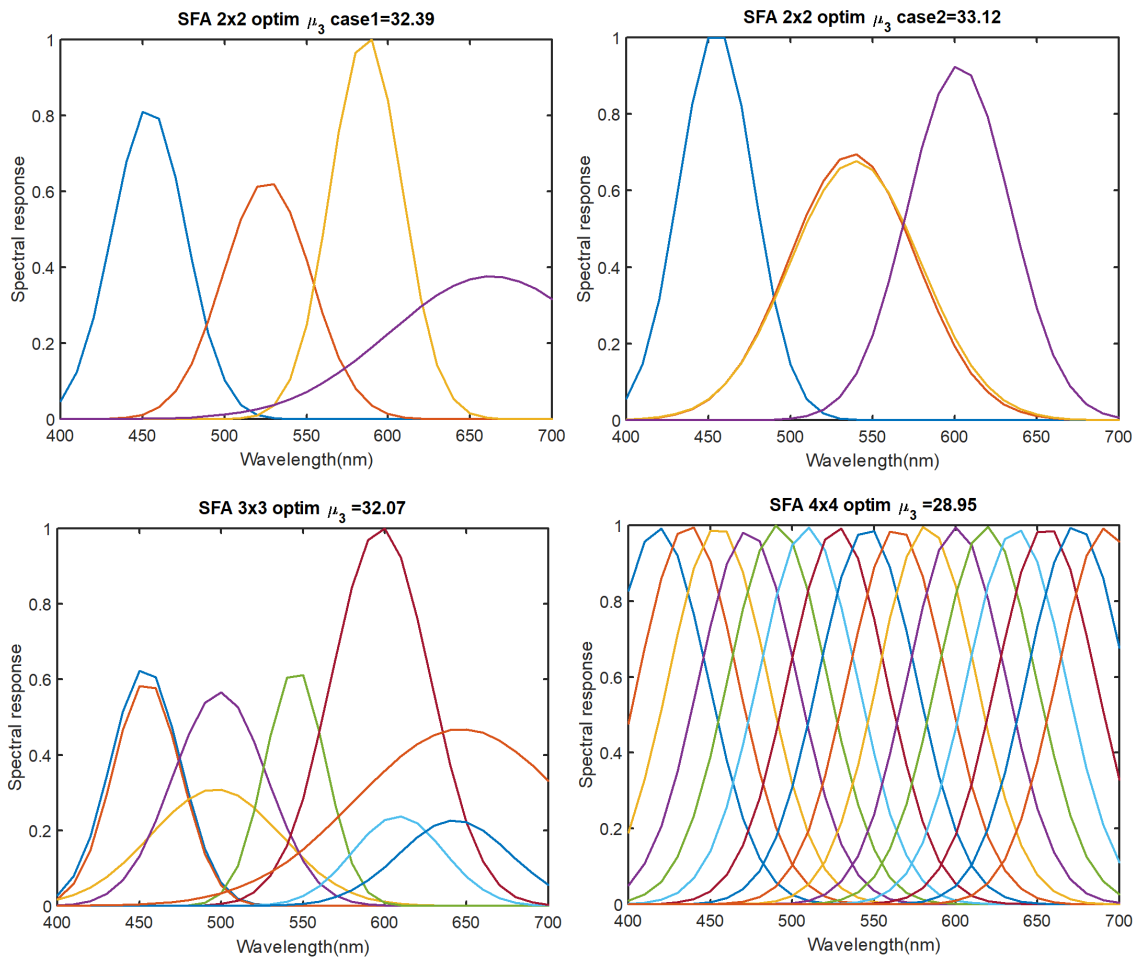


Figure 6-15: Optimized filters with  $\mu_3$  as the criteria. We present two cases for the 2x2 filters obtained using different initialisation. In the second case the filter converged to a Bayer pattern. optimized filters for neighborhood of 7, optimized on the Finlayson database.

Table 6.3: Result of optimized filters for different SFA size for the Finlayson database, neighborhood size 7.

| Optim.       | SFA   | $\mu_1$      | $\mu_2$      | $\mu_3$      | $\Delta E_1$ | $\Delta E_2$ | $\Delta E_3$ | $\mu_Y$      | $\mu_{Z(\text{chart})}$ | $\mu_Z$      | $SSIM_1$      | $SSIM_2$      | $SSIM_3$      |
|--------------|-------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-------------------------|--------------|---------------|---------------|---------------|
| $\mu_Y$      | 2x2   | 8.75         | 51.13        | 8.74         | 42.26        | 0.51         | 42.44        | <b>55.96</b> | 20.61                   | 27.46        | 0.0464        | <b>0.9923</b> | 0.0445        |
|              | 3x3   | 9.25         | 48.03        | 9.25         | 40.13        | 0.57         | 40.20        | 51.11        | 30.86                   | 31.72        | 0.1046        | 0.9913        | 0.1022        |
|              | 4x4   | 8.58         | 49.63        | 8.58         | 42.76        | 0.26         | 42.80        | 52.37        | 31.99                   | 32.21        | 0.0351        | 0.9906        | 0.0350        |
| $\mu_Z$      | 2x2   | 29.26        | 33.66        | 27.83        | 5.24         | 3.13         | 6.38         | 40.47        | 28.30                   | 31.93        | 0.9639        | 0.9687        | 0.9357        |
|              | 3x3   | 26.81        | 32.94        | 25.70        | 5.40         | 3.47         | 6.98         | 39.09        | 34.21                   | 32.54        | 0.9840        | 0.9587        | 0.9423        |
|              | 4x4   | 8.21         | <b>63.63</b> | 8.21         | 44.00        | <b>0.07</b>  | 44.01        | 39.25        | 34.23                   | <b>32.70</b> | 0.0095        | 0.9994        | 0.0095        |
| $\Delta E_3$ | 2x2   | 55.60        | 33.22        | <b>33.18</b> | 0.25         | 3.10         | <b>3.11</b>  | 40.49        | 25.04                   | 30.73        | 0.9998        | 0.9708        | <b>0.9708</b> |
|              | 3x3   | 56.05        | 31.88        | 31.86        | 0.33         | 3.75         | 3.78         | 39.57        | 29.29                   | 31.56        | 0.9998        | 0.9642        | 0.9640        |
|              | 4x4   | 59.93        | 30.86        | 30.84        | 0.21         | 5.03         | 5.05         | 41.96        | <b>34.48</b>            | 32.43        | 0.9999        | 0.9478        | 0.9476        |
|              | bayer | 51.61        | 32.99        | 32.84        | 0.34         | 3.24         | 3.28         | 41.58        | 22.67                   | 29.81        | 0.9997        | 0.9690        | 0.9686        |
| $\mu_1$      | 2x2   | 57.91        | 31.36        | 31.37        | 0.24         | 4.29         | 4.28         | 40.52        | 27.79                   | 31.71        | 0.9999        | 0.9579        | 0.9577        |
|              | 3x3   | 61.22        | 30.94        | 30.95        | 0.21         | 4.92         | 4.92         | 39.53        | 31.46                   | 31.76        | 0.9999        | 0.9535        | 0.9536        |
|              | 4x4   | <b>71.92</b> | 23.68        | 23.69        | <b>0.05</b>  | 98.37        | 98.36        | 38.58        | 31.14                   | 31.36        | <b>1.0000</b> | 0.8908        | 0.8909        |
| $\mu_3$      | 2x2   | 51.55        | 33.13        | 33.12        | 0.46         | 3.24         | 3.27         | 42.14        | 22.77                   | 29.88        | 0.9998        | 0.9707        | 0.9706        |
|              | 3x3   | 53.10        | 32.10        | 32.07        | 0.42         | 3.90         | 3.96         | 43.81        | 29.93                   | 31.67        | 0.9996        | 0.9638        | 0.9635        |
|              | 4x4   | 58.77        | 28.95        | 28.95        | 0.15         | 6.29         | 6.30         | 37.88        | 33.11                   | 31.78        | <b>1.0000</b> | 0.9327        | 0.9327        |

## 6.5 Effect on demosaicing performance due to illuminant

The  $D$  operator we defined depends on the illuminant  $l$  matrix (equation 6.6). Therefore it is important to characterize the role of illuminant difference in terms of demosaicing performance. Let us consider  $D65$  to be the illuminant used for training  $D$ . We use this  $D$  to reconstruct SFA images rendered under different standard illumination conditions ( $D65$ ,  $D50$ ,  $A$ ,  $F2$ ).

Table 6.4: Result of  $\mu_Y$  for demosaicing operator trained on  $D65$  and used to demosaic SFA images rendered for different standard illuminants, for different SFA sizes and  $\lambda_\mu = 3$ .

|           | Illuminant | 2x2   | 3x3   | 4x4   |
|-----------|------------|-------|-------|-------|
| Finlayson | D65        | 40.81 | 39.10 | 37.81 |
|           | D50        | 39.74 | 38.00 | 36.71 |
|           | A          | 42.33 | 39.65 | 38.26 |
|           | F2         | 39.90 | 40.70 | 37.76 |
| Cave      | D65        | 45.00 | 42.88 | 41.44 |
|           | D50        | 43.67 | 41.64 | 40.16 |
|           | A          | 45.50 | 43.41 | 41.60 |
|           | F2         | 45.05 | 44.85 | 42.97 |

Table 6.4 shows the  $\mu_Y$  obtained for the Finlayson and Cave database for gaussian shaped filters having  $\lambda_\mu = 3$  for different SFA sizes. It can be seen that despite changes in illumination the differences in  $\mu_Y$  is not high. Thus we can conclude that LMMSE is quite robust to illumination changes between learning and reconstruction.

## 6.6 Radiance recovery for Visible+NIR filter array

Let us consider SFA having gaussian shaped spectral filters spanning both visible and NIR wavelength. For purpose of testing we consider the SCIEN database. The images provided are in Radiance and not Reflectance, therefore we attempt to recover radiance. The procedure we follow is as described earlier, we simply replace  $LZ$  by



the radiance values provided in the images. This can be done by considering  $L$  to be 1 and  $Z$  to be Radiance (instead of reflectance). We keep this consideration only for this section. Otherwise one can also write another matrix model between  $X$  and  $Radiance$  and determine the LMMSE solution for it and come to the same conclusion.

We simulated gaussian shaped filter across wavelength range of 414.7 to 950.5nm. Figure 6-16 shows the simulated filters for a  $\sigma$  of 6. Figure 6-17 and 6-18 shows the recovered radiances for 4 different pixels points in the StandfordDish image for SFAs of 3x3 and Bayer type. We see that for the 3x3 SFA, the recovered radiance follows the measured (from original test images) more closely than the Bayer SFA. Table 6.5 shows the evaluation of performance metrics for variation in  $\lambda_\sigma$  for different SFA type. We find that a SFA size of 3x3 is more appropriate compared to a Bayer kind of an arrangement. Even though a 2x2 SFA has similar performance for  $\mu_Z$  compared to 3x3 SFA, the color difference  $\Delta E_3$  are very high so it won't be suitable for mixed applications involving both color demosaicing and spectral recovery.

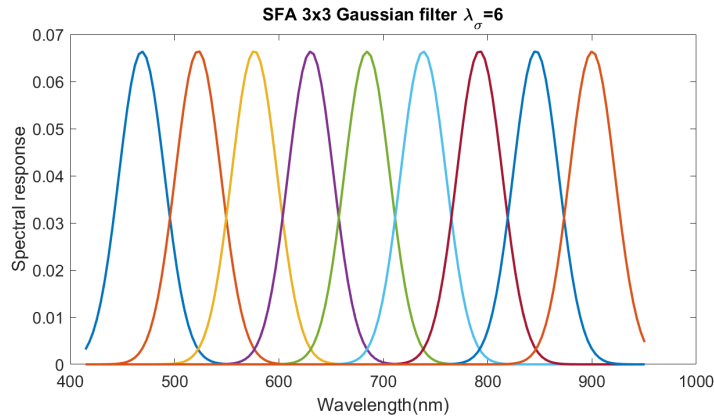


Figure 6-16: Simulated Gaussian Filters on the Visible+NIR wavelength

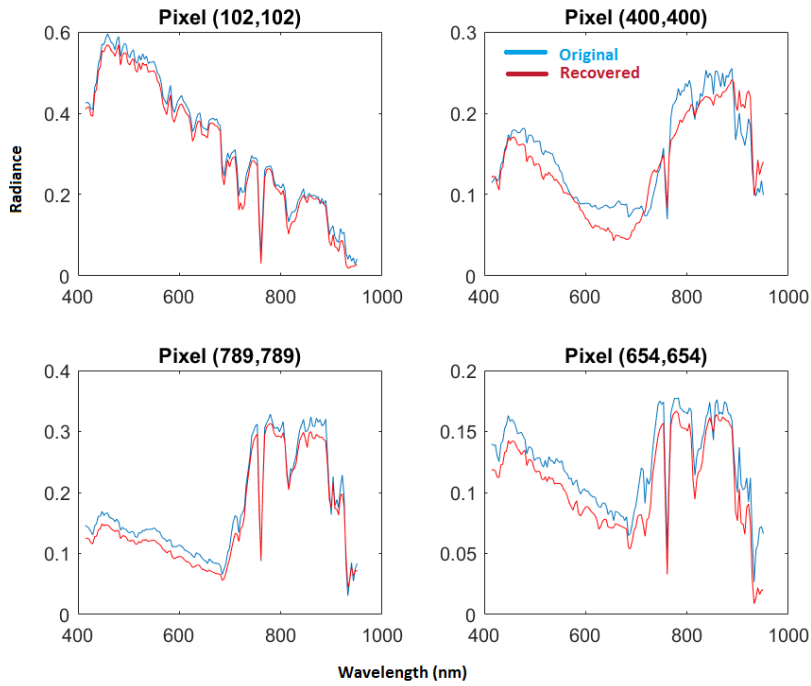


Figure 6-17: Recovered Radiance from 4 different pixel points on the StandfordDish image from SCIEN for SFA  $3 \times 3$   $\lambda_\sigma=6$

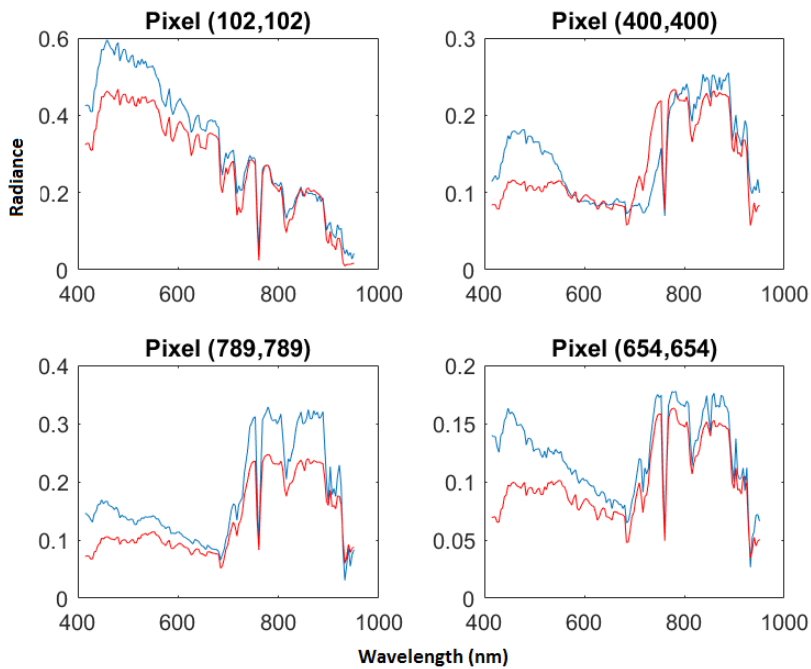


Figure 6-18: Recovered Radiance from 4 different pixel points on the StandfordDish image from SCIEN for SFA Bayer  $\lambda_\sigma=0.25$ . We choose to present this to purposefully highlight a bad case.

Table 6.5: Result of varying  $\lambda_\sigma$  for different SFA sizes for Visible+NIR filters. All values are averages across images of the SCIEN database.

| SFA   | $\lambda_\sigma$ | $\mu_1$      | $\mu_2$      | $\mu_3$      | $\Delta E_1$ | $\Delta E_2$ | $\Delta E_3$ | $\mu_Y$      | $\mu_Z$      |       |
|-------|------------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|--------------|-------|
| 2x2   | 0.25             | 11.11        | 39.65        | 11.12        | 54.72        | 1.58         | 54.70        | 45.69        | 36.89        |       |
|       | 0.5              | 11.87        | 40.02        | 11.87        | 52.33        | 1.60         | 52.33        | 46.12        | <b>36.99</b> |       |
|       | 1                | 12.58        | 40.41        | 12.57        | 48.86        | 1.58         | 48.87        | 46.00        | 36.83        |       |
|       | 2                | 12.70        | 40.80        | 12.70        | 47.13        | 1.54         | 47.14        | 46.28        | 36.55        |       |
|       | 3                | 12.83        | 41.05        | 12.83        | 45.71        | 1.50         | 45.73        | 46.39        | 36.07        |       |
|       | 4                | 12.94        | 41.35        | 12.93        | 43.82        | 1.45         | 43.86        | 46.47        | 35.69        |       |
|       | 5                | 13.02        | 41.76        | 13.02        | 41.31        | 1.37         | 41.35        | 46.58        | 35.56        |       |
|       | 6                | 13.29        | 42.18        | 13.29        | 38.32        | 1.29         | 38.37        | 46.72        | 35.56        |       |
|       | 7                | 14.86        | 42.39        | 14.85        | 35.13        | 1.23         | 35.19        | 46.87        | 35.62        |       |
|       | 8                | 16.06        | <b>42.42</b> | 16.05        | 32.01        | 1.19         | 32.07        | 47.03        | 35.70        |       |
|       | 9                | 17.10        | 42.27        | 17.08        | 29.16        | <b>1.18</b>  | 29.20        | 47.20        | 35.70        |       |
|       | 10               | 18.00        | 41.98        | 17.99        | 26.67        | 1.20         | 26.70        | 47.38        | 35.73        |       |
|       | 12               | 19.51        | 41.33        | 19.50        | 22.62        | 1.26         | 22.63        | 47.80        | 35.85        |       |
|       | 15               | 21.46        | 40.70        | 21.45        | 17.75        | 1.35         | 17.73        | 48.54        | 35.88        |       |
|       | 18               | <b>23.21</b> | 40.18        | <b>23.19</b> | <b>13.97</b> | 1.47         | <b>13.94</b> | <b>49.28</b> | 35.82        |       |
|       | 3x3              | 0.25         | 20.55        | 42.53        | 20.51        | 19.43        | 1.12         | 19.51        | 44.39        | 35.84 |
|       |                  | 0.5          | 22.09        | 42.66        | 22.04        | 14.17        | 1.08         | 14.26        | 44.95        | 35.70 |
|       |                  | 1            | 24.38        | 42.58        | 24.30        | 9.28         | 1.06         | 9.43         | 44.73        | 35.72 |
| 2     |                  | 24.61        | 42.69        | 24.52        | 8.95         | 1.02         | 9.10         | 44.92        | 36.11        |       |
| 3     |                  | 24.55        | 42.87        | 24.47        | 8.91         | 0.99         | 9.05         | 45.02        | <b>36.16</b> |       |
| 4     |                  | 24.75        | <b>42.97</b> | 24.66        | 8.59         | <b>0.99</b>  | 8.74         | 45.15        | 36.07        |       |
| 5     |                  | 25.37        | 42.85        | 25.27        | 7.69         | 1.03         | 7.85         | 45.33        | 36.08        |       |
| 6     |                  | 26.07        | 42.66        | 25.95        | 6.56         | 1.10         | 6.74         | 45.57        | 36.15        |       |
| 7     |                  | 26.55        | 42.50        | 26.43        | 5.61         | 1.17         | 5.82         | 45.85        | 36.13        |       |
| 8     |                  | 26.79        | 42.35        | 26.67        | 5.03         | 1.24         | 5.27         | 46.14        | 35.95        |       |
| 9     |                  | <b>26.83</b> | 42.18        | <b>26.73</b> | <b>4.81</b>  | 1.32         | <b>5.07</b>  | 46.45        | 35.77        |       |
| 10    |                  | 26.72        | 41.97        | 26.65        | 4.85         | 1.40         | 5.11         | 46.75        | 35.61        |       |
| 12    |                  | 26.12        | 41.44        | 26.15        | 5.29         | 1.60         | 5.55         | 47.32        | 35.30        |       |
| 15    |                  | 24.77        | 40.36        | 24.98        | 6.24         | 1.95         | 6.42         | 48.01        | 34.82        |       |
| 18    |                  | 21.97        | 40.10        | 22.18        | 8.42         | 2.04         | 8.57         | <b>48.57</b> | 34.48        |       |
| bayer |                  | 0.25         | 9.08         | 47.84        | 9.08         | 74.77        | 0.79         | 74.77        | 44.93        | 31.08 |
|       |                  | 0.5          | 9.08         | 47.99        | 9.08         | 74.44        | 0.76         | 74.44        | 45.29        | 31.33 |
|       |                  | 1            | 9.09         | 48.17        | 9.09         | 73.64        | 0.73         | 73.64        | 45.71        | 31.67 |
|       | 2                | 9.15         | 48.05        | 9.14         | 72.11        | 0.70         | 72.11        | 45.80        | 32.64        |       |
|       | 3                | 9.62         | 48.19        | 9.62         | 70.12        | 0.68         | 70.12        | 45.77        | 33.50        |       |
|       | 4                | 10.45        | 48.63        | 10.45        | 67.47        | 0.66         | 67.47        | 45.81        | 34.10        |       |
|       | 5                | 11.06        | <b>48.69</b> | 11.06        | 64.28        | 0.65         | 64.28        | 45.90        | 34.53        |       |
|       | 6                | 11.59        | 48.61        | 11.59        | 60.63        | 0.63         | 60.63        | 46.01        | 34.76        |       |
|       | 7                | 12.04        | 48.46        | 12.04        | 56.59        | 0.62         | 56.59        | 46.16        | 34.81        |       |
|       | 8                | 12.41        | 48.28        | 12.41        | 52.32        | 0.61         | 52.32        | 46.33        | 34.89        |       |
|       | 9                | 12.70        | 48.07        | 12.70        | 48.00        | 0.59         | 48.00        | 46.52        | 34.97        |       |
|       | 10               | 12.90        | 47.82        | 12.90        | 43.78        | 0.58         | 43.78        | 46.72        | 35.08        |       |
|       | 12               | 14.55        | 47.14        | 14.55        | 36.16        | 0.57         | 36.17        | 47.15        | 35.31        |       |
|       | 15               | 17.58        | 46.22        | 17.57        | 26.44        | <b>0.56</b>  | 26.47        | 47.82        | 35.52        |       |
|       | 18               | <b>19.74</b> | 45.44        | <b>19.72</b> | <b>20.00</b> | 0.58         | <b>20.04</b> | <b>48.51</b> | <b>35.53</b> |       |

## 6.7 Conclusion

In this chapter we introduced the framework for optimization of filters for color demosaicing and spectral reflectance recovery purposes. We presented a method to approximate lengthy evaluation metrics calculations over a database of images by approximating it using the correlation matrix on the image database in reflectance. Then, we proposed SFAs with optimized spectral sensitivities of filters for various applications. However, we have artifacts in demosaiced images for SFAs having large number of filters with small correlation between them. We couldn't find a metric which correspond to these artifacts. More work needs to be done to understand the nature of these artifacts so that a cost function could be developed and filters optimized which avoid these artifacts. The optimized filters proposed are not absolute or the best filters as the solutions obtained are not unique in nature. The purpose was to develop a workflow to optimize filters as ultimately the manufacturer of filters would have their own physicals models which determine the shape of the spectral sensitivities. Using the framework proposed it could be possible to incorporate such physical models into the design process. Also, in the evaluation of metrics we saw that a major reason for color errors for small SFA sizes is not due to demosaicing but rather due to the filter space to sRGB color space transform. The proposed transform is just a linear pseudo inverse and most complicated solutions exist in the state of the art. The advantage of the linear transform is that it can be incorporated in our linear demosaicing model which allows faster optimization. However, it would be interesting to evaluate other solutions for this color transformation in the future.



# Chapter 7

## Demosaicing using Dual Layer Feedforward Neural Network

### 7.1 Introduction

In previous chapters, LMMSE (linear minimum mean square error) based algorithm was proposed which can be used to demosaic random CFAs<sup>44,95</sup>. We considered demosaicing to be an inverse problem of estimating a linear operator which inverses the effect of mosaic by learning its weights on an image database. The linear operator was further stabilized by considering neighboring pixel. The same paradigm can be extended by considering a non linear solution. A non linear solution extends the degree of freedom for the weights and therefore can give better results. Neural Networks have been shown to be good candidate for such problems and machine learning based solutions are already used extensively for image classification/recognition, pattern matching, etc<sup>104</sup>. The common idea here is that the neural network learns to recognize patterns on a huge image database.

Recently, a three layer neural network using deep learning for demosaicing<sup>105</sup> was proposed. They trained their network on 2992 downsampled images from Flickr. Gharbi et al.<sup>28</sup> have proposed a neural network based on convolutional neural networks having 15 layers training them on more than a million images. They use sRGB images which are downsampled using bicubic interpolation to avoid incorporating the

distortions caused by the image processing pipeline. They demonstrate that their network generalizes well to linear data also. By linear data they imply the RAW image. They need to make this distinction because they train on sRGB images which have been post processed, for e.g. they have gamma (which is non linear) applied. This creates a difference because they train on non-linear images however; in reality we demosaic linear RAW images. Probably, their network is able to generalize as the network trains on millions of images; the network is very generic and therefore has averaged out the differences in image processing pipeline, sRGB transform, noise removal, sharpening, jpg compression. Both these approaches train their network on already demosaiced images.

Recently there is a growing interest in development of Spectral Filter Arrays (SFAs), going beyond three color filters and even adding NIR filter on the same mosaic. IMEC has proposed SFAs with 32 color channel<sup>33</sup>, Silios<sup>32</sup> with 9, are commercial propositions. Image databases which provide ground truth images for such filters may not be available. We have image databases in reflectance domains like Finlayson<sup>17</sup>, Cave<sup>65</sup> and SCIEN<sup>19</sup> which can be used to render color images for any filter sensitivity. These databases are usually small in size, therefore it is important to consider possibility of a neural network approach which doesn't need large number of training images. Ideally, one would prefer to have large database of ground truth images available, however this is not the case for hyperspectral databases.

In this chapter, we propose a simple dual layer neural network for demosaicing which can be trained for demosaicing any random CFA. We train this network on the Kodak database<sup>15</sup> and compare results with the state of art. Also, we train a neural network on 5-band TokyoTech multispectral image database **5-BAND Tokyo**<sup>1</sup> and use it to demosaic a 5 color SFA (see Figure 7-3-h, Monno5ch SFA)<sup>20</sup>. This database is a true-color image database like the Kodak database, but instead of 3 it has 5 color channels. Further, we train a neural network on images from TokyoTech 31 band Multispectral database **31-BAND Tokyo**<sup>2</sup>. This is a hyperspectral image database

---

<sup>1</sup><http://www.ok.sc.e.titech.ac.jp/res/MSI/MSIdata.html>

<sup>2</sup><http://www.ok.sc.e.titech.ac.jp/res/MSI/MSIdata31.html>

with images from 420nm to 720nm with images every 10nm, therefore we need to first render images using filter spectral sensitivity of prototype camera<sup>21</sup> and use it to demosaic RAW images from their cameras and compare it to state of art. Similarly, we could also train our neural network on Finlayson or Cave hyperspectral database and use this to demosaic RAW images. To note the filter sensitivities here are not same as those for the 5-Band dataset, therefore it cannot be directly used. Demosaicing exercise on the 5-Band dataset serves more to compare demosaicing algorithms as ground-truth image (like Kodak) are available. The authors have already shown their approach using Weighted Guided Filter<sup>21</sup> to be better than Binary-Tree Edge Sensing algorithm<sup>29,83,84</sup> therefore we choose not to repeat results for Binary Tree. For this camera we present results both on rendered images from Hyperspectral image databases like Finlayson, Cave and TokyoTech 31-band and also on real RAW images captured with our camera. Simulations on rendered images allow us to measure metrics like PSNR as ground truth is available which allows quantitative comparison.

## 7.2 Neural Network Configuration

We previously defined a 'super-pixel' to be the basis pattern, the most basic pattern of mosaic of filters which when repeated across the surface of the sensor forming the Color Filter Array. For the Bayer the super-pixel [R G; G B] is of size 2x2. For Fujifilm XTrans CFA the super-pixel is of size 6x6. We consider demosaicing problem to be of block shift invariant, i.e. same solution is proposed for each super-pixel pattern. Now for  $h \times w$  pixels in 'super-pixel', the goal of demosaicing is to estimate  $Phw$  pixels, full color image, where  $P$  is number of color channels.  $P$  is three for RGB CFAs. When dealing with SFAs we want to estimate  $P_s h w$  pixels, where  $P_s$  is number of spectral channels.. We do this by training a neural network on a database of true color and simulated CFA images, where we fine tune the weights and bias of connecting neurons with the target of reducing the Mean Square Error. We consider a constant window of neighborhood pixels of  $n'_h$  vertical and  $n'_w$  horizontal pixels to increase the size of input. We earlier defined  $n'_h = h + n_h - 1$  and  $n'_w = w + n_w - 1$  to ensure equivalence



between constant and sliding neighborhood. Unlike the implementation for LMMSE we don't have anything like the  $S_1$  matrix as we directly reconstruct  $y$ .

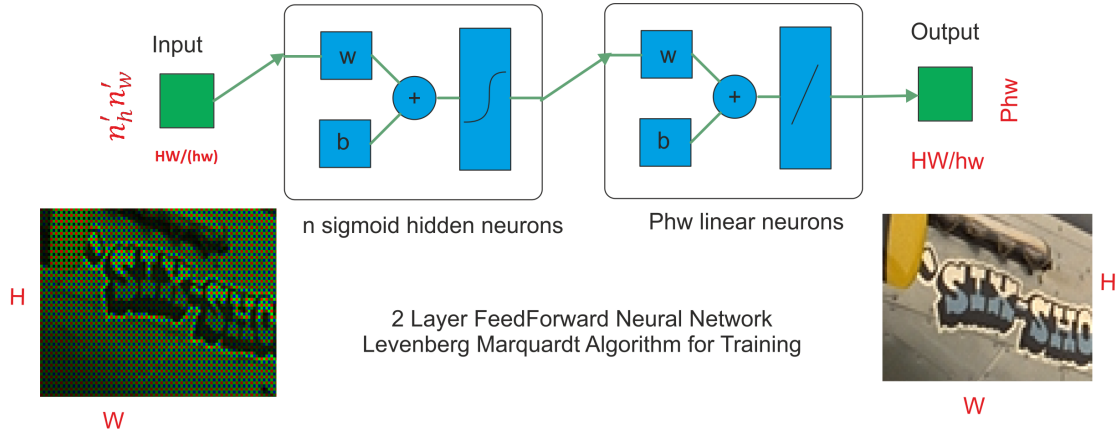


Figure 7-1: Neural Network model. For RGB CFAs,  $P = 3$ . In case of *SFAs*, we need to replace  $P$  by  $P_s$  which is number of spectral filters.

We used a two layer feed-forward neural network (see Figure 7-1) of fitting type to solve the demosaicing problem. The first layer is composed of  $n$  sigmoid neurons and the second layer of  $Phw$  linear neurons. We choose this particular configuration as we consider demosaicing to be a data fitting problem and not a pattern matching or classification problem. A radial basis configuration was also considered but rejected as it required lot of memory for training purpose.

We trained the neural network using Matlab neural networking toolbox, [Matlab](https://www.mathworks.com/products/neural-network.html)<sup>3</sup>. Training data was simulated over the Kodak image database for RGB CFAs for the sake of comparison with other state of art demosaicing algorithms. Similarly, for the Monno5ch SFA we trained on the 5-Band Multispectral dataset for comparing with state of art algorithm. For the prototype camera we trained on rendered images from hyperspectral databases. In general we used 70% of data for training of network, 15% was used for validation and 15% for testing. Data was divided randomly. This is the protocol used by default by Matlab for deciding when to stop the training. However when we present result for average PSNR and other metrics we consider 100% of the pixels for testing. We trained the network using scaled conjugate gradient method

<sup>3</sup><https://www.mathworks.com/products/neural-network.html>

with back-propagation.

## 7.3 Results

### 7.3.1 Testing for Color Filter Arrays

We trained the neural network by paralleling the task on both CPU and GPU. The system consisted of Intel i7 6700K with Nvidia GTX 1080. After training the network based on procedure described earlier we tested the network for the entire Kodak image database. For achieving a good PSNR we found there are two factors, having a big neighborhood window size and more neurons in layer number one. The training time also depends on the number of neurons. However, increasing the number of neurons greater than the size of input gives diminishing result in performance with regard to computational time increases and memory requirement. Table 7.1 shows the result for testing our Neural Network approach on the Kodak image database. All the values reported are averages across the image database. As expected for CNRS and 4x4 #2, we have a lower  $\sigma_{rgb}$  as the CFAs have quasi-equal distribution of color filters unlike Bayer or Fuji where green channel is better reconstructed compared to other two.  $\sigma$  denotes the variance of PSNR across all the images across the database, therefore a lower value indicates that all the images are well reconstructed. Figure 7-2 shows the fence region of the Lighthouse image demosaiced using our algorithm on the different CFAs. CFAs like CNRS, 4x4 #2 to some extent avoid false colors and moire due to random arrangement in the mosaic compared to the Bayer CFA. Execution times reported are for execution on CPU only.

In Table 7.1, the Bayer CFA, a neighborhood window size  $(n'_h, n'_w)$  of 10 was used with 100 neurons, the training took 20.6 hours giving a performance of 40.71dB. Using only 40 neurons with  $(n'_h, n'_w)$  of 11 gave us 40.19dB in training time of 7.6 hours. For Fuji, we used  $(n'_h, n'_w)$  of 15 with 225 neurons. For 4x4 #2,  $(n'_h, n'_w)$  of 13, with 40 neurons we have 40.32dB, 100 neurons 40.70dB, 169 neurons 41dB. The training

Table 7.1: Performance of Neural Network, expressed as averages across images in Kodak database

| CFA       | $\mu$        | SSIM          | $\Delta E$  | $\sigma_{rgb}$ | $\sigma$    | time(s) |
|-----------|--------------|---------------|-------------|----------------|-------------|---------|
| Bayer     | 40.71        | 0.9930        | <b>1.25</b> | 4.39           | <b>5.17</b> | 0.15    |
| Fuji      | 39.10        | 0.9912        | 1.53        | 3.42           | 5.61        | 0.15    |
| CNRS      | 40.01        | 0.9928        | 1.41        | <b>0.58</b>    | 5.85        | 0.15    |
| 4x4 #2    | <b>41.00</b> | <b>0.9940</b> | 1.31        | 0.77           | 5.47        | 0.22    |
| RGBW      | 40.54        | 0.9932        | 1.47        | 1.25           | 5.20        | 0.20    |
| Kodak 2.0 | 38.82        | 0.9905        | 1.76        | 2.05           | 5.46        | 0.20    |
| SonyRGBW  | 38.11        | 0.9891        | 1.86        | 2.80           | 5.20        | 0.20    |

Table 7.2: Result of Average PSNR for Kodak database for our method compared with state of art. In Chapter 3, LMMSE which were not clipped between [0 1], here we clip all results. Gharbi<sup>28</sup> is a Neural Networking approach using deep learning. Another Neural Network based approach<sup>105</sup> reports RMSE values for Kodak equivalent value for Average PSNR is 37.18 for 19 images only. For ACUDE we test Bayer our-self, while for other CFAs values reported as per their paper<sup>14</sup>.

| CFA       | Average PSNR $\mu$ |                     |                     |                      |
|-----------|--------------------|---------------------|---------------------|----------------------|
|           | Ours               | LMMSE <sup>95</sup> | ACUDE <sup>14</sup> | Gharbi <sup>28</sup> |
| Bayer     | 40.71              | 39.13               | 40.71               | <b>41.2</b>          |
| Fuji      | 39.10              | 39.03               | <b>39.54</b>        |                      |
| CNRS      | 40.01              | <b>40.03</b>        |                     |                      |
| 4x4 #2    | <b>41.00</b>       | 40.68               |                     |                      |
| RGBW      | <b>40.54</b>       | 39.74               |                     |                      |
| Kodak 2.0 | <b>38.82</b>       | 38.43               | 38.70               |                      |
| SonyRGBW  | <b>38.11</b>       | 37.38               | 38.10               |                      |

time varies from 3.7 hours to 16.9 hours. As the training time is quite long, it is not possible to do a complete leave one out testing.

Depending on application we can reduce the number of neurons / neighborhood to slightly reduce Average PSNR in order to gain computational speed.

Table 7.2 shows the result for average PSNR ( $\mu$ ) for our method compared with the state of art. Our Neural Network gives a peak performance of 40.71dB using 2 layers only compared to 41.2 dB for the 15 layer Gharbi’s Neural Network. It outperforms LMMSE in  $\mu$  however it is slower than it in execution speed. Compared to ACUDE it is slightly better for RGB plus panchromatic (White pixel) CFAs. Here the white pixel is simulated as the linear combination of RGB pixels.

LMMSE is a linear solution and it takes 0.10s/image using constant neighborhood.

Gharbi NN<sup>28</sup> report 2.9s per Mpixel which corresponds to roughly 1.14s for a Kodak image. However, they use a slower processor which has 14% lower clock speed, so we extrapolate to 0.98s per image. For sure they use 15 layers while we use only 2 layers, so ours will be faster. For ACUDE, with the code publicly available it takes approximately 1.6 hour to process a single image with Bayer CFA, however the authors claim on their website that they can process in under 1s per image [ACUDE](http://www.eecs.qmul.ac.uk/~phao/CFA/acude/)<sup>4</sup>.

For Bayer despite having PSNR higher than 40dB, both ours Neural Network and ACUDE exhibit false colors in the fence part of Lighthouse image. Gharbi et al. don't present the result. Only LLSC<sup>27</sup> avoids that, it has an average PSNR of 41.46dB, however, it takes approximately 6 minutes per image (on Xeon e5 1620, note the compiled code is single threaded so potential to improve the timing). We do not know how to adapt their algorithm for CFAs other than Bayer; therefore we do not present the result in the Table 7.2.



Figure 7-2: Lighthouse Image crop. Demosaiced using Neural Network method. (Top Row: Left to Right) Original, Bayer, Fuji, CNRS. (Bottom Row: Left to Right) 4x4 # 2, RGBW, Kodak 2.0, Sony RGBW.

<sup>4</sup><http://www.eecs.qmul.ac.uk/~phao/CFA/acude/>

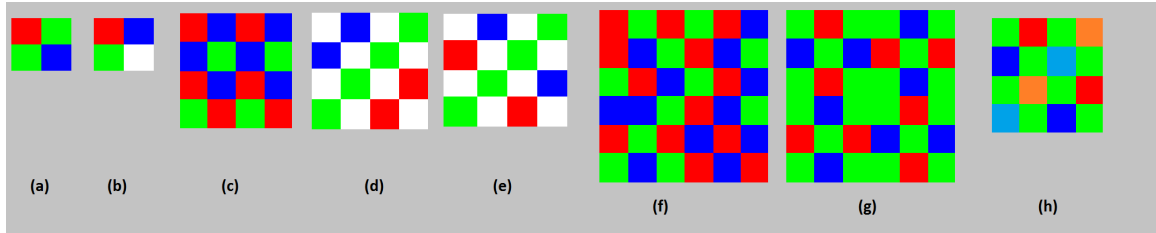


Figure 7-3: CFAs and SFA tested. (a) Bayer, (b) RGBW, (c) 4x4 # 2, (d) Kodak 2.0<sup>106</sup>, (e) Sony RGBW<sup>107</sup>, (f) CNRS<sup>52</sup>, (g) Fuji<sup>25</sup>, (h) Monno 5ch SFA<sup>21</sup>

### 7.3.2 Testing for Spectral Filter Arrays

We consider the Monno5ch SFA<sup>20</sup>, see Figure 7-3-h as proposed by Monno et al. We consider two cases, first with training and simulation on the 5-band multispectral dataset as provided by the authors. This is similar to the training in the chapter on Random RGB CFAs as all the color channels are present and the mosaiced image can be directly simulated. So, this is training using the color spatial image formation model. Secondly, we consider the real camera as developed by the authors where we demosaic RAW images. For this we consider the spectral image formation model and we train on the Finlayson image database (we also trained on Cave or the TokyoTech-31 database). We present results on the RAW images and also on rendered images from these databases.

### 7.3.3 Monno5ch SFA with images from 5-band multispectral dataset

For the Monno5ch SFA, we trained the Neural Network on the 5-band multispectral dataset.<sup>20</sup> We have about 147 million pixels of color data available. Training on the entire dataset would be very long. Therefore, we selected the first 6.25% of pixels from the database for training purpose. Of this again there was (70%, 15%, 15%) re-partition between training, validation and testing. Finally, we report results for averaging the PSNR for all the images for the 5 bands, see Table 7.3 using 100% of pixels. We achieved an average PSNR of 44.72dB for a neighborhood window size of 10, using 100 neurons. It took 1.77 hours to train this network. Increasing the

number of neurons or window size didn't yield better results. Training over entire database might give better results, however we are limited by memory considerations as of now. We found that LMMSE has the best performance for this dataset in terms of Average PSNR. Figure 7-4 and 7-5 show crops of demosaiced images converted from 5 channels to sRGB for the proposed Neural Network method, LMMSE and WGF method. The demosaiced images are 5 channels, for displaying them we first need to convert them to sRGB domain for which we used the transform as provided by Monno et al. It is difficult to visually make out much difference between the three. However it seems MSRI has a slight advantage, less false colors in reflection of light in the toy car for instance.

Table 7.3: Result of Average PSNR across 5 channel and time for our method compared with state of art for Monno5ch SFA for 5 channel multispectral Tokyo dataset.

| Monno5ch SFA TokyoTech 5-band |               |                     |                     |                   |
|-------------------------------|---------------|---------------------|---------------------|-------------------|
|                               | Ours          | LMMSE <sup>95</sup> | MSRI <sup>108</sup> | WGF <sup>21</sup> |
| PNSR                          | 44.72         | <b>45.16</b>        | 44.45               | 43.11             |
| SSIM                          | <b>0.9946</b> | 0.9945              | 0.9942              | 0.9923            |
| $\sigma_{5ch}$                | 9.74          | 12.91               | 9.74                | <b>8.50</b>       |
| $\sigma$                      | <b>9.40</b>   | 12.40               | 17.50               | 16.30             |
| Time(s)                       | 0.78          | <b>0.46</b>         | 14.63               | 31.21             |



Figure 7-4: Demosaiced Image from 5-band TokyoTech set data. Left to Right: Ours NN, LMMSE, MSRI



Figure 7-5: Demosaiced Image from 5-band TokyoTech set data. Left to Right: Ours NN, LMMSE, MSRI

### 7.3.4 Monno5ch SFA with RAW images captured from actual camera

Further Monno et al. implemented the SFA physically and they have provided the spectral sensitivity of the realized filters and RAW images obtained from this camera system<sup>21</sup>. The sensitivities of these physical filters differ from 5-band multispectral database which was used earlier; therefore Neural Network trained on this dataset is no longer useful. They have also shared a 31-band Multispectral image database. Similar hyperspectral image databases like Finlayson<sup>17</sup> and Cave<sup>65</sup> are available. We can render full resolution 5 channel images using filter sensitivities for this camera using the image formation model described in previous chapter. To recall, the full color image  $Y$  is the product of illuminant  $L$  multiplied by scene reflectance  $Z$ , rendered by the filter sensitivities  $F_{Ps}$ , i.e.  $Y = F_{Ps}LZ$ , where  $Ps$  is number of spectral channels. This full color image is subsampled by the mosaic  $M$  to give the SFA image  $X$ . For the Neural Network training  $X$  is used as input and  $Y$  as output. We generate  $X$  and  $Y$  from a hyperspectral image database.

Table 7.4 shows the evaluation of metrics on rendered images from the Finlayson, Cave and TokyoTech 31-band hyperspectral database. We use only WGF method for evaluation as this is what the authors use to present results from RAW camera and not MSRI. The values are for 5 channel demosaiced images, no post processing like gamma is applied here. We find that Neural Network outperforms the rest in 2 out of 3 image databases. Also, the WGF method has very high  $\sigma$ , i.e. variance of PSNR

across all images, which implies that some of the images are very well reconstructed while other not so much.

Figure 7-6, 7-7 and 7-8 shows the comparison of three approaches on RAW image captured by this camera. With Ours and LMMSE although the output image is sharper with less false colors but artifacts are present. A textured pattern is present in flat regions of the scene. These artifacts can be removed by application of a small median filter, see figure 7-9. This is a simple solution just for illustration, using more complicated algorithms may preserve edges. A filter to sRGB space transform as used by Monno et al. was applied for displaying images.

To demonstrate generality of our training and method we show results of demosaicing RAW images where learning was performed on Finlayson hyperspectral database. We have similar results if training is done on Cave or TokyoTech-31 dataset.



Figure 7-6: Demosaiced RAW Image from Monno5ch SFA. Left to Right: Ours NN, LMMSE, WGF. Notice Text is sharper with NN and LMMSE, there are less false colors however there are artifacts in flat regions.



Figure 7-7: Demosaiced RAW Image from Monno5ch SFA camera. Left to Right: Ours NN, LMMSE, WGF. Notice less false colors with NN and LMMSE, however there are artifacts in flat regions.



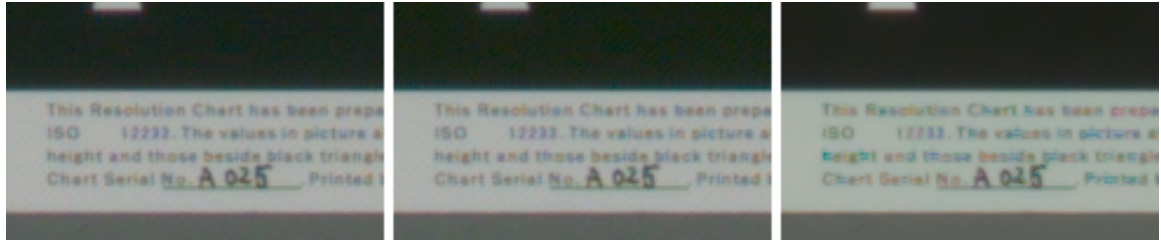


Figure 7-8: Demosaiced RAW Image from Monno5ch SFA camera. Left to Right: Ours NN, LMMSE, WGF. Notice Text is sharper with NN and LMMSE, there are less false colors however there are artifacts in flat regions.



Figure 7-9: Demosaiced RAW Image from Monno5ch SFA camera. Left to Right: Ours NN with 3x3 Median filter applied to remove artifacts, WGF

## 7.4 Discussion

For the Kodak image database, our neural network is mostly better than LMMSE approach. This is probably because being non linear it had more degree of freedom to adapt to the training data. However, this come at the cost of being slightly slower. It is still several magnitudes faster than other state of art algorithms like ACUDE<sup>14</sup> or the Neural Network approach by Gharbi<sup>28</sup>. Unlike Gharbi which gains 0.5dB on PSNR, our approach doesn't need to train on millions of images while being twice as fast in demosaicing. Both still have false colors in high frequency areas. Compared to other state of art algorithms it is generally better, especially for CFAs with white pixels.

Comparing for the Monno 5 channel SFA, with images coming from 5-band multi-spectral dataset, in terms of PSNR, LMMSE gives the best performance. Visually it is difficult to evaluate, however it appears than MSRI<sup>109</sup> has a very slight advantage

Table 7.4: Result of Average PSNR across 5 channel and time for our method compared with state of art for Monno5ch SFA with filter sensitivities as implemented simulated on Finlayson, Cave and TokyoTech 31 database. Illuminant is emuda5 as measured.

| Monno5ch SFA RAW camera |                |               |                     |                   |
|-------------------------|----------------|---------------|---------------------|-------------------|
|                         |                | Ours          | LMMSE <sup>95</sup> | WGF <sup>21</sup> |
| Finlayson               | PNSR           | <b>41.63</b>  | 41.42               | 39.90             |
|                         | SSIM           | <b>0.9919</b> | 0.9918              | 0.9902            |
|                         | $\sigma_{5ch}$ | 6.10          | 6.27                | <b>1.98</b>       |
|                         | $\sigma$       | 4.36          | <b>4.19</b>         | 10.31             |
|                         | Time(s)        | 0.09          | <b>0.004</b>        | 1.34              |
| Cave                    | PNSR           | 47.68         | <b>48.20</b>        | 45.32             |
|                         | SSIM           | 0.9964        | <b>0.9967</b>       | 0.9942            |
|                         | $\sigma_{5ch}$ | 4.23          | 4.73                | <b>1.57</b>       |
|                         | $\sigma$       | <b>15.20</b>  | 15.80               | 21.40             |
|                         | Time(s)        | 0.12          | <b>0.01</b>         | 3.40              |
| TokyoTech 31            | PNSR           | <b>45.78</b>  | 45.17               | 44.70             |
|                         | SSIM           | <b>0.9956</b> | 0.9950              | 0.9948            |
|                         | $\sigma_{5ch}$ | 4.29          | 4.26                | <b>4.11</b>       |
|                         | $\sigma$       | 27.80         | <b>20.60</b>        | 40.06             |
|                         | Time(s)        | 0.17          | <b>0.01</b>         | 4.93              |

over false colors. However this comes at a cost of execution time, MSRI being considerably slower. One of the limitation with this study was for Neural Network training only 6.25% of image database was used for training, while 100% is used for LMMSE, due to memory constraints. Probably there is potential with Neural Network to do better.

For the RAW images from the 5 channel SFA camera, both Neural Network and LMMSE gives artifacts in flat regions of the image. However, compared to the algorithm proposed by Monno et al. the output image is sharper with less false colors. Especially if we note the text in the images, we see it is more readable. The artifacts are problematic, by using simple post processing one can remove them at cost of image sharpness. This shows that there is still potential to do better image demosaicing for such SFAs, as information is present in the RAW image. One thing to remember is that the algorithm Weighted Guided Filter proposed by Monno et al. was jointly developed for this SFA arrangement while the LMMSE and Neural Network approach is generic in nature. Further, LMMSE has an advantage over Neural Network approach,

that it is twice as fast. However, for demanding applications where image quality is paramount Neural Network affords more flexibility as by increasing the number of neurons and layers we can further improve the result.

## 7.5 Conclusion

We presented a dual-layer Neural Networking approach to demosaicing. We presented demosaicing results for both RGB Color Filter Arrays and Spectral Filter Array having five channels. We showed that our approach gives competitive results compared to state of art for RGB and RGB with panchromatic filters. For Monno5ch SFA the output image is sharper but with artifacts, although we can post-process out these artifacts. Therefore we have demonstrated that there is potential to do better as information is present in the RAW image and demosaicing for SFAs is still an interesting problem. The solution proposed by us is a relatively simple Neural Network and probably by considering more neurons or adding layers, considering different topologies may give better result which is a work to be pursued in future.

# Chapter 8

## Conclusion

### 8.1 Summary

Industrial implementations for capturing color images favor utilizing regular Color Filter Arrays (CFA) like Bayer as they are apparently easy to demosaic. The problem of false color and artifacts is overcome by use of increasing complex algorithms. However, nature favors randomness and the HVS is composed of random mosaic of cones<sup>1</sup>. In this thesis, we developed a demosaicing algorithm based on LMMSE which works with any random arrangement of color filters. This solution is based on defining the inverse problem as linear in nature and using correlation matrices to estimate the demosaicing operator which minimizes the MSE between the original and the estimated full color image. We stabilized the linear solution by considering a large neighborhood to ensure that the linear system is over determined and trained the spatio-spectral model on an image database. Our method is so fast that we can demonstrate through systematic evaluation of all arrangements of colors in a super-pixel of sizes 2,3 and 4, that random CFAs performed better than regular CFAs. Our algorithm is computationally simple and compares well with state of art solutions. We also proposed optimized CFAs namely 4x4 #1 and 4x4 #2 which are optimized for LMMSE (Figure 8-1).

We demonstrated that using a linear solution considering neighborhood pixels improves the performance and makes it comparable to non-linear solutions. We proposed

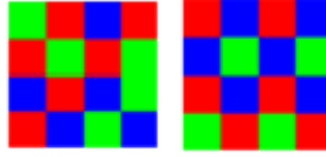


Figure 8-1: Proposed 4x4 CFA

two methods sliding and constant for forming the neighborhood matrices and showed them to be equivalent when the same spatial extent of pixels are considered. The proposed constant method reduces the memory footprint and the computational requirement considerably which is beneficial especially when considering large neighborhood and big super-pixel sizes. LMMSE method is generic and fast, suitable for real-time operations like those required by embedded systems. Our study on comparison with several state of art algorithm showed that PSNR is not satisfactory in determining the usability/performance of an algorithm. We found that artifacts/false colors may be present in images reporting higher PSNR and vice-versa which is counter-intuitive. Also SSIM differences are very small for good performing algorithm to be of much use in differentiation. Therefore we feel the need of searching a metric which is more representative of color noise and artifacts.

Further we developed a method of estimating the MSE by correlation matrix which enabled us to quickly evaluate any CFA arrangement for performance metrics. This formulation was used to propose CFAs considering new colors filter which are linear combination of RGB filters by solving an optimization problem for finding the best mixing matrix. In the state of art the method of designing optimized CFAs is by frequency selection wherein the position of luminance and chrominance is played with to easily separate the two. Our proposed CFAs cannot be demosaiced by frequency selection method as for them frequency separation by filtering is not at all possible.

We then considered a more general image formation model in the spectral domain and developed our demosaicing methodology to be able to demosaic any spatial-spectral sampling of Spectral Filter Arrays (SFAs). SFAs due to having more independence in the number and choice of filter spectral sensitivities present an interesting

challenge for demosaicing algorithms. Although demosaicing algorithms in state of art have been extended from the RGB domain, more effort has been towards SFAs which respect the Binary-Tree condition<sup>29</sup> of filter arrangement. In this thesis, the algorithm we proposed is generic so it works for any kind of SFA and we demonstrated that results are better for several different SFAs. Even for SFAs having Visible + NIR filter our algorithm gives results which are sharper with less false colors however with some block artifacts. We also proposed usage of no-reference metrics like BLIIND-II and BRISQUE for evaluating demosaiced RAW images. This is in response to our study on RGB CFAs which showed the limitations of PSNR and SSIM. Even BLIIND-II and BRISQUE was not found appropriate in this case, especially for the NIR channel. For the moment PSNR along with visual inspections remains the metric of choice.

Further, we developed the workflow for optimizing spectral filter sensitivities for different requirements, sRGB color reconstruction, demosaicing performance or spectral signature reconstruction. We defined performance metrics estimation by evaluating the correlation matrix directly on a given hyperspectral database. For simulation purposes we consider gaussian shaped filters as approximation of Faber-perot process and proposed optimized filters for different super-pixel sizes for different metrics chosen.

Finally, we developed a neural networking approach for demosaicing by considering a simple two layer feed-forward network which evaluates the super-pixel with neighboring pixels and gives results better than LMMSE. For sure a non-linear solution will provide a better solution to the problem of inverting the mosaicking operation. We demonstrated results both for RGB CFAs and also 5 color SFA as proposed by Monno et al<sup>21</sup>. We showed that our method gives higher PSNR and gain resolution at the cost of artifacts.

## 8.2 Future Work

Although LMMSE outperforms other state of art algorithms for the different SFAs we tested, however we still have block/zipper artifacts. These artifacts tend to appear when the number of filters is large. The presence/absence of these artifacts couldn't be correlated with PSNR. Therefore it would be interesting to find a metric which could quantify these artifacts. Secondly one needs to think of a way to compensate for them. The straightforward solution is to do a median filtering however this tends to negate the resolution gain due to LMMSE demosaicing. In the literature, for linear inverse problems, Tikhonov regularization is proposed to smoothen the results. We considered this by incorporating it into our demosaicing operator  $D$ . Although the results we got were artifact free but they also got de-saturated. This could be due to mixing of spatial-spectral sampling due to neighborhood consideration in our  $D$ . Therefore the regularization operation needs to be further researched.

For the LMMSE solution we proposed it could be interesting to propose an edge aware demosaicing operation, i.e. to have two demosaicing matrices  $D_H$  and  $D_W$ , along height and width, which are employed according to presence of an edge which could further improve the results.

For filter space to sRGB color space conversion for the moment we consider a simple linear pseudo-inverse operation. The advantage of this is that being linear it can be incorporated into our demosaicing chain and used for optimization procedure also. However non-linear solutions for this do exist and need to be evaluated, especially for the consideration of filter optimization where the final goal is color reproduction. Further white-balancing is actually shifting the relative response of the filters for the *sRGB* rendering. One needs to research the effect of white-balancing in the filter optimization workflow for the *sRGB* images.

The Neural Networking solution we presented was trained using the constant neighborhood. For the moment we are limited by the large number of neurons required to test sliding neighborhood however in future it could be an interesting exercise. Also the neural networking solution needs to be exhaustively studied as the relation been

the number of layers, neurons, neighborhood size, etc. is not clear unlike LMMSE. At the moment it is also not possible to propose optimized filter arrangement for the Neural Networking approach as the training is depending on the mosaic and not directly on the true-color database like for LMMSE. If possible to find something similar to  $M_1$  and  $S_1$  for Neural Networking workflow it would make the solution more universal. Finally the demosaicing model we presented in this thesis, assumes that the spatial modeling, like the Point Spread Function (PSF) between the training database and RAW images from a camera is the same. However in reality it is not the case. Therefore it would be interesting to model this spatial sampling area and incorporate into our image formation model.



# Appendices

# Appendix A

## Image databases used for training

Throughout the thesis several standard image databases are used for either training, rendering or testing, etc. Below you will find the images from the databases as small thumbnails to illustrate the data used. We would like to thank all the authors who have provided these images.

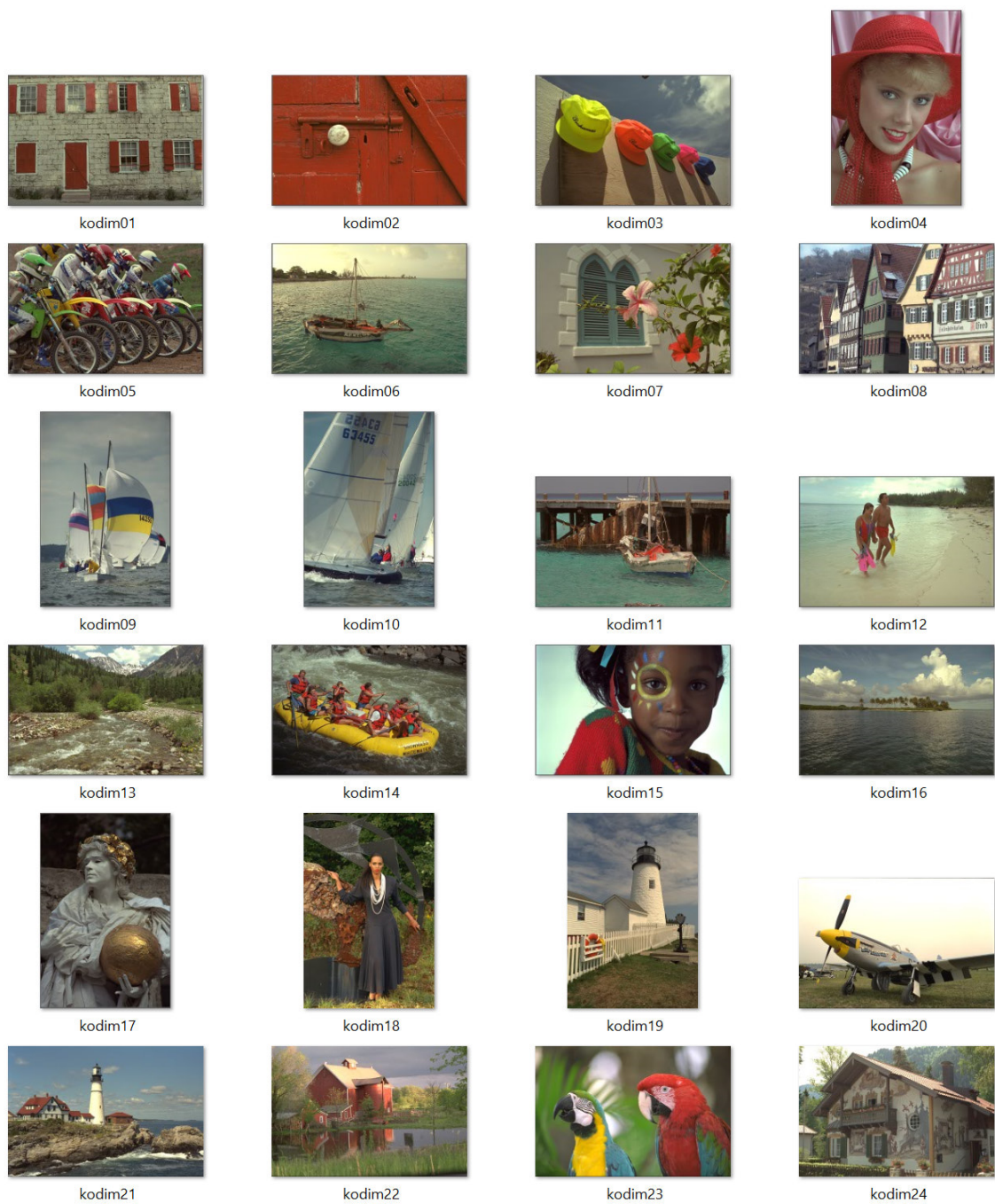


Figure A-1: Kodak<sup>15</sup> image database

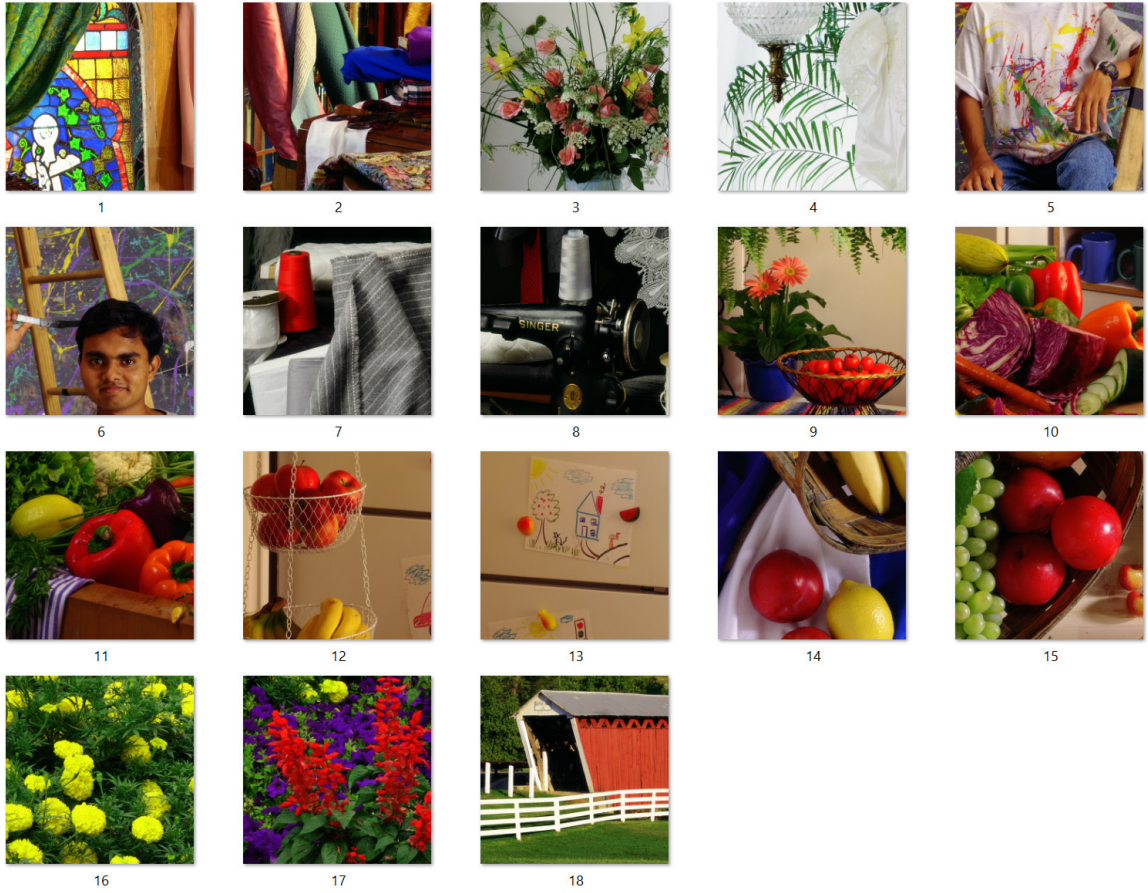


Figure A-2: McM<sup>16</sup> image database

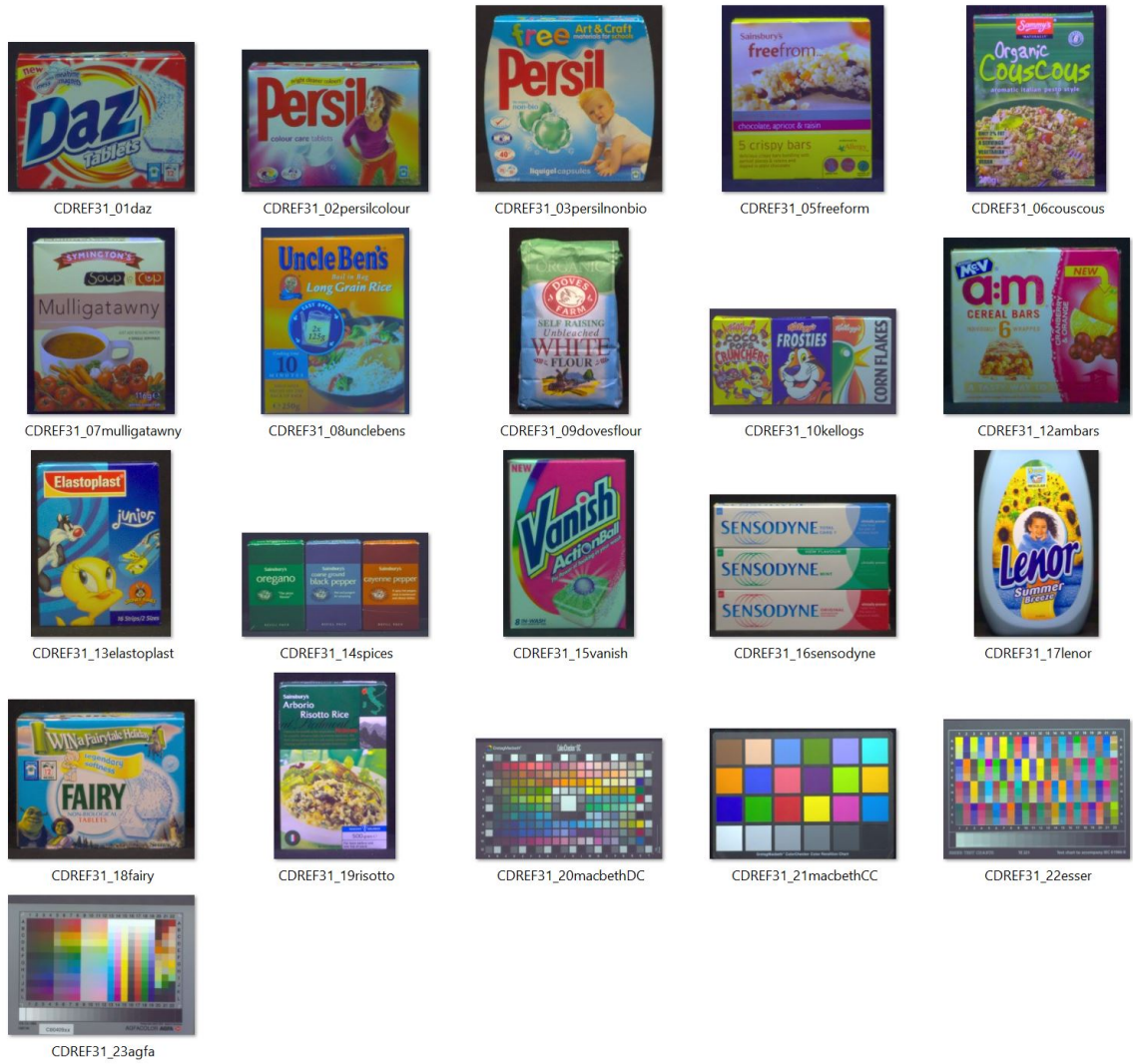


Figure A-3: Finlyason<sup>17</sup> hyperspectral database images 400nm to 700nm, every 10nm rendered for Nikon D300 using D65 illuminant. At the time of testing Image no 4 (goaheadbars) and 11(twinings) couldn't be downloaded or wasn't readable.

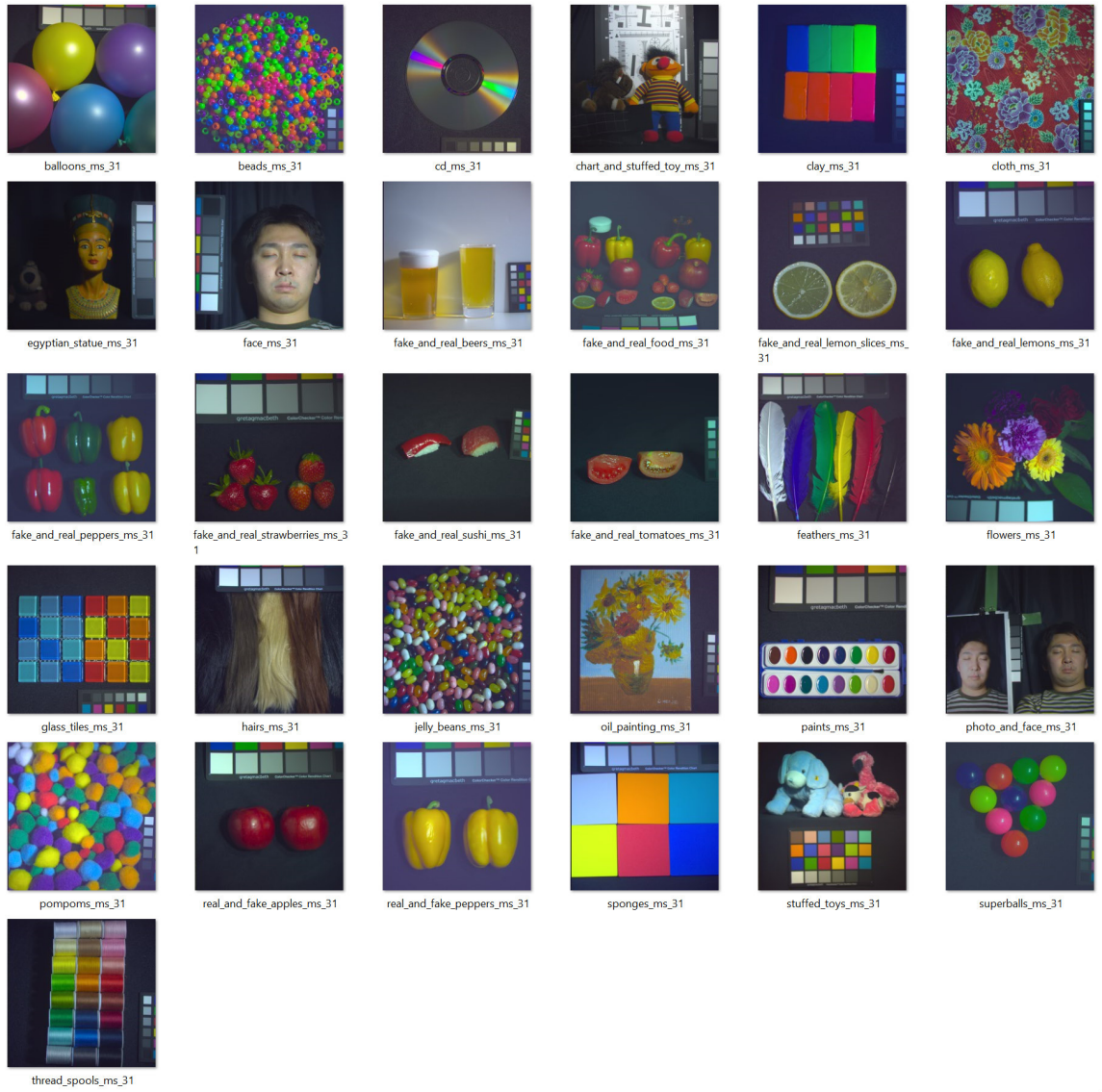


Figure A-4: Cave<sup>18</sup> hyperspectral database images 400nm to 700nm, every 10nm rendered for Nikon D300 using D65 illuminant.



Figure A-5: SCIEN<sup>19</sup> hyperspectral database without polarising filter images from 414.72nm to 950.49nm, every 3.64nm rendered for Fuji XPro using D65 illuminant.

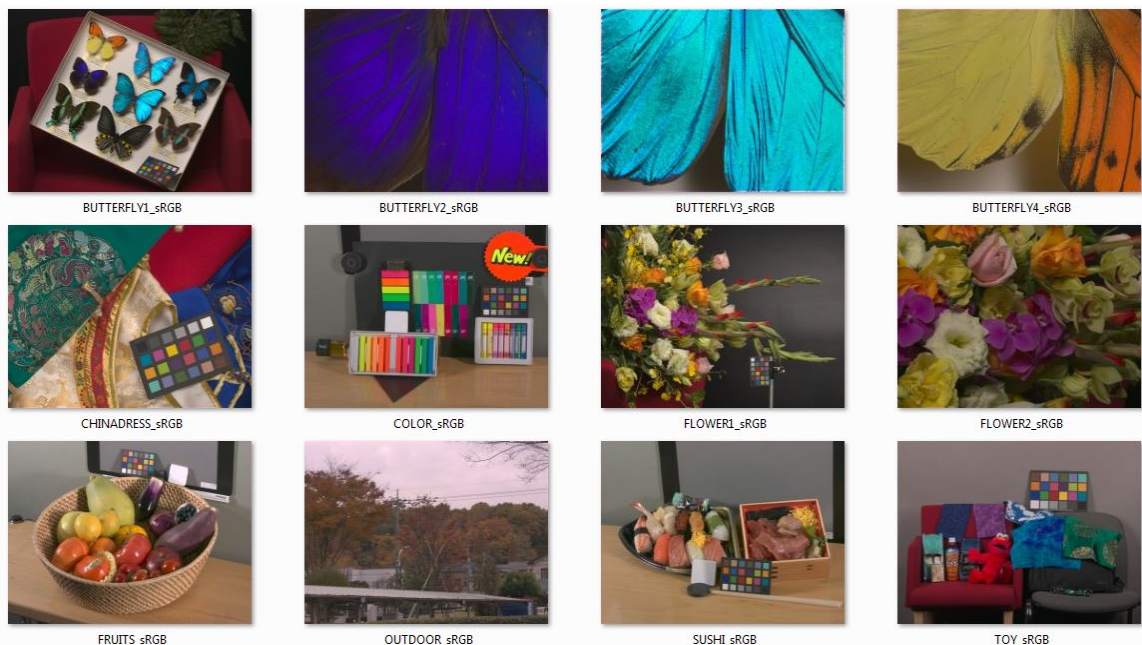


Figure A-6: TokyoTech 5-band multispectral database<sup>20</sup> images rendered for Nikon D300 using D65 illuminant.

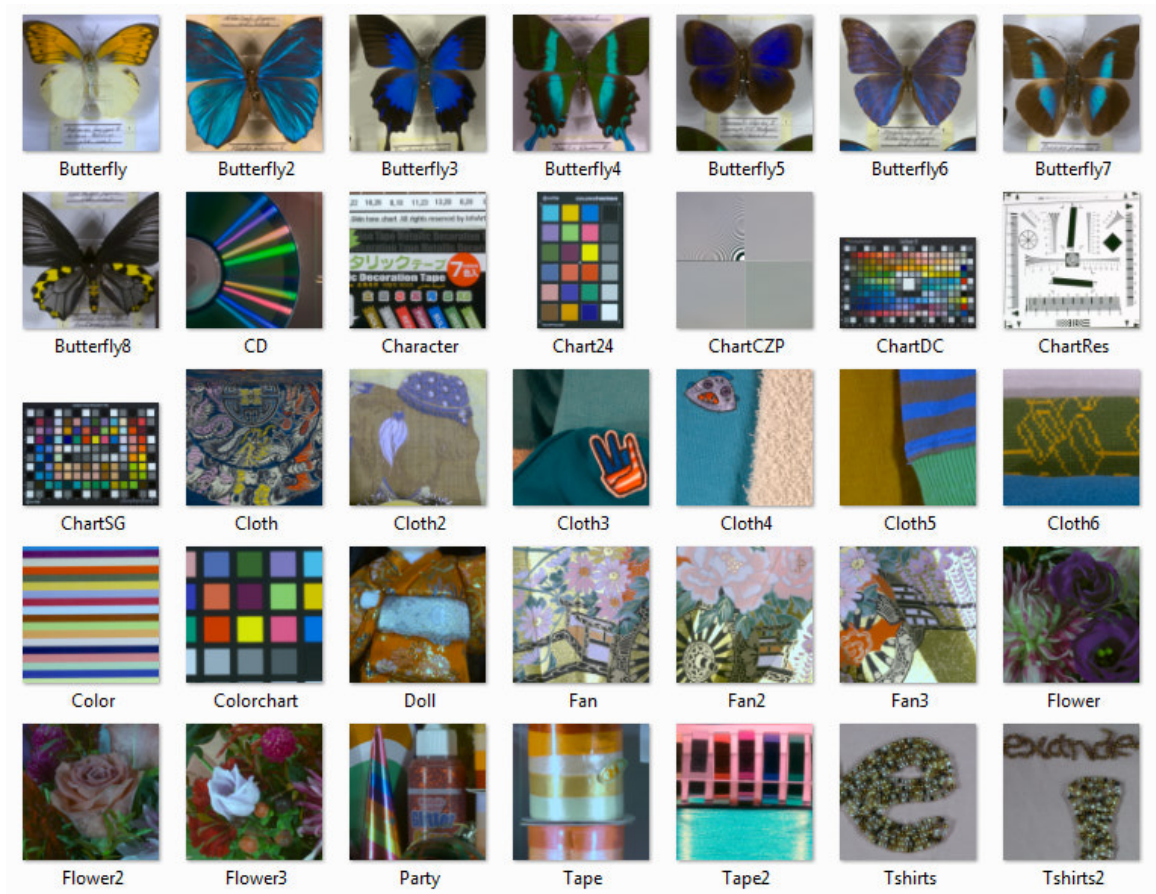


Figure A-7: TokyoTech 31-band hyperspectral database 420nm to 720nm, every 10nm<sup>21</sup> images rendered for Nikon D300 using D65 illuminant.





# Appendix B

## Demosaicing comparison on CFAs

### B.1 LMMSE on Different CFAs

Table B.1: Ordering of images B-1 B-2

|          |       |                   |       |        |
|----------|-------|-------------------|-------|--------|
| Original | Bayer | 2x2#1             | 2x2#2 | 2x2#3  |
| 3x3#1    | 3x3#2 | 3x3#3             | 4x4#1 | 4x4#2  |
| Yamanaka | Lukac | Holladay halftone | CNRS  | Condat |

### B.2 Comparison of LMMSE with other Demosaicing algorithms

Table B.2: Ordering of images B-3 B-4

|          |         |       |          |
|----------|---------|-------|----------|
| Original | LPAici  | Lian  | DA       |
| HD       | SA      | DFPD  | DLMSSE   |
| AP       | LI      | HA    | Bilinear |
| LLSC     | LDI NAT | NLM   | WECD     |
| LU       | 4x4#1   | 4x4#2 | ACUDE    |

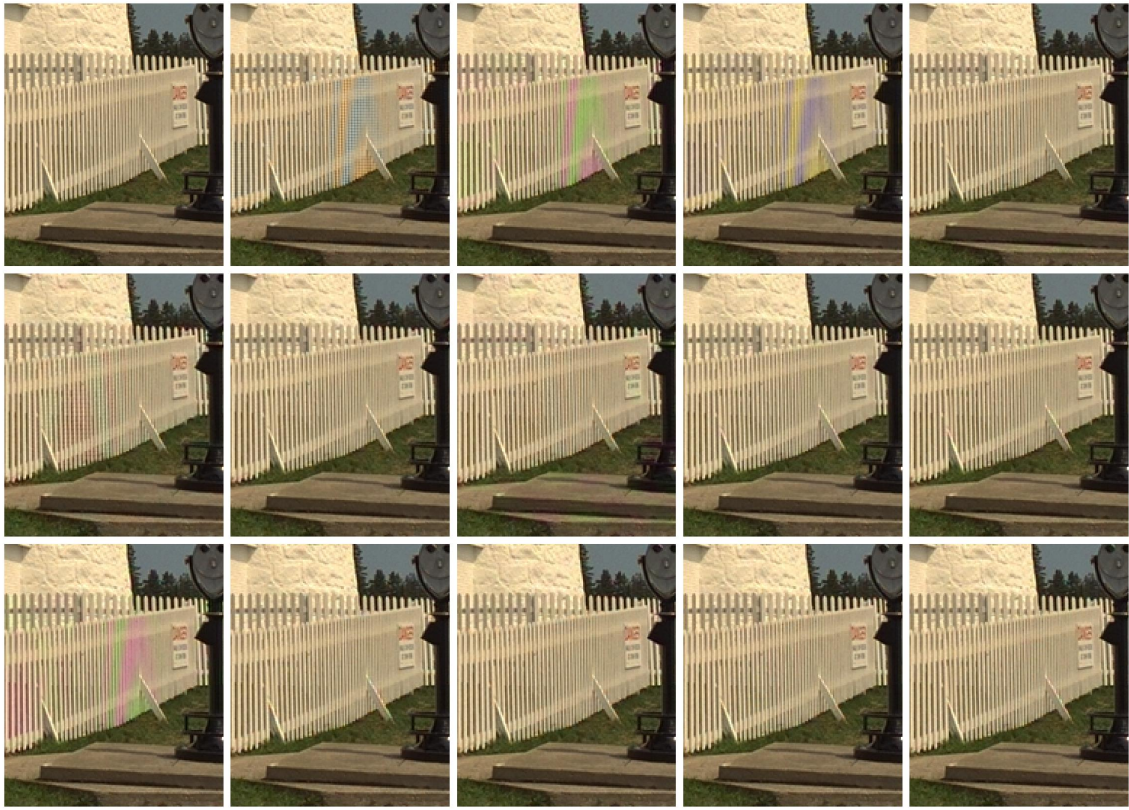


Figure B-1: LMMSE on Different CFAs for Kodak



Figure B-2: LMMSE on Different CFAs for McM

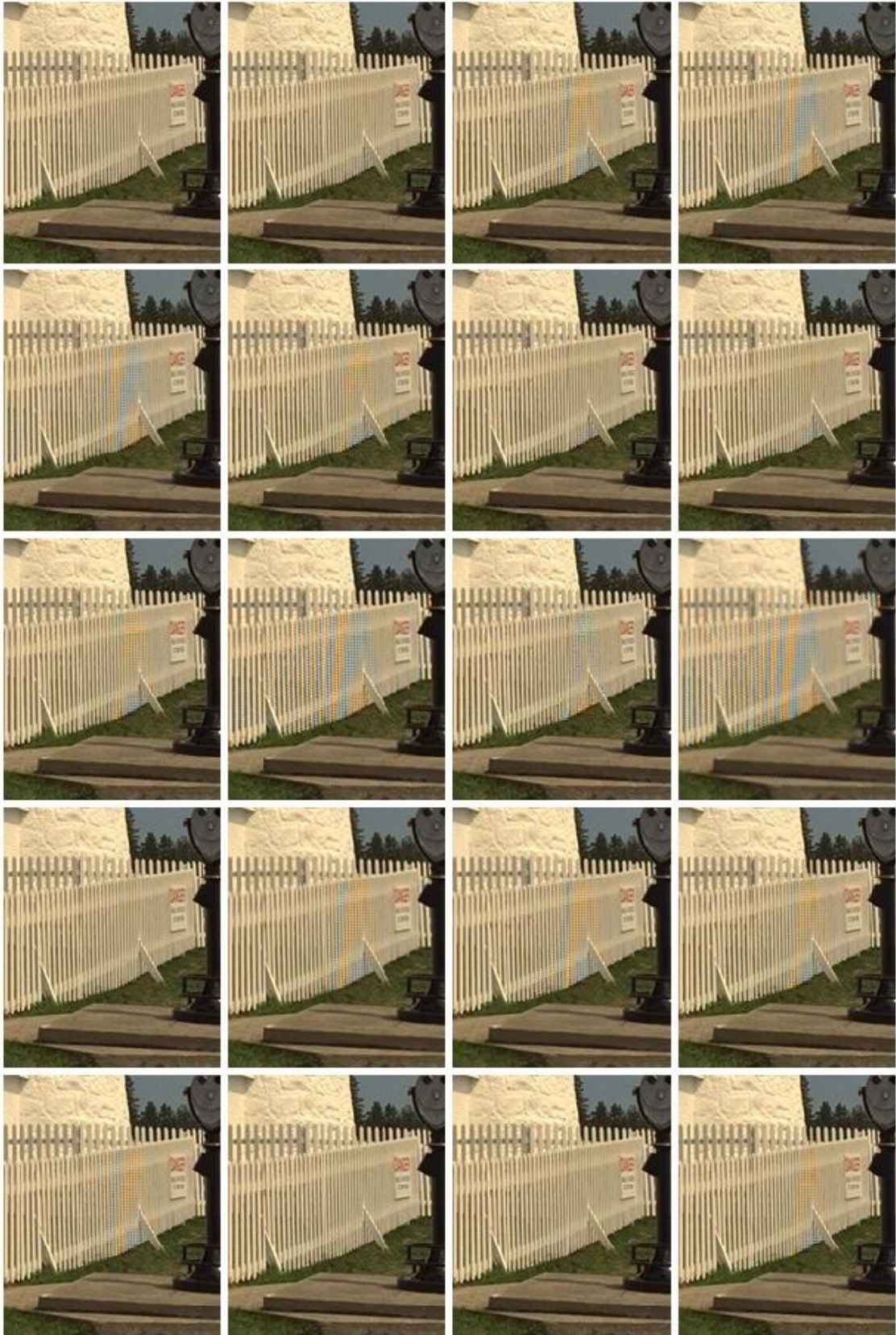


Figure B-3: Comparison of LMMSE with other Demosaicing algorithms for Kodak

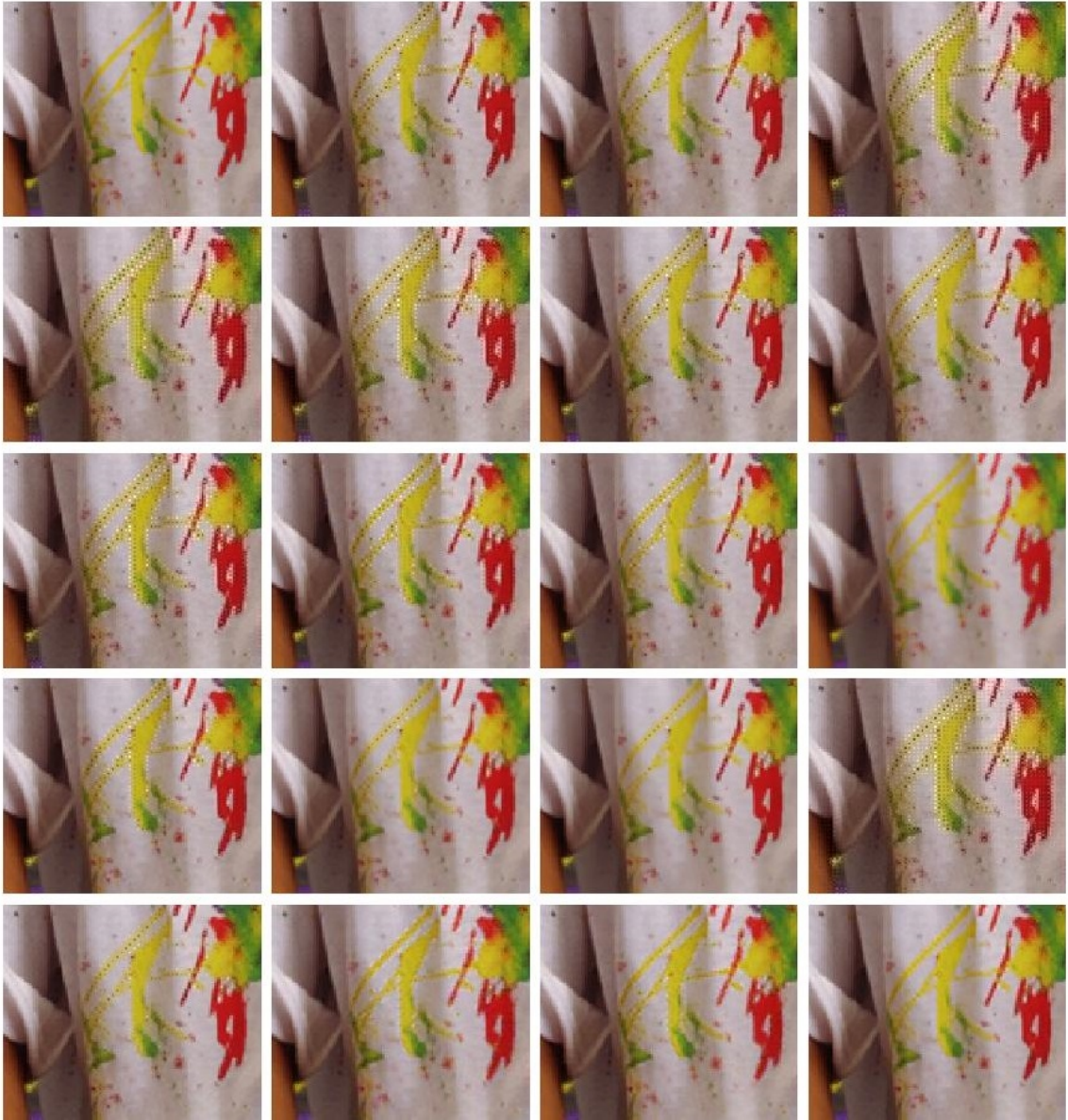


Figure B-4: Comparison of LMMSE with other Demosaicing algorithms for McM

### B.3 Effect of Leave one out

Left to Right Bayer, 3x3#2, 4x4#1, 4x4#2: Top row: when learning entire database;  
Bottom row: without learning Lighthouse image. See figure B-5 and B-6

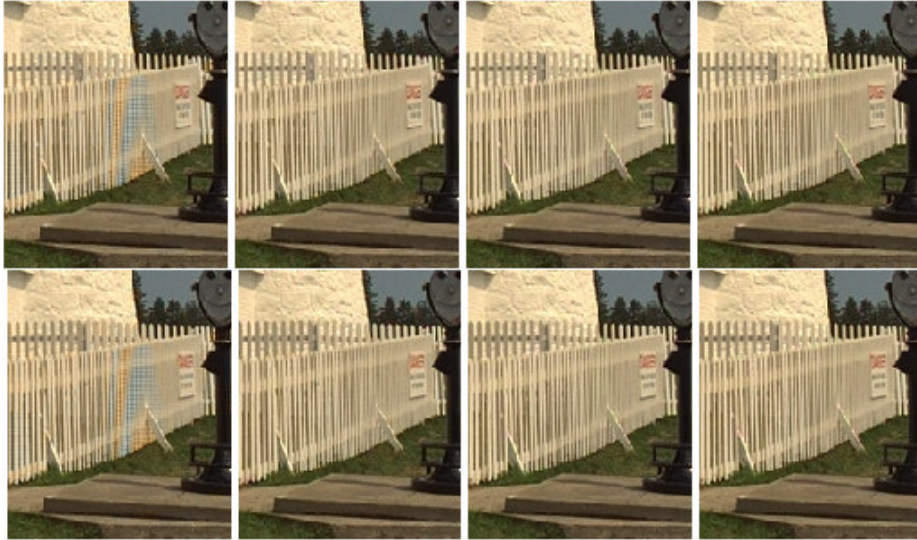


Figure B-5: Effect of Leave one out

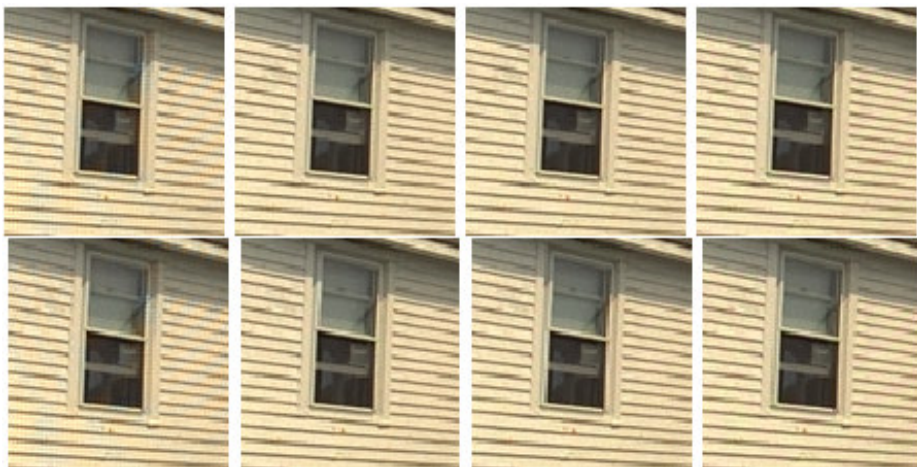


Figure B-6: Effect of Leave one out

## B.4 DFTs on Different CFAs

Discrete Fourier Transform (DFT) on images is used to transform an image from spatial domain to frequency domain. It is calculated by multiplying each spatial point in the image by the basis functions (which are sine and cosine waves) and summing the result.

Table B.3: Ordering of images B-7

|       |          |        |                   |
|-------|----------|--------|-------------------|
| Bayer | 2x2#1    | 2x2#2  | 2x2#3             |
| 3x3#1 | 3x3#2    | 3x3#3  | 4x4#1             |
| 4x4#2 | Yamanaka | Lukac  | Holladay halftone |
| CNRS  | Fuji     | Condat | Totally Random    |

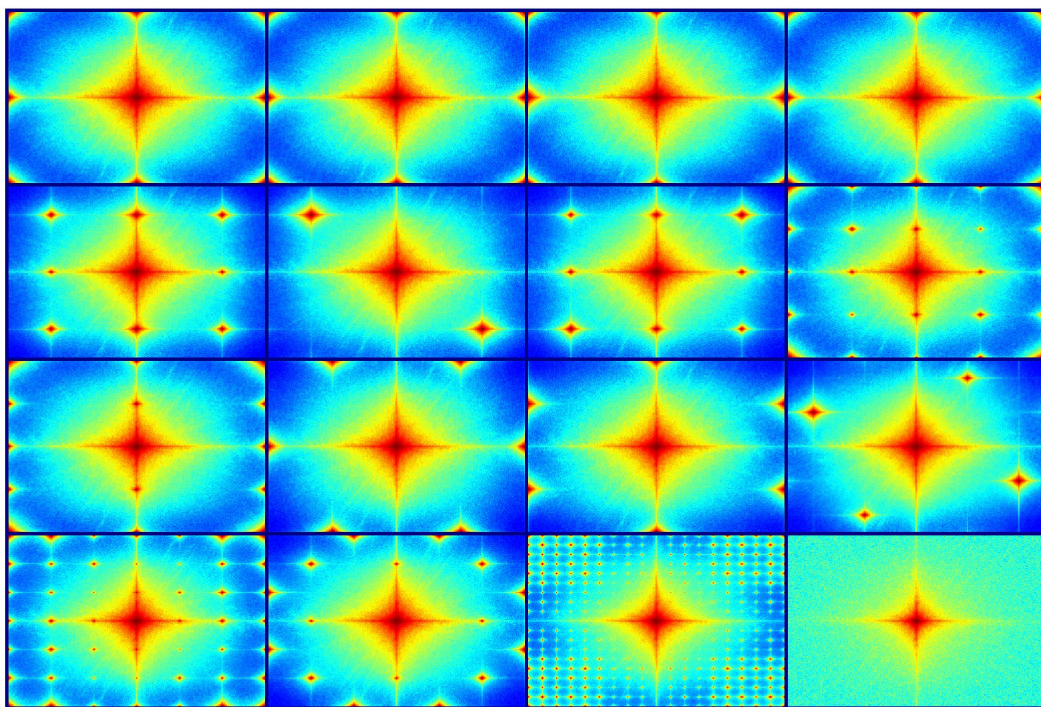


Figure B-7: FFT on different CFAs. Refer to table B.3 for ordering

# Appendix C

## Spectral Filter Arrays

### C.1 Estimating Sensor Spectral Sensitivity

Let us consider  $I$  to be an image recorded by a sensor of filter sensitivity  $F$  when illuminated with a light source of  $L$ . Then  $F$  can be estimated as follows.

$$I = FL \tag{C.1}$$

$$IL^t = FLL^t \tag{C.2}$$

$$F = (IL^t)(LL^t)^{-1} \tag{C.3}$$

$$\tag{C.4}$$

So the idea is to use a monochromator to display light of single wavelength step-wise and to use a spectrophotometer to measure  $L$  for each step size. At the same time photograph the said light at different exposure times. One needs to verify that  $I$  varies linearly with exposure time. If true, find the exposure time when the sensor is not saturated and complete the exercise.  $F$  is now calculated as per equation above. **Check for Linearity for different exposure time.** We used a Flame spectrophotometer with Ocean Optics monochromator to measure Filter response of a Nikon D300 camera.



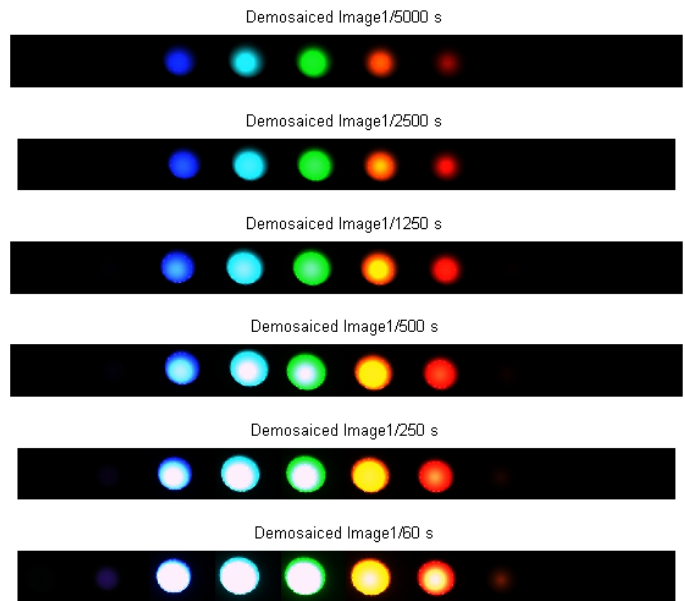


Figure C-1: Check for Linearity in data from RAW file

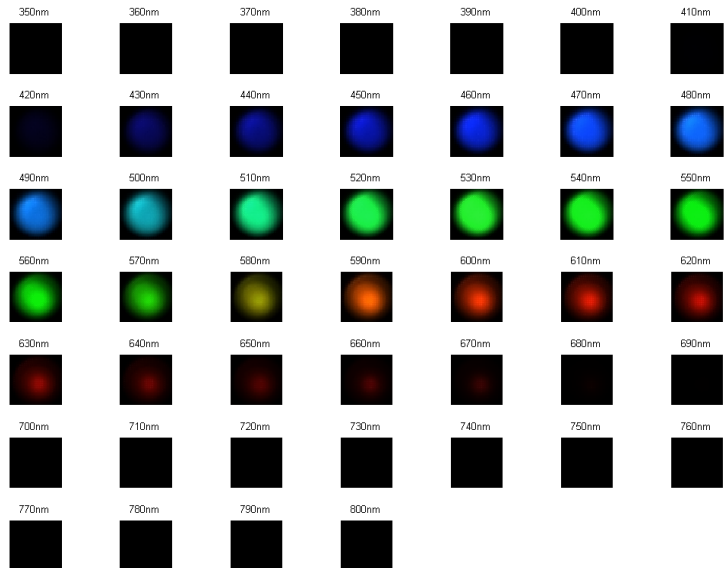


Figure C-2: Photographing different wavelengths with a Nikon D300 camera for exposure time 1/5000s

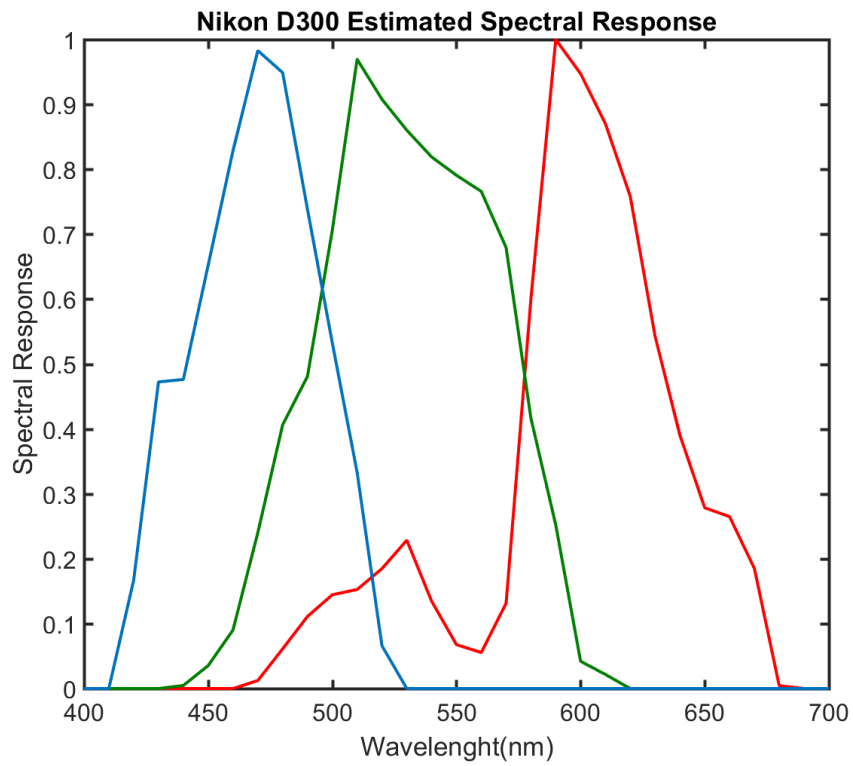


Figure C-3: Estimated Filter response for the Nikon D300



# Appendix D

## Comparison with Compressive Sensing demosaicing for SFAs

### D.1 Testing

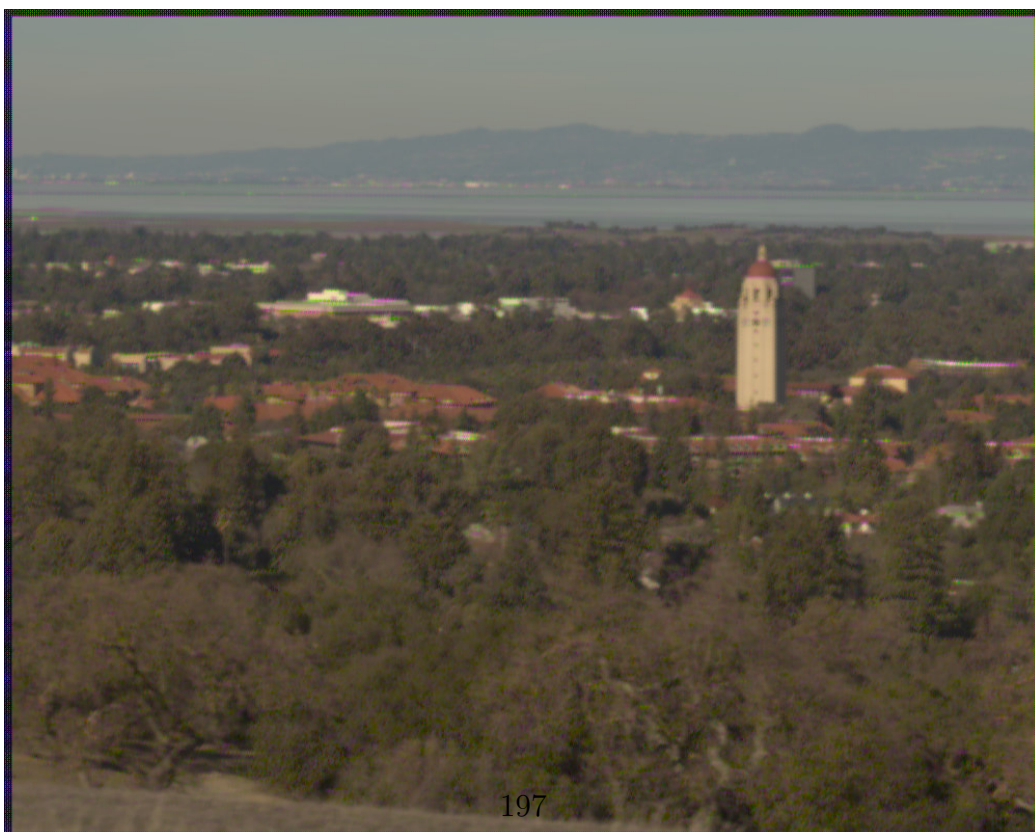
Agarwal et al.<sup>7</sup> were kind enough to share their code for demosaicing based on Compressive Sensing with us. We could more or less reproduce the results (other than GroupSparse) they quote in their paper. One of issues with their methodology is that they provide results for demosaicing for Cave Hyperspectral database. They take the first 3/4 bands (400:10:440nm) images and consider them to be the true color, full resolution images on which they apply their SFA and then demosaic. For Balloon image they report 39.04dB for 3 color diagonal uniform pattern. Using their code we obtain CPSNR of 43.46 in 130s. The difference is probably because I believe for CMSE they do  $MSE = \frac{\sum \sum (\hat{y} - y)^2}{HW}$  where  $H$  and  $W$  are image  $Y$  sizes, however for us it should be  $MSE = \frac{\sum \sum (\hat{y} - y)^2}{3HW}$ . Using LMMSE we get 50.92dB in about 0.2s. We are definitely better, for both PSNR and computation time which is critical in embedded systems. Now you will appreciate that these 3/4 bands are highly correlated. However the correlation with filters like in our camera is different so the performance is even worse. Using Uniform KCS we get an average sPSNR of 39.24dB for SCIEN images, compared to 46.41dB for BST and 53.74dB for LMMSE. Also their method takes on average 502s for each SCIEN image compared to 0.36s for LMMSE. Figure

D-1a, D-1b, D-1c and D-1d shows the comparison of different demosaicing method on one of the SCIEN image. Another issue is that before demosaicing for uniform SFA pattern, they demosaic on a random pattern, get a parameter  $\alpha_{\text{Recon}}$  which is a function of RAW image (of random pattern). And this parameter is then passed as initialization to the basis pursuit solver for uniform SFA. If we omit this step we see a loss of 19dB for the Cave image (balloon) as that reported in their paper. Similarly for the SCIEN simulated images for our sensor we lose 17dB for SanFrancisco image. Earlier figure of average sPSNR 39.24dB was including this step of demosaicing first a random pattern. Now the issue is for real RAW images, like for our sensor, how do we learn the  $\alpha_{\text{Recon}}$  when I don't have access to RAW image coming from random sampling for our filters. Without this step the demosaiced images are not so good, therefore it is not interesting to compare them. It would be incorrect in main chapter to only provide results for SCIEN images and not RAW images. We would normally expect Compressive Sensing approach to have worked better than this, maybe there is some issue in their protocol of initialization parameters. Hemant et al., themselves state in their paper, "In theory such uniform sampling patterns are not conducive to CS recovery". Now in terms of physical feasibility we can have random arrangement within the SFA basis pattern. However for the entire sensor this pattern needs to be repeated.

Figure D-1: Comparison of demosaicing for the JB Sensor simulated on SCIEN database image



(a) Original



(b) Binary-Tree demosaicing



(c) LMMSE demosaicing.



(d) KCS demosaicing

# Bibliography

- [1] A. Roorda and D. R. Williams, “The arrangement of the three cone classes in the living human eye,” *Nature*, vol. 397, no. 6719, pp. 520–522, 1999.
- [2] X. Li, B. Gunturk, and L. Zhang, “Image demosaicing: A systematic survey,” in *Electronic Imaging 2008*. International Society for Optics and Photonics, 2008, pp. 68 221J–68 221J.
- [3] B. Bayer, “Color imaging array,” Jul. 20 1976, uS Patent 3,971,065. [Online]. Available: <https://www.google.com/patents/US3971065>
- [4] “Foveon.” [Online]. Available: <http://www.foveon.com/article.php?a=69>
- [5] 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.
- [6] B. C. de Lavarene, “L’échantillonnage spatio-chromatique dans la rétine humaine et les caméras numériques,” Ph.D. dissertation, Université Joseph-Fourier-Grenoble I, 2007.
- [7] H. K. Aggarwal and A. Majumdar, “Compressive sensing multi-spectral demosaicing from single sensor architecture,” in *Proceedings of 2nd IEEE China Summit and International Conference on Signal and Information Processing (ChinaSIP), 2014*, Jul. 2014.
- [8] “Kodak film spectral sensitivity.” [Online]. Available: <http://www.kodak.com/global/en/professional/support/techPubs/e4046/e4046.pdf>



- [9] Wikipedia, “Cym filter — wikipedia, the free encyclopedia,” 2014, [Online; accessed 24-January-2017]. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=CYGM\\_filter/&oldid=620987344](https://en.wikipedia.org/w/index.php?title=CYGM_filter/&oldid=620987344)
- [10] —, “Rgbe filter — wikipedia, the free encyclopedia,” 2015, [Online; accessed 24-January-2017]. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=RGBE\\_filter/&oldid=686207415](https://en.wikipedia.org/w/index.php?title=RGBE_filter/&oldid=686207415)
- [11] Y. S., “Solid state camera,” 1977, uS Patent US4054906A. [Online]. Available: <https://patents.google.com/patent/US4054906A>
- [12] M. Parmar and S. J. Reeves, “Selection of optimal spectral sensitivity functions for color filter arrays,” *IEEE Transactions on Image Processing*, vol. 19, no. 12, pp. 3190–3203, 2010.
- [13] C. Bai, J. Li, Z. Lin, and J. Yu, “Automatic design of color filter arrays in the frequency domain,” *IEEE Transactions on Image Processing*, vol. 25, no. 4, pp. 1793–1807, 2016.
- [14] C. Zhang, Y. Li, J. Wang, and P. Hao, “Universal demosaicking of color filter arrays,” *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5173–5186, 2016.
- [15] “Kodak image database.” [Online]. Available: <http://r0k.us/graphics/kodak/>
- [16] L. Zhang, X. Wu, A. Buades, and X. Li, “Color demosaicking by local directional interpolation and nonlocal adaptive thresholding,” *Journal of Electronic imaging*, vol. 20, no. 2, pp. 023 016–023 016, 2011.
- [17] G. D. Finlayson, S. D. Hordley, and P. Morovic, “Using the spectracube to build a multispectral image database,” in *Conference on Colour in Graphics, Imaging, and Vision*, vol. 2004, no. 1. Society for Imaging Science and Technology, 2004, pp. 268–274.

- [18] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar, “Generalized assorted pixel camera: Postcapture control of resolution, dynamic range, and spectrum,” *IEEE Transactions on Image Processing*, vol. 19, no. 9, pp. 2241–2253, Sept 2010.
- [19] T. Skauli and J. Farrell, “A collection of hyperspectral images for imaging systems research,” ser. Proc. SPIE, vol. 8660, 2013, pp. 86 600C–86 600C–7. [Online]. Available: <http://dx.doi.org/10.1117/12.2007097>
- [20] Y. Monno, M. Tanaka, and M. Okutomi, “Multispectral demosaicking using adaptive kernel upsampling,” in *18th IEEE International Conference on Image Processing (ICIP)*, Brussels, Sep. 2011, pp. 3157–3160.
- [21] Y. Monno, S. Kikuchi, M. Tanaka, and M. Okutomi, “A practical one-shot multispectral imaging system using a single image sensor,” *IEEE Transactions on Image Processing*, vol. 24, no. 10, pp. 3048–3059, Oct 2015.
- [22] Wikipedia, “Apple quicktake — wikipedia, the free encyclopedia,” 2017, [Online; accessed 22-September-2017]. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=Apple\\_QuickTake&oldid=796996334](https://en.wikipedia.org/w/index.php?title=Apple_QuickTake&oldid=796996334)
- [23] “Phaseone 100mp trichromatic,” <https://www.phaseone.com/fr-FR/Products/Camera-Systems/IQ-Digital-Backs/IQ3-100MP-Trichromatic.aspx>, accessed: 2017-09-22.
- [24] Wikipedia, “The dress — wikipedia, the free encyclopedia,” 2017, [Online; accessed 22-September-2017]. [Online]. Available: [https://en.wikipedia.org/w/index.php?title=The\\_dress&oldid=801322491](https://en.wikipedia.org/w/index.php?title=The_dress&oldid=801322491)
- [25] “Fuji xtrans sensor.” [Online]. Available: <https://www.fujifilm.eu/uk/products/digital-cameras/model/x-pro1/features-4483/aps-c-16m-x-trans-cmos>
- [26] H. S. Malvar, L.-w. He, and R. Cutler, “High-quality linear interpolation for demosaicing of bayer-patterned color images,” in *Acoustics, Speech, and Sig-*

- nal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, vol. 3. IEEE, 2004, pp. iii–485.
- [27] J. Mairal, F. Bach, J. Ponce, G. Sapiro, and A. Zisserman, “Non-local sparse models for image restoration,” in *Computer Vision, 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 2272–2279.
- [28] M. Gharbi, G. Chaurasia, S. Paris, and F. Durand, “Deep joint demosaicking and denoising,” *ACM Transactions on Graphics (TOG)*, vol. 35, no. 6, p. 191, 2016.
- [29] L. Miao, H. Qi, and R. Ramanath, “Generic msfa mosaicking and demosaicking for multispectral cameras,” in *Digital Photography II*, ser. Proc. SPIE, vol. 6069, 2006, pp. 606 909–606 909–10. [Online]. Available: <http://dx.doi.org/10.1117/12.642366>
- [30] D. R. Williams, “Imaging single cells in the living retina,” *Vision research*, vol. 51, no. 13, pp. 1379–1396, 2011.
- [31] “e2v.” [Online]. Available: <https://www.e2v.com/products/imaging/cmos-image-sensors/onyx-ev76c664/>
- [32] “Silios color shades.” [Online]. Available: <https://www.silios.com/cms-cameras-1>
- [33] “Imec hyperspectral imaging.” [Online]. Available: <https://www.imec-int.com/en/hyperspectral-imaging>
- [34] J. F. Hamilton Jr and J. E. Adams Jr, “Adaptive color plan interpolation in single sensor color electronic camera,” May 13 1997, uS Patent 5,629,734.
- [35] K. Hirakawa and P. J. Wolfe, “Spatio-spectral color filter array design for optimal image recovery,” *IEEE Transactions on Image Processing*, vol. 17, no. 10, pp. 1876–1890, 2008.

- [36] D. Alleysson, S. Süsstrunk, and J. Héroult, “Color demosaicing by estimating luminance and opponent chromatic signals in the fourier domain,” in *Color and Imaging Conference*, vol. 2002, no. 1. Society for Imaging Science and Technology, 2002, pp. 331–336.
- [37] D. Alleysson, S. Susstrunk, and J. Héroult, “Linear demosaicing inspired by the human visual system,” *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 439–449, 2005.
- [38] E. Dubois, “Frequency-domain methods for demosaicking of bayer-sampled color images,” *IEEE Signal Processing Letters*, vol. 12, no. 12, pp. 847–850, 2005.
- [39] A. Ribes and F. Schmitt, “Linear inverse problems in imaging,” *IEEE Signal Processing Magazine*, vol. 25, no. 4, 2008.
- [40] H. J. Trussell, “A mmse estimate for demosaicking,” in *Image Processing, 2001. Proceedings. 2001 International Conference on*, vol. 3. IEEE, 2001, pp. 358–361.
- [41] H. J. Trussell and R. E. Hartwig, “Mathematics for demosaicking,” *IEEE Transactions on image processing*, vol. 11, no. 4, pp. 485–492, 2002.
- [42] D. Taubman, “Generalized wiener reconstruction of images from colour sensor data using a scale invariant prior,” in *Image Processing, 2000. Proceedings. 2000 International Conference on*, vol. 3. IEEE, 2000, pp. 801–804.
- [43] M. Parmar and S. J. Reeves, “A perceptually based design methodology for color filter arrays [image reconstruction],” in *Acoustics, Speech, and Signal Processing, 2004. Proceedings. (ICASSP'04). IEEE International Conference on*, vol. 3. IEEE, 2004, pp. iii–473.
- [44] B. C. De Lavarène, D. Alleysson, and J. Héroult, “Practical implementation of lmmse demosaicing using luminance and chrominance spaces,” *Computer Vision and Image Understanding*, vol. 107, no. 1, pp. 3–13, 2007.

- [45] D. Alleysson, S. Süssstrunk, J. Hérault, and B. Chaix de Lavarène, “Linear minimum mean square error demosaicking,” in *Single-sensor imaging: Methods and applications for digital cameras*. CRC Press, 2008, pp. 213–237.
- [46] Y. M. Lu and M. Vetterli, “Optimal color filter array design: Quantitative conditions and an efficient search procedure,” in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2009, pp. 725 009–725 009.
- [47] Y. M. Lu, C. Fredembach, M. Vetterli, and S. Süssstrunk, “Designing color filter arrays for the joint capture of visible and near-infrared images,” in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 3797–3800.
- [48] Z. Sadeghipoor, Y. M. Lu, and S. Süssstrunk, “Correlation-based joint acquisition and demosaicing of visible and near-infrared images,” in *Image Processing (ICIP), 2011 18th IEEE International Conference on*. IEEE, 2011, pp. 3165–3168.
- [49] ———, “Optimum spectral sensitivity functions for single sensor color imaging,” in *IS&T/SPIE Electronic Imaging*. International Society for Optics and Photonics, 2012, pp. 829 904–829 904.
- [50] D. Alleysson and B. C. de Lavarène, “Frequency selection demosaicking: A review and a look ahead,” in *Electronic Imaging 2008*. International Society for Optics and Photonics, 2008, pp. 68 221M–68 221M.
- [51] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [52] D. Alleysson, B. De Lavarene, and J. Herault, “Digital image sensor, image capture and reconstruction method and system for implementing same,” Oct. 22 2013, uS Patent 8,564,699. [Online]. Available: <https://www.google.fr/patents/US8564699>

- [53] R. Lukac and K. N. Plataniotis, “Color filter arrays: Design and performance analysis,” *IEEE Transactions on Consumer Electronics*, vol. 51, no. 4, pp. 1260–1267, 2005.
- [54] L. Condat, “A generic variational approach for demosaicking from an arbitrary color filter array,” in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 1625–1628.
- [55] “Demosaicing codes online.” [Online]. Available: [http://web.engr.illinois.edu/~khashab2/files/2013\\_2014\\_demosaicing/demosaicing.html](http://web.engr.illinois.edu/~khashab2/files/2013_2014_demosaicing/demosaicing.html)
- [56] D. Paliy, V. Katkovnik, R. Bilcu, S. Alenius, and K. Egiazarian, “Spatially adaptive color filter array interpolation for noiseless and noisy data,” *International Journal of Imaging Systems and Technology*, vol. 17, no. 3, pp. 105–122, 2007.
- [57] N.-X. Lian, L. Chang, Y.-P. Tan, and V. Zagorodnov, “Adaptive filtering for color filter array demosaicking,” *IEEE Transactions on Image Processing*, vol. 16, no. 10, pp. 2515–2525, 2007.
- [58] K. Hirakawa and T. W. Parks, “Adaptive homogeneity-directed demosaicing algorithm,” *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 360–369, 2005.
- [59] X. Li, “Demosaicing by successive approximation,” *IEEE Transactions on Image Processing*, vol. 14, no. 3, pp. 370–379, 2005.
- [60] D. Menon, S. Andriani, and G. Calvagno, “Demosaicing with directional filtering and a posteriori decision,” *IEEE Transactions on Image Processing*, vol. 16, no. 1, pp. 132–141, 2007.
- [61] L. Zhang and X. Wu, “Color demosaicking via directional linear minimum mean square-error estimation,” *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2167–2178, 2005.

- [62] B. K. Gunturk, Y. Altunbasak, and R. M. Mersereau, “Color plane interpolation using alternating projections,” *IEEE transactions on image processing*, vol. 11, no. 9, pp. 997–1013, 2002.
- [63] C.-Y. Su, “Highly effective iterative demosaicing using weighted-edge and color-difference interpolations,” *IEEE Transactions on Consumer Electronics*, vol. 52, no. 2, pp. 639–645, 2006.
- [64] W. Lu and Y.-P. Tan, “Color filter array demosaicking: new method and performance measures,” *IEEE Transactions on Image Processing*, vol. 12, no. 10, pp. 1194–1210, 2003.
- [65] F. Yasuma, T. Mitsunaga, D. Iso, and S. K. Nayar, “Generalized assorted pixel camera: Post-capture control of resolution, dynamic range and spectrum,” Department of Computer Science, Columbia University CUCS-061-08, Tech. Rep., Nov 2008, <http://www1.cs.columbia.edu/CAVE/databases/multispectral/>.
- [66] L. Condat, “A new color filter array with optimal properties for noiseless and noisy color image acquisition,” *IEEE Transactions on image processing*, vol. 20, no. 8, pp. 2200–2210, 2011.
- [67] —, “A new color filter array with optimal sensing properties,” in *Image Processing (ICIP), 2009 16th IEEE International Conference on*. IEEE, 2009, pp. 457–460.
- [68] P. Hao, Y. Li, Z. Lin, and E. Dubois, “A geometric method for optimal design of color filter arrays,” *IEEE Transactions on Image Processing*, vol. 20, no. 3, pp. 709–722, 2011.
- [69] J. Couillaud, A. Horé, and D. Ziou, “Nature-inspired color-filter array for enhancing the quality of images,” *JOSA A*, vol. 29, no. 8, pp. 1580–1587, 2012.
- [70] P. Amba, J. Dias, and D. Alleysson, “Random color filter arrays are better than regular ones,” *Journal of Imaging Science and Technology*, vol. 60, no. 5, pp. 50 406–1, 2016.

- [71] S.-P. Han, “A globally convergent method for nonlinear programming,” *Journal of optimization theory and applications*, vol. 22, no. 3, pp. 297–309, 1977.
- [72] M. J. Powell, “A fast algorithm for nonlinearly constrained optimization calculations,” in *Numerical analysis*. Springer, 1978, pp. 144–157.
- [73] ———, “The convergence of variable metric methods for non-linearly constrained optimization calculations,” *Nonlinear programming*, vol. 3, 1978.
- [74] T. Yamagami, T. Sasaki, and A. Suga, “Image signal processing apparatus having a color filter with offset luminance filter elements,” Jun. 21 1994, uS Patent 5,323,233. [Online]. Available: <https://www.google.com/patents/US5323233>
- [75] P.-J. Lapray, X. Wang, J.-B. Thomas, and P. Gouton, “Multispectral filter arrays: Recent advances and practical implementation,” *Sensors*, vol. 14, no. 11, p. 21626, 2014. [Online]. Available: <http://www.mdpi.com/1424-8220/14/11/21626>
- [76] X. Wang, J.-B. Thomas, J. Y. Hardeberg, and P. Gouton, “Multispectral imaging: narrow or wide band filters?” *Journal of the International Colour Association*, vol. 12, pp. 44–51, 2014.
- [77] X. Wang, P. J. Green, J.-B. Thomas, J. Y. Hardeberg, and P. Gouton, *Evaluation of the Colorimetric Performance of Single-Sensor Image Acquisition Systems Employing Colour and Multispectral Filter Array*. Cham: Springer International Publishing, 2015, pp. 181–191. [Online]. Available: [http://dx.doi.org/10.1007/978-3-319-15979-9\\_18](http://dx.doi.org/10.1007/978-3-319-15979-9_18)
- [78] Y. Benezeth, D. Sidibé, and J.-B. Thomas, “Background subtraction with multispectral video sequences,” in *IEEE ICRA workshops*, Hong-Kong, 2014, p. 6 p. [Online]. Available: <http://hal-univ-bourgogne.archives-ouvertes.fr/hal-00986168>



- [79] R. Ramanath, W. E. Snyder, G. L. Bilbro, and W. A. Sander, “Robust multispectral imaging sensors for autonomous robots,” <http://www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA396058>, North Carolina State University, Tech. Rep., 2001, retrieved 16 June 2014.
- [80] R. Ramanath, W. E. Snyder, and H. Qi, “Mosaic multispectral focal plane array cameras,” in *Infrared Technology and Applications XXX*, ser. Proc. SPIE, vol. 5406, 2004, pp. 701–712. [Online]. Available: <http://dx.doi.org/10.1117/12.543418>
- [81] J. Brauers and T. Aach, “A color filter array based multispectral camera,” in *12th Workshop Farbbildverarbeitung*, G. C. Group, Ed., Ilmenau, October 2006.
- [82] S. Mihoubi, O. Losson, B. Mathon, and L. Macaire, “Multispectral demosaicing using intensity in edge-sensing and iterative difference-based methods,” in *IEEE SITIS 2016*, ser. Proceedings of the 12th International IEEE Conference on SIGNAL IMAGE TECHNOLOGY & INTERNET BASED SYSTEMS, Napoli, Italy, Dec. 2016, pp. 805–810. [Online]. Available: <https://hal.archives-ouvertes.fr/hal-01409705>
- [83] L. Miao, H. Qi, R. Ramanath, and W. E. Snyder, “Binary tree-based generic demosaicking algorithm for multispectral filter arrays,” *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3550–3558, Nov 2006.
- [84] L. Miao, H. Qi, and R. Ramanath, “A generic binary tree-based progressive demosaicking method for multispectral filter array,” in *International Conference on Image Processing (ICIP), 2006*. IEEE, Oct 2006, pp. 3221–3224.
- [85] S. Mihoubi, O. Losson, B. Mathon, and L. Macaire, “Multispectral demosaicing using intensity-based spectral correlation,” in *2015 International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Nov 2015, pp. 461–466.
- [86] SILIOS-TECHNOLOGIES, “Micro-optics supplier,” <http://www.silios.com/>.

- [87] PIXELTEQ, “Micro-patterned optical filters,” <https://pixelteq.com/>.
- [88] J.-B. Thomas, P.-J. Lapray, P. Gouton, and C. Clerc, “Spectral characterization of a prototype SFA camera for joint visible and NIR acquisition,” *Sensors*, vol. 16, no. 7, p. 993, 2016. [Online]. Available: <http://www.mdpi.com/1424-8220/16/7/993>
- [89] P.-J. Lapray, J.-B. Thomas, and P. Gouton, “A database of spectral filter array images that combine visible and nir,” *6th Computational Color Imaging Workshop, CCIW’17*, 2017.
- [90] F. Gasparini, F. Marini, R. Schettini, and M. Guarnera, “A no-reference metric for demosaicing artifacts that fits psycho-visual experiments,” *EURASIP Journal on Advances in Signal Processing*, vol. 2012, no. 1, p. 123, 2012. [Online]. Available: <http://dx.doi.org/10.1186/1687-6180-2012-123>
- [91] P.-J. Lapray, J.-B. Thomas, P. Gouton, and Y. Ruichek, “Energy balance in spectral filter array camera design,” *Journal of the European Optical Society-Rapid Publications*, vol. 13, no. 1, p. 1, 2017. [Online]. Available: <http://dx.doi.org/10.1186/s41476-016-0031-7>
- [92] P.-J. Lapray, J.-B. Thomas, and P. Gouton, “A multispectral acquisition system using msfas,” *Color and Imaging Conference*, vol. 2014, no. 2014, pp. 97–102, 2014-11-03T00:00:00. [Online]. Available: <http://www.ingentaconnect.com/content/ist/cic/2014/00002014/00002014/art00016>
- [93] L. Miao, H. Qi, and W. E. Snyder, “A generic method for generating multispectral filter arrays,” in *International Conference on Image Processing (ICIP)*, vol. 5. IEEE, Oct. 2004, pp. 3343–3346.
- [94] L. Miao and H. Qi, “The design and evaluation of a generic method for generating mosaicked multispectral filter arrays,” *IEEE Transactions on Image Processing*, vol. 15, no. 9, pp. 2780–2791, Sept 2006.

- [95] P. Amba, J. Dias, and D. Alleysson, “Random color filter arrays are better than regular ones,” *Journal of Imaging Science and Technology*, vol. 60, no. 5, pp. 50406–1, 2016.
- [96] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE transactions on image processing*, vol. 13, no. 4, pp. 600–612, 2004.
- [97] A. Mittal, A. K. Moorthy, and A. C. Bovik, “Referenceless image spatial quality evaluation engine,” November 2011.
- [98] —, “No-reference image quality assessment in the spatial domain,” *IEEE Transactions on Image Processing*, vol. 21, no. 12, pp. 4695–4708, Dec 2012.
- [99] M. A. Saad, A. C. Bovik, and C. Charrier, “Dct statistics model-based blind image quality assessment,” in *2011 18th IEEE International Conference on Image Processing*, Sept 2011, pp. 3093–3096.
- [100] —, “Blind image quality assessment: A natural scene statistics approach in the dct domain,” *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3339–3352, Aug 2012.
- [101] J. B. Thomas, “Illuminant estimation from uncalibrated multispectral images,” in *2015 Colour and Visual Computing Symposium (CVCS)*, Aug 2015, pp. 1–6.
- [102] B. Arad and O. Ben-Shahar, “Sparse recovery of hyperspectral signal from natural rgb images,” in *European Conference on Computer Vision*. Springer, 2016, pp. 19–34.
- [103] “Hamamatsu sensor response,” [https://www.hamamatsu.com/resources/pdf/ssd/image\\_sensor\\_kmpd0002e.pdf](https://www.hamamatsu.com/resources/pdf/ssd/image_sensor_kmpd0002e.pdf), accessed: 2017-08-22.
- [104] I. Kanellopoulos and G. Wilkinson, “Strategies and best practice for neural network image classification,” *International Journal of Remote Sensing*, vol. 18, no. 4, pp. 711–725, 1997.

- [105] Y.-Q. Wang, “A multilayer neural network for image demosaicking,” in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 1852–1856.
- [106] T. Kijima, H. Nakamura, J. Compton, J. Hamilton, and T. DeWeese, “Image sensor with improved light sensitivity,” Nov. 22 2007, uS Patent App. 11/419,574. [Online]. Available: <https://www.google.com/patents/US20070268533>
- [107] I. Hirota, “Solid-state imaging device, method for processing signal of solid-state imaging device, and imaging apparatus,” May 7 2013, uS Patent 8,436,925. [Online]. Available: <https://encrypted.google.com/patents/US8436925>
- [108] Y. Monno, D. Kiku, S. Kikuchi, M. Tanaka, and M. Okutomi, “Multispectral demosaicking with novel guide image generation and residual interpolation,” in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 645–649.
- [109] —, “Multispectral demosaicking with novel guide image generation and residual interpolation,” in *2014 IEEE International Conference on Image Processing (ICIP)*, Oct 2014, pp. 645–649.



# Curriculum Vitae

## Prakhar Amba

Date of Birth: 21/02/1985

Nationality: Indian

Situation of Family: Married

Sex: Male

Permis B

Phd. Student / Engineer

6, Avenue Leon Blum

Grenoble - 38100

France

Ph +33 7 82 92 23 56

prakhar.amba@univ-grenoble-alpes.fr

---

### Skills

|                       |  |
|-----------------------|--|
| Domain                | Hyperspectral Imaging, Colorimetry, Image Processing, Machine Learning, Software development, F&B, Footwear CAD/CAM & production, Project Management |
| Languages             | English (Bilingual), French (Bilingual), Hindi (maternal)  |
| Programming Languages | Matlab, Verilog/VHDL, Spice, C, C++, VB, SQL, Unix Shell programming, Assembly   |
| Utilities             | Latex, MS Office, Xilinx ISE, ModelSim   |

---

### Education

- Université Grenoble Alpes, FR Mar'15 - current  
Doctorant in "Study of cone signal demosaicing in color vision"
- Institut Supérieur d'Electronique de Paris,FR Oct'07 - Dec'08  
MSc (Bac+5) in Electronics & Telecommunications
- Dhirubhai Ambani Institute of ICT, India Gandhinagar Jul'03 - May'07  
B.Tech in Information & Communication Technology

## Work Experience

---

- **CNRS - LPNC** Grenoble, France Mar'15 - current  
Currently working as Phd. Student in LPNC lab holding a position of Engineer (IE) with CNRS. My work is related to developing image processing pipeline for digital color cameras, spectral cameras, NIR cameras for night vision, etc. I am involved with spatio-spectral optimization of filter arrays, designing RAW converters, etc.
- **Strategies** ([www.romans-cad.com](http://www.romans-cad.com)) Rungis, France Mar'14 - Feb'15  
Indian representatives of Strategies which makes CAD software for footwear and bag industry. Organised the booth for trade shows in Delhi, Chennai and Agra. Represented the firm at the International Footwear Conference'14 organised by Council of Leather Exports.
- **Jodha the restaurant** ([www.jodha.co.in](http://www.jodha.co.in)) Agra, India Feb'12 - Jan'15  
Proprietor. Managed a 96 cover fine dine restaurant. I employed team of 25 people. Established ourselves in a highly competitive market.
- **Amba Cameras** ([www.ambacameras.com](http://www.ambacameras.com)) Agra, India May'11 - May'13  
Indian representatives of Seitz Phototechnik (Switzerland) in India. Involved in marketing and sales of high-end panoramic cameras.
- **CEA-LETI**([www.cea.fr](http://www.cea.fr)) Grenoble, France Feb'09 - Apr'11  
Engineer doing research on bioinspired architectures for image sensors. My work is related to adaptation of features of human visual system to massively parallel architectures that can be implemented on a digital camera sensor. It leads to improvement in color fidelity, noise reduction, dynamic range, resolution, etc. of captured images.
- **Seitz Phototechnik** ([www.roundshot.ch](http://www.roundshot.ch)) Lustdorf, Switzerland Feb'09 - Jan'11  
Independent consultant to a Swiss based digital camera manufacturer. Services provided related to their digital camera image processing workflow. Algorithms designed for RAW conversion of images.
- **University Joseph Fourier**, Grenoble, France Jan'10 - July'10  
Teacher. Taught a course on "Data analysis using Principal Component Analysis" to Master students.

## Internship

---

- **ATMEL** -Rousset, France July'08 - Dec'08  
Study of process variations in CMOS memory design (SRAM, ROM) for stability and power from 150nm-65nm. Leakage reduction techniques using voltage biasing evaluated.
- **Reliance Infocomm** New Delhi, India May'05 - June'05  
To understand the communication network of the mobile operator, implemented voice based alarm system to notify hardware faults in the switches. Also created a database of per calls measurement data and built front-end user interface.

## Patents / Publications / Talks

---

- **Patent.** David Alleysson and Prakhar Amba. "Method of improving color imagery from single sensor cameras and its implementation method." (2017) (Pending).
- **Journal.** Prakhar Amba, Jérôme Dias, and David Alleysson. "Random Color Filter Arrays are Better than Regular Ones." *Journal of Imaging Science and Technology* 60.5 (2016): 50406-1
  1. **ITEX Award for best Journal Paper of Journal of Imaging Science and Technology 2016.**
  2. **Best Student Paper Award at CIC24 (Color and Imaging Conference), 2016 San Diego, USA**
- **Journal.** Amba, Prakhar, Jean Baptiste Thomas, and David Alleysson. "N-LMMSE Demosaicing for Spectral Filter Arrays." *Journal of Imaging Science and Technology* 61.4 (2017): 40407-1
- **Journal.** Amba, Prakhar, David Alleysson, and Martial Mermillod. "Demosaicing using Dual Layer Feedforward Neural Network." *IEEE Transactions on Neural Networks and Learning Systems*, Vol. xx, No. x, Month 201x. **Under Review**
- **Talk.** Color Filter Arrays imitating the random nature of color arrangement in human cone mosaic. 6th European Workshop on Visual Information Processing (EUVIP). Marseille, France. 25 - 27 October 2016
- **Talk.** Random Color Filter Arrays are Better than Regular Ones. Norwegian University of Science and Technology, Gjøvik, Norway. 6th Feb 2017
- **Talk.** Learning methods for demosaicing (color and spectral filter arrays), Journées NeuroSTIC 2017, Brest, France. 6th Feb 2017
  - **Talk.** Learning methods for demosaicing (color and spectral filter mosaics), Grenoble Fest @ LPP, Université Paris Descartes. Paris, France. 8th Dec 2017
  - **Poster.** Random CFAs are better than periodic ones. Lyon, France. GDR BioComp 2016

## Interests

---

- Photography. My work has been exhibited at CEA bibliotheque, published with BBC, commercialized
  - Reading English literature, favorite authors are Grass, Rushdie, Dickens, and Tagore.
  - Travelling.