Università degli Studi di Padova
Dipartimento di Ingegneria dell'Informazione
Corso di Laurea Magistrale in Ingegneria Informatica

TESI DI LAUREA

Color Image Reconstruction via Sparse Signal Representation

LAUREANDO:

Mattia Rossi

RELATORE:

Ch.mo Prof. Giancarlo Calvagno

A.A. 2012-2013

Abstract

In a typical consumer digital camera, the image sensor at each pixel senses only one out of the three color components (usually red, green, and blue) the representation of a digital color image calls for. This raises the problem of reconstructing the missing color components at each pixel from the data acquired, a problem widely known in the litterature as demosaicing. Recently, demosaicing algorithms based on Sparse Recovery Theory have been proposed. These algorithms, we will refer to as sparse-based demosaicing algorithms, aim at estimate the original image by the one compatible with the acquired data and admiting the sparsest representation under a given sparsifying dictionary. In this thesis a recent sparse-based demosaicing algorithm proposed in the litterature is considered. As opposed to other sparse-based demosaicing algorithms, the one studied here employes a dictionary which explicitly takes into account the inter-pixel (spatial) and inter-channel (color) image correlation, usually leading to reconstructed images with notably less visual artifact than leading demosaicing algorithms, even when the PSNR measure would suggest the converse. As the authors did not made any software implementation of their algorithm available, we decided to develop our implementation of the algorithm. This thesis paralles the work done toward this direction, from the study of the demosaicing problem and the fundamental results of Sparse Recovery Theory, to the simulation results we obtained.

Contents

Chapter 1

Introduction

Since their introduction in the mid-1990s, consumer digital cameras have undergone a rapid growth, mainly due to their capability to store a large amount of images at a low cost, allowing the user to decide if discard, archive, or post-process the image after its acquisition. Today a typical consumer camera has over 12 million pixels, but the sensor at each pixel senses only one out of the three color components (usually red, green, and blue) the representation of a digital color image calls for. As the camera senses only one-third of the required information, the remaining two-thirds need to be reconstructed from the available data via a proper algorithm. This particular reconstruction problem is refered to as *demosaicing*. Clearly the quality of the final image heavily depends on the demosaicing procedure, and this is also one of the reasons why the pixel counter of a digital camera usually cannot be referred to as its spatial resolution.

Due to the success of digital cameras and their increasing spread, also embedded in other devices such as smartphones, the demosaicing problem has received large attention by the *Digital Signal Processing (DSP)* community, and several demosaicing algorithms have been proposed. Recently, some authors have considered to apply the interesting approach of *Sparse Recovery Theory* to the demosaicing problem, thus introducing a new class of demosaicing algorithms we will refer to as *sparse-based algorithms*. Sparse Recovery Theory deals with the recovery of a vector of interest from a number of proper linear measurements smaller than the original vector size, provided the vector admits a sparse or compressible representation. It happens that the relationship between the image we would ideally acquire,

and the data that is actually acquired by the camera, can be stated as a linear system of equations. Moreover natural color images are known to admit highly sparse or compressible representations due to their high correlation. Therefore Sparse Recovery Theory appears as an interesting strategy to attack the demosaicing problem. However, as it is often the case in engineering, when trying to apply an interesting mathematical theory on a practical problem, this usually happens to not fit all the assumptions that the considered mathematical theory would call for. In order to guarantee exact recovery of the original vector, or a well approximation of it, Sparse Recovery clearly requires the linear measurements on the vector to satisfy specific assumptions. However, the linear system involved in the demosaicing problem hardly happens to satisfy all these assumptions. Nevertheless, empirical results show that sparse-based demosaicing algorithms perform well in practice, and moreover they usually outperform the non-sparse-based leading algorithms.

This thesis focuses on the sparse-based demosaicing framework by Moghadam et al. presented in [11]. Our interest for this framework obviously comes from the simulation results reported by Moghadam et al. Their tests on the KODAK dataset show that the images demosaiced via their framework usually exibit notably less visual artifacts when compared to images demosaiced by other leading algorithms, and this happens to be true even when PSNRs would suggest the converse. This property makes the framework by Moghadam et al. very appealing, as in real cases the full-resolution image is no longer available, therefore no PSNR values can be computed, and the visual quality of the image is all that matter.

Unfortunately Moghadam et al. did not made their implementation of the framework available, neither the source code, nor an executable version of it. This led us to the decision to try developing our implementation of the framework. This thesis parallels the work done toward this direction. Chapter 2 formally introduces the demosaicing problem, analyzes the relation among the full-resolution image we would ideally acquire and the actually acquired one, finally presents how Sparse Recovery theory meets the demosaicing problem, automatically leading to the formulation of the general structure of a sparse-based demosaicing algorithm. Chapter 3 provides some fundamental results of Sparse Recovery theory, those we expect to be useful in understanding the approach adopted by sparse-based demosaicing algorithms. Chapter 4 presents the demosaicing framework by Moghadam et al. as described

in [11]. Chapter 5 has a double purpose. The first one is to present the simulation results provided by Moghadam et al. The second one is to present the results we achieved with our implementation of the framework, and provide some observations on the work of Moghadam et al. Finally, Chapter 6 concludes this thesis.

The remainder of this chapter introduces some notation. For a number $q \in \mathbb{N}$, we define $[q] \triangleq \{1, 2, \ldots, q\}$. The cardinality of a finite set S will be denoted as $card(S)$. Matrices will be represented by bold font (e.g., $\boldsymbol{A}$), while vectors or scalars will be represented by normal font (e.g., x). Let us consider a matrix $\boldsymbol{A} \in \mathbb{R}^{s_1 \times s_2}$ and a vector $x \in \mathbb{R}^{s_1}$. The entry of $\boldsymbol{A}$ at position $(i, j) \in [s_1] \times [s_2]$ will be denoted by $\boldsymbol{A}_{i,j}$. Similarly, the i-th entry of vector x will be denoted by x_i. We define the support of vector x as $supp(x) \triangleq \{i \in [s_1] : x_i > 0\}$. We will use MATLAB notation for identifying submatrices of matrix $\boldsymbol{A}$, or subvectors of vector x. For instance, for $v_1 \subseteq [s_1]$ and $v_2 \subseteq [s_2]$, we will denote by $\boldsymbol{A}_{v_1,:}$ the submatrix of $\boldsymbol{A}$ restricted to the rows of v_1 and all the columns, and by $\boldsymbol{A}_{:,v_2}$ the submatrix of $\boldsymbol{A}$ restricted to the columns of v_2 and all the rows. Similarly we will denote by x_{v_1} the subvector of x restriced to the indices of v_1. According to the MATLAB notation, we will also denote the square $s_1 \times s_1$ diagonal matrix containing vector x on its main diagonal and zeros outside, by $diag(x)$. The transpose of matrix $\boldsymbol{A}$ will be denoted by $\boldsymbol{A}^T$. To simplify equations, when matrix $\boldsymbol{A}$ represents a bidimensional image, such as a single channel of a color image, we will usually deal with the vectorized form of $\boldsymbol{A}$. The vectorized form of $\boldsymbol{A}$ is the $s_1 s_2 \times 1$ vector A obtained by stacking the columns of $\boldsymbol{A}$ on top of one another. Formally is $A \triangleq \left[(\boldsymbol{A}_{:,\{1\}})^T \ (\boldsymbol{A}_{:,\{2\}})^T \ \ldots \ (\boldsymbol{A}_{:,\{s_2\}})^T\right]^T$. Finally we will denote the Hadamard (point wise) product by $\odot$, and the Kronecker tensor product by $\otimes$.

Chapter 2

Demosaicing: a Reconstruction Problem

In a digital camera designed for gray scale images only, the acquisiton could be performed through a bidimensional array of sensors, *CMOS (Complementary Metal Oxide Silicon)* or *CCD (Charged-Coupled Device)*, each one measuring the intesity of the visible light signal at its location, usually refered to as *pixel*. In a color digital camera instead, at each pixel location three values should be measured: the intensities of the light signal around the red color frequencies, the green frequencies, and the blue ones. This seems to suggest that three sensors per pixel may be required, each sensing a particular set of wavelengths. This however introduces a nontrivial problem about the positioning of the sensors. One approach is to use a beam-splitter along the optical path to project the image onto three separate bidimensional arrays of sensors, each one equipped with a color filter ahead. The filters allow each array to sense only the frequencies related to one of the three colors, and the result are three full-resolution color images. This is a costly approach as it requires three sensors per pixel, and moreover these have to be aligned precisely in order to avoid phase-delay effects, not a simple challenge to mechanical design [1]. Another approach, still costly because of the number of sensors employed, is to stack the three sensors on top of one another, as in Foveon cameras [2], but this arrangment suffers of signal attenuation problems as the ligth has to penetrate three levels of silicon. Therefore, most of the color digital cameras adopt a "one sensor per pixel" approach. They use one bidimensional array of sensors, as in the gray

scale camera, but with a *Color Filter Array (CFA)* ahead that allows each sensor to capture only one out of the red, green, and blue colors, or possibly a linear combination of them. Since the acquired image contains only partial information about the red, green, and blue channels, a color reconstruction step is required in order to get a full-color image. As the image acquired looks like a mosaic, the color reconstruction process is called *demosaicing*.

Let us formalize the acquisition process for a color digital camera employing the "one sensor per pixel" approach. Suppose the camera acquires $s_1 \times s_2$ images, and let $\boldsymbol{R}, \boldsymbol{G}, \boldsymbol{B} \in \mathbb{R}^{s_1 \times s_2}$ be respectively the red, green, and blue color planes of our target image, with $\boldsymbol{R}_{i,j}$, $\boldsymbol{G}_{i,j}$, and $\boldsymbol{B}_{i,j}$ the values of the red, green, and blue channels at pixel (i, j). Using a generic CFA, the value sensed by the camera at pixel (i, j) can be represented as

$$\boldsymbol{y}_{i,j} = \boldsymbol{\alpha}_{i,j}\boldsymbol{R}_{i,j} + \boldsymbol{\beta}_{i,j}\boldsymbol{G}_{i,j} + \boldsymbol{\gamma}_{i,j}\boldsymbol{B}_{i,j} \qquad \forall\, (i, j) \in [s_1] \times [s_2] \tag{2.1}$$

where $\boldsymbol{\alpha}_{i,j}$, $\boldsymbol{\beta}_{i,j}$, and $\boldsymbol{\gamma}_{i,j}$ are some positive weigths describing the CFA sensitivity to the red, green, and blue colors at pixel (i, j), with the constraint $\boldsymbol{\alpha}_{i,j} + \boldsymbol{\beta}_{i,j} + \boldsymbol{\gamma}_{i,j} = 1$.

Extending formulation (2.1) to the whole image sensed by the camera, the *CFA image* hereafter, yields

$$\boldsymbol{y} = \boldsymbol{\alpha} \odot \boldsymbol{R} + \boldsymbol{\beta} \odot \boldsymbol{G} + \boldsymbol{\gamma} \odot \boldsymbol{B}. \tag{2.2}$$

The demosaicing problem deals with the reconstruction of matrices $\boldsymbol{R}$, $\boldsymbol{G}$, $\boldsymbol{B}$, from $\boldsymbol{y}$.

CFAs whose weights vectors $\left[\boldsymbol{\alpha}_{i,j}\, \boldsymbol{\beta}_{i,j}\, \boldsymbol{\gamma}_{i,j}\right]$ belong to the canonical basis of $\mathbb{R}^3$ are usually refered to as *pure-color* filters, since at each pixel they allow the sensor to capture only one out of the red, green, and blue colors. Those filters perform a straight sampling of the three color planes. Non-pure-color CFAs are usually refered to as *panchromatic*.

Although several color filter arrays have been proposed since the introduction of the "one sensor per pixel" digital color camera, the most common one is the *Bayer CFA*, after the name of its inventor, Bryce Bayer, who designed it for the Eastman Kodak Company in 1976 [3]. The Bayer filter belongs to the pure-color category: it samples the green color plane using a quincux grid, while the red and blue ones

using a rectangular grid, as shown in Fig. 2.1.

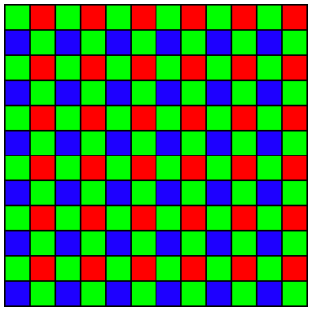


Figure 2.1: The Bayer CFA.

Because of its large usage most of the demosaicing algorithms in the litterature have been developed for the Bayer CFA. These algorithms can be partioned mainly into five categories, based on the tools they use and the strategies they adopt: *heuristic methods*, *edge-directed methods*, *frequency-based methods*, *wavelet-based methods*, and *reconstruction methods*. This last category, reconstruction methods, gathers together all demosaicing algorithms assuming some prior knowledge on the target image and adopting a MAP or MMSE estimator as their reconstruction strategy. *Sparse-based demosaicing algorithms* belong to this category, as they assume that the target image admits a sparse representation with respect to some basis or frame, and set up the reconstruction on this prior. Their belonging to the reconstruction methods category becomes even more evident if we consider that, under some reasonable assumptions on the stochastic nature of the target image sparsity, it can be shown that the related MAP estimator happens to be well approximated by the optmization problem at the core of sparse-based demosaicing, later refered to as problem (P_0) (see Chapter 9 of [4] for a formal proof).

Since Bayer filter is the most common CFA and it will be empoyed in this thesis too, it is worth considering some of its features. These will be the subject of the next section.

2.1 The Bayer filter and frequency-based demosaicing

For the Bayer filter, matrices $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$, can be defined as follows, with $(i, j) \in [s_1] \times [s_2]$:

$$\begin{aligned} \boldsymbol{\alpha}_{i,j} &= \frac{1}{4}\left(1-(-1)^i\right)\left(1+(-1)^j\right) \\ \boldsymbol{\beta}_{i,j} &= \frac{1}{2}\left(1+(-1)^{i+j}\right) \\ \boldsymbol{\gamma}_{i,j} &= \frac{1}{4}\left(1+(-1)^i\right)\left(1-(-1)^j\right). \end{aligned} \tag{2.3}$$

In 2002, Alleyson et al. [5] showed that the (Bayer) CFA image can be represented as a linear combination of an achromatic component at baseband, refered to as *luminance*, and two *chrominance* components modulated at high frequency. This happens to be strictly connected to the strategy adopted in *PAL* and *SECAM* analog television standards. The luminance component contains the spatial resolution of the image, while the chrominace components take account of its colors. In [6] Dubois gives a simplified derivation of the frequency-domain representation of the CFA image developed by Alleyson et al. Here we present Dubois's derivation using the original notation.

Let $m_R(n_1, n_2)$, $m_G(n_1, n_2)$, and $m_B(n_1, n_2)$, with $(n_1, n_2) \in \mathbb{Z}\times\mathbb{Z}$, be respectively the extensions of matrices $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$ to the whole plane:

$$\begin{aligned} m_R(n_1, n_2) &= \frac{1}{4}\left(1-(-1)^{n_1}\right)\left(1+(-1)^{n_2}\right) \\ m_G(n_1, n_2) &= \frac{1}{2}\left(1+(-1)^{n_1+n_2}\right) \\ m_B(n_1, n_2) &= \frac{1}{4}\left(1+(-1)^{n_1}\right)\left(1-(-1)^{n_2}\right). \end{aligned} \tag{2.4}$$

Similarly, let $f_R(n_1, n_2)$, $f_G(n_1, n_2)$, and $f_B(n_1, n_2)$, with $(n_1, n_2) \in \mathbb{Z} \times \mathbb{Z}$, be the extensions, padded with zeros outside $[s_1] \times [s_2]$, of matrices $\boldsymbol{R}$, $\boldsymbol{G}$, and $\boldsymbol{B}$.

The CFA image can then be expressed as

$$\begin{aligned} f_{CFA}(n_1,n_2) &= \sum_{i\in\{R,G,B\}} f_i(n_1,n_2)\, m_i(n_1,n_2) \\ &= \frac{1}{4} f_R(n_1,n_2)\left(1-(-1)^{n_1}\right)\left(1+(-1)^{n_2}\right) \\ &\quad + \frac{1}{2} f_G(n_1,n_2)\left(1+(-1)^{n_1+n_2}\right) \\ &\quad + \frac{1}{4} f_R(n_1,n_2)\left(1+(-1)^{n_1}\right)\left(1-(-1)^{n_2}\right). \end{aligned} \tag{2.5}$$

This can be rearranged as

$$\begin{aligned} f_{CFA}(n_1,n_2) &= \left(\frac{1}{4} f_R(n_1,n_2) + \frac{1}{2} f_G(n_1,n_2) + \frac{1}{4} f_B(n_1,n_2)\right) \\ &\quad + \left(-\frac{1}{4} f_R(n_1,n_2) + \frac{1}{2} f_G(n_1,n_2) - \frac{1}{4} f_B(n_1,n_2)\right) \\ &\quad \times (-1)^{n_1+n_2} \\ &\quad + \left(-\frac{1}{4} f_R(n_1,n_2) + \frac{1}{4} f_B(n_1,n_2)\right) \\ &\quad \times \left((-1)^{n_1}-(-1)^{n_2}\right) \\ &\triangleq f_L(n_1,n_2) + f_{C1}(n_1,n_2)(-1)^{n_1+n_2} \\ &\quad + f_{C2}(n_1,n_2)\left((-1)^{n_1}-(-1)^{n_2}\right). \end{aligned} \tag{2.6}$$

Moreover, using the identity $-1 = \exp(j\pi)$ yields

$$\begin{aligned} f_{CFA}(n_1,n_2) &= f_L(n_1,n_2) + f_{C1}(n_1,n_2)\exp\left(j2\pi\left(n_1+n_2\right)/2\right) \\ &\quad + f_{C2}(n_1,n_2)\left(\exp\left(j2\pi n_1/2\right)-\exp\left(j2\pi n_2/2\right)\right) \end{aligned} \tag{2.7}$$

where term f_L can be considered as a luminance component at baseband, term f_{C1} as a first chrominace component modulated at spatial frequency $(0.5, 0.5)$, and term f_{C2} as a second chrominance component modulated at spatial frequencies $(0.5, 0)$ and $(0, 0.5)$. The modulation becomes even clearer if we take the Fourier transform

of f_{CFA}:

$$\begin{aligned} F_{CFA} &= F_L(u_1, u_2) \\ &\quad + F_{C1}(u_1 - 0.5, u_2 - 0.5) \\ &\quad + F_{C2}(u_1 - 0.5, u_2) + F_{C2}(u_1, u_2 - 0.5)\,. \end{aligned} \tag{2.8}$$

In Fig. 2.2, which plots the Fourier transform magnitude of the Kodak "lighthouse" CFA image, the modulated luminance-chrominance components are easily visible.

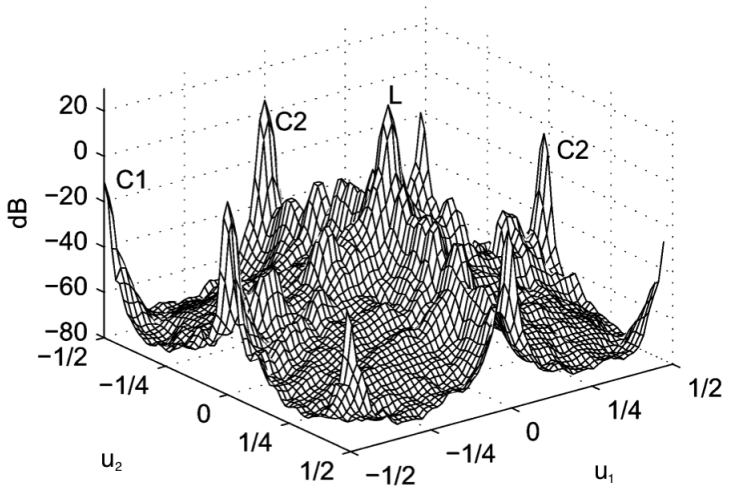


Figure 2.2: Fourier transform magnitude of the Kodak "lighthouse" image.

The relationship between the RGB components and the luminance-chrominance ones, previously used in (2.6), is given by

$$\begin{bmatrix} f_L \\ f_{C1} \\ f_{C2} \end{bmatrix} \triangleq \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ -\frac{1}{4} & \frac{1}{2} & -\frac{1}{4} \\ -\frac{1}{4} & 0 & \frac{1}{4} \end{bmatrix} \begin{bmatrix} f_R \\ f_G \\ f_B \end{bmatrix} \tag{2.9}$$

$$\begin{bmatrix} f_R \\ f_G \\ f_B \end{bmatrix} \triangleq \begin{bmatrix} 1 & -1 & -2 \\ 1 & 1 & 0 \\ 1 & -1 & 2 \end{bmatrix} \begin{bmatrix} f_L \\ f_{C1} \\ f_{C2} \end{bmatrix} \tag{2.10}$$

and this relationship enforces the interpretation of f_{C1} and f_{C2} as chrominance components, as it shows that for an achromatic signal, that is a signal with $f_R = f_G = f_B$,

signals f_{C1} and f_{C2} happen to be zero.

The importance of the result by Alleyson et al. comes from the low energy and small spatial bandwith usually components f_{C1} and f_{C2} exhibit if compared to the luminance one, f_L. This is well illustrated in Fig.2.2: the peak of F_{C1} at spatial frequency $(0.5, 0.5)$, and those of F_{C2} at spatial frequencies $(0.5, 0)$ and $(0, 0.5)$, are smaller that the peak of F_L at baseband, moreover F_{C1} and F_{C2} show a faster decay than F_L, i.e., smaller bandwith.

These interesting properties of luminance-chrominance components suggest a simple demosaicing algorithm:

1. estimate f_{C1} from its modulated copy at spatial frequency $(0.5, 0.5)$ in f_{CFA} using an appropriate bandpass filter and applying a demodulation step;

2. estimate f_{C2} from one of its modulated copy at spatial frequencies $(0.5, 0)$ and $(0, 0.5)$ in f_{CFA} using an appropriate bandpass filter and applying a demodulation step;

3. estimate f_L from f_{CFA} using a lowpass filter;

4. estimate triplet f_R, f_G, f_B, from the estimated f_L, f_{C1}, f_{C2}, using (2.10).

Unfortunately this algorithm happens to be naive: in fact, although chrominance components have a faster decay than luminance, components F_L and F_{C2} usually exhibit a significative spectral overlap, or *crosstalk*, that introduces noticeable artifacts in the reconstructed image. In Fig. 2.2 the crosstalk between F_L and F_{C2} is well visible. Since F_{C2} has a faster decay than F_L, surely the estimated f_{C2} is the most affected by the crosstalk phenomenon: that is why the artifacts produced by the "naive" algorithm mainly concern the introduction of false colors in the reconstructed image.

In order to avoid the crosstalk phenomenon, most of the color digital cameras use an anti-aliasing filter, but this only attenuates the problem, so frequency-based demosaicing algorithms need to take explicitly into account the crosstalk problem.

A simple but quite effective solution has been proposed by Dubois [6], based on the smart observation that two modulated copies of f_{C2} exist, each one with a different overlap with the luminance component f_L in the frequency domain. Let define the modulated components of f_{C2} at spatial frequencies $(0.5, 0)$ and $(0.5, 0)$

as $f_{C2ma}(n_1, n_2) \triangleq f_{C2}(n_1, n_2)(-1)^{n_1}$ and $f_{C2mb}(n_1, n_2) \triangleq f_{C2}(n_1, n_2)(-1)^{n_2}$ respectively. As Fig. 2.2 shows, the high horizontal frequencies in f_{C2ma} mainly overlap with high horizontal frequencies of f_L, while the high vertical frequencies of f_{C2ma} are almost crosstalk-free. Conversely, the high vertical frequencies in f_{C2mb} mainly overlap with high vertical frequencies of f_L, while the high horizontal frequencies of f_{C2mb} are almost crosstalk-free. Therefore, Dubois suggested to use a pair of complementary asymmetric filters to recover the vertical frequencies of f_{C2} mainly from f_{C2ma}, and the horizontal ones mainly from f_{C2mb}, thus obtaining a better estimate of f_{C2}. Once estimated f_{C1} as in the "naive algorithm", the algorithm estimates the luminance component f_L through equation (2.7). Finally (2.10) allows the transition from the luminance-chrominance domain to the RGB one.

Dubois's algorithm, as many others, do not solve the crosstalk problem, but tries to handle it. The spectral overlap between f_L and f_{C2} has caused some information to be definitely lost, so this information can now be only estimated, and this is the goal of Dubois's algorithm. However a completely different approach could take into account the design of a different CFA performing a better "packing" of the chrominance components in the frequency plane, thus avoiding the call for a crosstalk handling: this is the subject of the next section.

2.2 Color filter array design

In [9] Hirakawa and Wolfe extended the analysis of the previous section to a general pure-color (rectangular and periodic) CFA, showing that, in this more general case too, the resulting CFA image can be represented as a linear combination of a luminance component at baseband and two chrominance components modulated at spatial frequencies dictated by the adopted CFA. Starting from this result, in [9] Hirakawa and Wolfe also addressed the problem of the design of a CFA avoiding the aliasing between the luminance component and any modulated chrominance copy, phenomenon previously refered to as crosstalk. The two authors base their argumentation on the following three assumptions, which allow a formal definition of the "optimal CFAs" they are looking for:

1. the spectral support of the luminance component and those of the chrominance ones are bounded and contained in balls of radii $r_L < \pi$ and $r_C < \pi$

respectively;

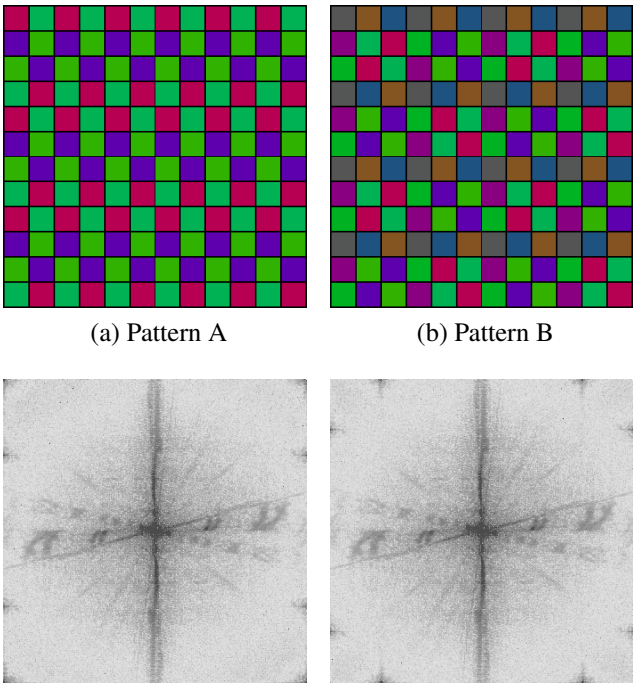
2. $r_L + r_C > \pi$, implying that aliasing between luma and chrominance components may occur, depending on the placement of chrominance modulated copies in the frequency domain;

3. $r_L > r_C$, as this relation is consistent with empirically reported results in the literature.

A CFA is thus defined "optimal" if it maximizes the spectral radii r_L and r_C subject to a zero-aliasing constraint. Surely Assumption 1 is not met in practice, since neither luminance nor chrominances are bounded, however the call for a maximization of the spectral radii promotes a minimization of the aliasing.

Hirakawa and Wolfe formally showed that the three assumptions above imply that every optimal CFA requires chrominance replicates to be placed along the perimeter of the $[-\pi, \pi) \times [-\pi, \pi)$ frequency plane, except at frequencies $(-\pi, 0)$ and $(0, -\pi)$, since *Assumption 2* assumes $r_L + r_C > \pi$. More formally, in a optimal CFA, chrominance copies must be modulated at frequencies belonging to the set $S \triangleq \{(u_1, u_2) : \max\{|u_1|, |u_2|\} = \pi\} \setminus \{(-\pi, 0), (0, -\pi)\}$. Moreover the two authors proved that every pure-color CFA is sub-optimal in this sense, as it cannot satisfy the previous constraint on chrominance replicates positioning. Because of this result, optimal CFAs are necessary panchromatic.

Hirakawa and Wolfe thus propose to design optimal CFAs directly in the Fourier domain, as this enables to specify the chrominance replicates positions in the frequency plane. Clearly, replicates positions have to be chosen according to set S in order for the CFA to be optimal. The resulting CFA can then be obtained via the inverse Fourier transform. Fig. 2.3 shows two of the four optimal CFAs presented in [9] together with the Fourier transform log-magnitudes of the Kodak "lighthouse" image filtered with these CFAs.



(a) Pattern A (b) Pattern B

(c) Pattern A (d) Pattern B

Figure 2.3: Two periodic CFA patterns proposed by Hirakawa and Wolfe (top row), and the corresponding "lighthouse" log-magnitude spectra (bottom row).

Since CFA of type A in Fig.2.3 will be employed later in this thesis, here we give its expression in terms of matrices $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$, as for the Bayer CFA:

$$\begin{aligned}
\boldsymbol{\alpha}_{i,j} &= \frac{1}{4}\left[\sqrt{2}\cos(\pi i)\cos\left(\frac{\pi}{2}j-\frac{\pi}{4}\right)+\cos(\pi i)\cos(\pi j)+2\right] \\
\boldsymbol{\beta}_{i,j} &= \frac{1}{4}\left[\sqrt{2}\cos(\pi i)\cos\left(\frac{\pi}{2}j-\frac{\pi}{4}\right)-\cos(\pi i)\cos(\pi j)+2\right] \\
\boldsymbol{\gamma}_{i,j} &= \frac{1}{2}\left[1-\sqrt{2}\cos(\pi i)\cos\left(\frac{\pi}{2}j-\frac{\pi}{4}\right)\right]
\end{aligned} \tag{2.11}$$

with $(i, j) \in [s_1] \times [s_2]$.

2.3 Sparse-based demosaicing

Let us consider equation (2.2), which states the relation between the target image and the acquired one. Once considered the vectorized forms of matrices $\boldsymbol{y}$, $\boldsymbol{R}$, $\boldsymbol{G}$,

$\boldsymbol{B} \in \mathbb{R}^{s_1 \times s_2}$, namely vectors y, R, G, $B \in \mathbb{R}^N$, with $N = s_1 s_2$, and defined the $N \times N$ diagonal matrices $\bar{\boldsymbol{\alpha}}$, $\bar{\boldsymbol{\beta}}$, and $\bar{\boldsymbol{\gamma}}$, as

$$\bar{\boldsymbol{\alpha}} \triangleq diag\,(\alpha) \quad \bar{\boldsymbol{\beta}} \triangleq diag\,(\beta) \quad \bar{\boldsymbol{\gamma}} \triangleq diag\,(\gamma) \tag{2.12}$$

with α, β, $\gamma \in \mathbb{R}^N$ the vectorized forms of matrices $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\gamma} \in \mathbb{R}^{s_1 \times s_2}$, then equation (2.2) can be rewritten as follows:

$$y = \left[\bar{\boldsymbol{\alpha}}\, \bar{\boldsymbol{\beta}}\, \bar{\boldsymbol{\gamma}}\right] \begin{bmatrix} R \\ G \\ B \end{bmatrix}. \tag{2.13}$$

Moreover, once defined the *CFA matrix* $\boldsymbol{\Phi} \in \mathbb{R}^{N \times 3N}$ as

$$\boldsymbol{\Phi} \triangleq \left[\bar{\boldsymbol{\alpha}}\, \bar{\boldsymbol{\beta}}\, \bar{\boldsymbol{\gamma}}\right], \tag{2.14}$$

and the vectorized target image $x \in \mathbb{R}^{3N}$ as

$$x \triangleq \begin{bmatrix} R \\ G \\ B \end{bmatrix}, \tag{2.15}$$

equation (2.13) can be further rewritten as

$$y = \boldsymbol{\Phi} x. \tag{2.16}$$

Equation 2.16 clearly shows that the target image x is related to the acquired one y through a linear system of equations, therefore demosaicing, which concerns the recovery of x from y, should take into account the solution of system (2.16). Unfortunately this system happens to be under-determined, as it has more rows than columns, an thus a infinite number of solutions exist. This poses the problem of how to recover from an infinite set of solutions the only one representing the target image.

In engineering, problems formulated as under-determined systems of linear equations are often encountered, and a common way to approach them is via "regularization", a tecnique where a function $J\,(x)$ evaluating the desiderability of each

solution x is introduced, and solutions leading to small values of $J(x)$ are preferred. The target signal, an image in our case study, is thus estimated solving the following optimization problem:

$$(P_J): \qquad \min_x \; J(x) \quad subject\ to \quad y = \mathbf{\Phi} x. \tag{2.17}$$

The choice of $J(x)$ is clearly critical, as it governs the quality of the estimated target signal, but also affects the feasibility of problem P_J, and a problem which is too time demanding may not be solved in pratice. Moreover the solution to P_J may not be unique.

The most common choice of $J(x)$ is the squared Euclidean norm $\|x\|_2^2$, also known in the DSP community as the *energy* of x. Its wide usage is due to its mathematical simplicity: the strict convexity of the energy function, together with the convexity of the feasible solutions space $\{x : \; y = \mathbf{\Phi} x\}$, guarantees a unique solution of P_J, which can be easly computed via the Moore-Penrose pseudo-inverse of matrix $\mathbf{\Phi}$. Obviously natural images does not necessary exibit low energy, and thus the squared Euclidean norm does not fit our problem.

Today one of the most popular schemes for signal compression is know as *Transform Coding*, and it relies on finding a basis or frame that provides sparse or compressible representations for signals in a class of interest. By a *sparse* representation, we mean that for a signal of cardinality n, we can represent it with a signal containing $k \ll n$ non-zero coefficients; by a *compressible* representation instead, we mean that the signal can be well approximated by a signal with $k \ll n$ non-zero coefficients. Clearly compression is achieved storing only the values and locations of the k non-zero coefficients. This approach, also know as *sparse approximation*, happens to be very effective for natural color images, as their strong inter-pixel and inter-channel (color) correlations allow the existance of basis or frames providing high sparsity levels, and that is why transform coding is included in successful image coding standards such as JPEG and JPEG2000. Sparse-based demosaicing exploits sparsity or compressibility of natural color images experimented under the area of sparse approximation to regularize problem (2.16).

Let us consider the set of all the $s_1 \times s_2$ natural color images in their vectorized form $x \in \mathbb{R}^{3N}$, and assume a sparsifing basis, frame, or more generally a "dictionary", $\mathbf{\Psi} \in \mathbb{R}^{3N \times M}$ with $M \geq 3N$ is provided for this class of signals. Thus, based on

the previous argumentation, our target image x can be expressed as $x = \mathbf{\Psi}\xi$, where

- either $\xi \in \mathbb{R}^M$ is k-sparse, i.e., it contains $k \ll 3N$ non-zero elements only,
- or ξ is compressible, and thus x is well approximated by $\mathbf{\Psi}\xi_k$, with $\xi_k \in \mathbb{R}^M$ a k-sparse vector containing the $k \ll 3N$ highest magnitude coefficients of ξ only.

Since $x = \mathbf{\Psi}\xi$ holds, we can rewrite equation (2.16), describing the relation between the target image x and the acquired one y, as

$$y = \mathbf{\Phi}\mathbf{\Psi}\xi \tag{2.18}$$

where, for the sake of semplicity, we assume ξ is sparse, defering to the next chapter the compressible case. System (2.18) is still under-determined, but now the solution ξ we are looking for is known to have few non-zero elements only, and this prior can be exploited to regularize the system. Sparse-based demosaicing algorithms thus propose to estimate ξ, the sparse representation of x, via the following regularization of system (2.18):

$$(P_0): \qquad \min_{\xi} \|\xi\|_0 \quad \textit{subject to} \quad y = \mathbf{\Phi}\mathbf{\Psi}\xi \tag{2.19}$$

with $\|\xi\|_0$ the number of non-zero elements of vector ξ, usually refered to as the "l_0-norm"[1] of ξ. The estimated ξ, let say $\hat{\xi}$, can then be used to estimate the target image x as $\hat{x} = \mathbf{\Psi}\hat{\xi}$.

Altough (P_0) seems a reasonable regularization, its introduction raises two interesting questions.

1) Does some theoretical result exist about the reconstruction performed via problem (P_0)*?*

It happens that problem (P_0) is a highly studied problem and there is a whole area of research, called *Sparse Recovery*, dedicated to it, as its applications go beyond

[1]Note that the term "l_0-norm" is misleading, as $\|\cdot\|_0$, the function mapping each vector to its number of non-zero elements, does not satisfy all the axiomatic requirements of a norm. Altough $\|\cdot\|_0$ satisfies the zero vector property and the triangular inequality, it does not satisfy the absolute homogeneity property. The reason why function $\|\cdot\|_0$ is usually refered to as the l_0-norm will be made clear in the next chapter.

demosaicing. Indeed, altough the previous argumentation has been centered on the demosaicing problem, it is easily seen that the starting problem (2.16) is very general, with matrix $\mathbf{\Phi}$ standing for some measurement matrix with less rows than columns. Most of the efforts in the Sparse Recovery area concern the development of conditions providing a unique solution to problem (P_0), as the l_0-norm is a non-convex function and thus a unique solution is usually not guaranteed. In particular, results from Sparse Recovery theory state that under some assumptions on matrix $\mathbf{\Phi\Psi}$ and the level of sparsity of ξ, the sparse representation of signal x over $\mathbf{\Psi}$, vector ξ happens to be the unique solution of problem (P_0), and thus (at least theoretically) vector ξ can be recovered, and so vector x, as $x = \mathbf{\Psi}\xi$ holds.

2) Can problem (P_0) *be solved efficiently?*

Unfortunately, due to the nature of the l_0-norm function, problem (P_0) can be shown to be NP-hard, and thus an efficient algorithm to solve it does not exist. The Sparse Recovery community has developed a number of algorithms trying to approach problem (P_0), and they can be clustered mainly into two groups: *greedy algorithms*, which attack directly problem (P_0), and *Basis Pursuit algorithms*, which relax problem (P_0) replacing the l_0-norm with the l_1 one. The second approach seems to be more effective, both from a theoretical point of view and from an empirical one.

A deeper argumentation on the previous two questions is provided in the next chapter, where some fundamental results in the Sparse Recovery area are presented.

Chapter 3

A Brief Introduction to Sparse Recovery Theory

Let us start by considering a signal of interest $x \in \mathbb{R}^p$. Suppose we do not have direct access to signal x, but we are provided with a projection of x into a smaller subspace of $\mathbb{R}^p$. Formally we are provided with vector $y \triangleq \mathbf{\Phi} x$, with $\mathbf{\Phi} \in \mathbb{R}^{n \times p}$ and $n < p$. Note that as matrix $\mathbf{\Phi}$, usually refered to as the *sensing matrix*, has less rows than columns, system $y = \mathbf{\Phi} x$ happens to be under-determined. Now suppose vector x admits a sparse representation $\xi \in \mathbb{R}^m$ over some basis or frame $\mathbf{\Psi} \in \mathbb{R}^{p \times m}$, thus $x = \mathbf{\Psi}\xi$ holds and we can rewrite $y = \mathbf{\Phi} x$ as $y = \mathbf{\Phi}\mathbf{\Psi}\xi$ with ξ sparse. Matrix $\mathbf{\Psi}$ is usually refered to as the *sparsifing matrix*. Once renamed the $n \times m$ matrix $\mathbf{\Phi}\mathbf{\Psi}$ as $\boldsymbol{A}$, usually refered to as the *projection matrix*, system $y = \mathbf{\Phi}\mathbf{\Psi}\xi$ can be further rewritten as $y = \boldsymbol{A}\xi$. Note that as $n < p$ and $p \leq m$ hold (the second one due to $\mathbf{\Psi}$ being a basis or frame), then $n < m$ holds too, therefore system $y = \boldsymbol{A}\xi$ is still under-determined.

Sparse Recovery efforts are directed toward understanding if vector ξ, the sparse representation of our signal of interest x, has any chance to be the unique solution of problem (P_0):

$$(P_0): \qquad \min_{\xi} \|\xi\|_0 \quad \textit{subject to} \quad y = \boldsymbol{A}\xi. \tag{3.1}$$

Note that, as $x = \mathbf{\Psi}\xi$ holds, ξ being the unique solution of (P_0) would directly translate into exact recovery of x, if we assume problem (P_0) can be solved efficently.

However, as stated at the end of Chapter 2, problem (P_0) is know to be NP-hard, thus Sparse Recovery research also concerns the development of algorithms trying to attack problem (P_0).

Sometimes, not only we do not have direct access to our signal of interest x, but its projection y via matrix $\mathbf{\Phi}$ is also assumed to be affected by some kind of additive random noise v. Vector y is thus defined as $y \triangleq \mathbf{\Phi}x + v$, with v a random vector in $\mathbb{R}^m$. For the sake of simplicity, we will assume vector v to have bounded energy, i.e., we will assume $\|v\|_2^2 \leq \epsilon^2$. As $x = \mathbf{\Psi}\xi$ with ξ sparse is assumed, the new definition of y lead us to system $y = \boldsymbol{A}\xi + v$. Due to the noise vector v, problem (P_0) is no longer appropriate to estimate ξ, therefore an error-tolerant version of (P_0), problem $\left(P_0^\epsilon\right)$, is introduced:

$$(P_0^\epsilon): \qquad \min_{\xi} \|\xi\|_0 \quad \textit{subject to} \quad \|\boldsymbol{A}\xi - y\|_2 \leq \epsilon. \tag{3.2}$$

Sections 3.1 and 3.2 provide some interesting results about the chances of recovering vector ξ in the noiseless and the noisy case, therefore problems (P_0) and $\left(P_0^\epsilon\right)$ are considered respectively. Since our vector of interest x is directly determinated by vector ξ via $x = \mathbf{\Psi}\xi$, we will assume ξ to be our target throughout this chapter.

3.1 The noiseless case

Here we assume y is not affected by noise and thus we deal with the under-determined system $y = \boldsymbol{A}\xi$ ($\boldsymbol{A} \in \mathbb{R}^{m\times n}$ with $n < m$). We restate here problem (P_0), as it will be the subject of the whole section:

$$(P_0): \qquad \min_{\xi} \|\xi\|_0 \quad \textit{subject to} \quad y = \boldsymbol{A}\xi. \tag{3.3}$$

According to the litterature, in this section we will assume matrix $\boldsymbol{A}$ has full-rank, but we note that this is not a loss of generality. The sparsifing matrix $\mathbf{\Psi}$ (with $\mathbf{\Psi} \in \mathbb{R}^{p\times m}$) is a basis or a frame and thus it has full-rank. From basic linear algebra it comes that $\mathbf{\Psi}$ having full-(row)rank implies $rank\,(\mathbf{\Phi\Psi}) = rank\,(\mathbf{\Phi})$ holds. Moreover matrix $\mathbf{\Phi}$ (with $\mathbf{\Phi} \in \mathbb{R}^{n\times p}$ and $n < p$) usually models some acquisition procedure which performes n measurements on vector x (just think to the demosaicing problem), but performing redundant measurements in an acquistion procedure

makes no sense, thus we can assume rows of $\boldsymbol{\Phi}$ to be linearly independent, and thus $\boldsymbol{\Phi}$ to have full-rank, leading to

$$rank\,(\boldsymbol{A}) = rank\,(\boldsymbol{\Phi}\boldsymbol{\Psi}) = rank\,(\boldsymbol{\Phi}) = n. \tag{3.4}$$

Matrix $\boldsymbol{A}$ thus happens to have full-rank.

A crucial key property for the study of problem (P_0) is the *spark* of matrix $\boldsymbol{A}$, a term introduced by Dohono and Elad in [10].

Definition. The *spark* of a given matrix $\boldsymbol{A}$ is the smallest number of columns from $\boldsymbol{A}$ that are linearly-dependent.

Now consider the null-space of $\boldsymbol{A}$, refered to as $ker\,(\boldsymbol{A}) \triangleq \{z \in \mathbb{R}^m :\ \boldsymbol{A}z = 0\}$. Vectors in $ker\,(\boldsymbol{A}) \setminus \{0 \in \mathbb{R}^m\}$ selects some columns in $\boldsymbol{A}$ via their non-zero coefficients, and linearly combine them into the null vector of $\mathbb{R}^n$. The columns of $\boldsymbol{A}$ selected by every vector $z \in ker\,(\boldsymbol{A}) \setminus \{0 \in \mathbb{R}^m\}$ need to be linear dependent, otherwise their linear combination can't lead to the null vector. Therefore, by definition of spark, the number of columns of $\boldsymbol{A}$ combined together by a vector $z \in ker\,(\boldsymbol{A}) \setminus \{0 \in \mathbb{R}^m\}$ must be at least equal to $spark\,(\boldsymbol{A})$, and thus $\|z\|_0 \geq spark\,(\boldsymbol{A})$ hold. Formally we have

$$\boldsymbol{A}z = 0 \ \ with \ \ z \neq 0 \quad \Longrightarrow \quad \|z\|_0 \geq spark\,(\boldsymbol{A})\,. \tag{3.5}$$

This relationship between the null space of $\boldsymbol{A}$ and its spark allows a simple criterion for claiming uniqueness of a sparse solution of problem (P_0). The criterion is stated as Theorem 1.

Theorem 1. *If a system of linear equations $\boldsymbol{A}\xi = y$ has a solution ξ obeying $\|\xi\|_0 < spark\,(\boldsymbol{A})\,/2$, this is the unique solution of (P_0).*

Because of the importance of Theorem 1, we provide its simple proof, which directly comes from result (3.5).

Proof. Consider a vector $\eta \in \mathbb{R}^m$ satisfying $\boldsymbol{A}\eta = y$ with $\eta \neq \xi$, i.e., an alternative solution to ξ. Since both $\boldsymbol{A}\eta = y$ and $\boldsymbol{A}\xi = y$ hold, we have $\boldsymbol{A}\,(\xi - \eta) = 0$. Moreover, as $\eta \neq \xi$ is assumed, is $\xi - n \neq 0$. Since $\boldsymbol{A}\,(\xi - \eta) = 0$ with $\xi - n \neq 0$ holds,

from (3.5) we have $\|\xi - \eta\|_0 \geq spark(\boldsymbol{A})$. Therefore, once applied the triangular inequality (which is trivially satisfied by the l_0-norm) we get

$$\|\xi\|_0 + \|\eta\|_0 \geq \|\xi - \eta\|_0 \geq spark(\boldsymbol{A}). \tag{3.6}$$

Since solution ξ is assumed to satisfy $\|\xi\|_0 < spark(\boldsymbol{A})/2$, (3.6) allow us to conclude $\|\eta\|_0 > spark(\boldsymbol{A})/2$ holds, therefore ξ is necessarly the sparsest possible solution, and thus the unique solution of (P_0). □

By its definition the spark of $\boldsymbol{A}$ belongs to the range $[2, n+1]$, thus the best scenario we can hope for is that with matrix $\boldsymbol{A}$ having spark equal to $n+1$, as the recovery of ξ from problem (P_0) would be guaranteed for vectors ξ having at most $n/2$ non-zero entries. However it happens that computing the spark of a general matrix $\boldsymbol{A}$ is at least as difficult as solving (P_0). A simpler way to guarantee uniqueness for problem (P_0) comes from another measure, namely the *mutual-coherence* of $\boldsymbol{A}$.

Definition. The *mutual-coherence* of a given matrix $\boldsymbol{A}$ is the largest absolute normalized inner product between different columns of $\boldsymbol{A}$. Denoting the k-th colums of $\boldsymbol{A}$ by a_k, the mutual-coherence is given by

$$\mu(\boldsymbol{A}) \triangleq \max_{1 \leq i,j \leq m,\, i \neq j} \frac{\left|a_i^T a_j\right|}{\|a_i\|_2 \left\|a_j\right\|_2}. \tag{3.7}$$

Coherence meaures the correlation among the columns of $\boldsymbol{A}$, and it is possible to shown that for a matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$, with $n < m$, coherence $\mu(\boldsymbol{A})$ is always in the range $\left[\sqrt{\frac{m-n}{n(m-1)}}, 1\right]$, with the lower bound known as the *Welch bound*.

By its definition $\mu(\boldsymbol{A})$ can be easily obtained by normalizing matrix $\boldsymbol{A}$, computing its Gramian, and then looking for the highest absolute value among the off-diagonal entries. Interestingly coherence allows to lower bound the spark. See [4] for a proof of the following Lemma.

Lemma. *For any matrix $\boldsymbol{A} \in \mathbb{R}^{n \times m}$, the following holds:*

$$spark(\boldsymbol{A}) \geq 1 + \frac{1}{\mu(\boldsymbol{A})}. \tag{3.8}$$

The above bound can be applied to Theorem 1, leading to an analog theorem now based on coherence instead of spark.

Theorem 2. *If a system of linear equations $\boldsymbol{A}\xi = y$ has a solution ξ obeying $\|\xi\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\boldsymbol{A})}\right)$, this is the unique solution of (P_0).*

The sufficient condition in Theorem 2 for vector ξ being the unique solution of (P_0) is very easy to verify due to coherence. However the introduction of coherence for lower bounding the spark makes Theorem 2 far less powerful than Theorem 1. In fact, as the Welch bound is always greater than $1/\sqrt{n}$, coherence can never be smaller than $1/\sqrt{n}$, therefore the sparsity bound in Theorem 2 is never larger than $\sqrt{n}/2$. On the other hand the spark can be as large as $n+1$, thus allowing Theorem 1 for a sparsity bound as large as $n/2$. Having a large sparsity bound happens to be really important, as when trying to recover our original signal x, we know it admits a sparse representation ξ, but we do not know how much sparse ξ is.

Now suppose sparse vector ξ is the unique solution of (P_0). How to recover ξ, as (P_0) is known to be NP-hard? As stated in Section 2.3, mainly two kind of algorithms exist: greedy algorithms, which tries to directly attack (P_0), among which *Orthogonal Matching Pursuit (OMP)* [21] is probably the most known, and the Basis Pursuit algorithms, which try to solve (P_0) via a relaxation of it named *Basis Pursuit*. Obviously algorithms mixing together the two different strategies exist too. Because of its effectiveness and wide usage in most pratical cases, here we will focus on the Basis Pursuit problem only.

Let us start by recalling the definition of the l_p-norms of $\mathbb{R}^m$,

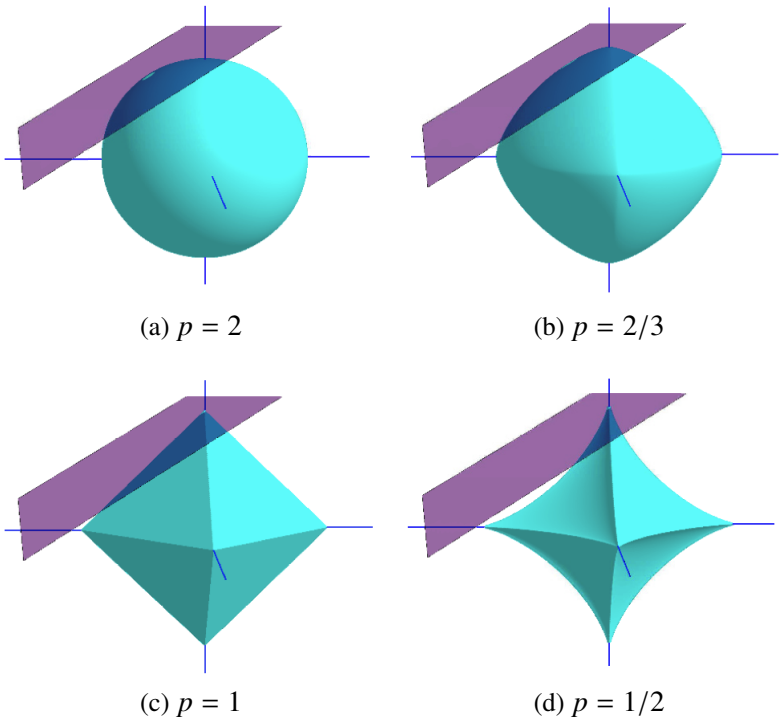
$$\|\xi\|_p \triangleq \begin{cases} \left(\sum_{i=1}^m |\xi_i|^p\right)^{\frac{1}{p}}, & p \in [1, \infty) \\ \max_{i=1,2,\ldots,m} |\xi_i|, & p = \infty, \end{cases} \tag{3.9}$$

and extend the definition to the whole positive semiaxis $p > 0$. For $0 < p < 1$ the functions obtained are no longer norms, but *quasi-norms*, as they do not satisfy the triangular inequality, but for the sake of simplicity we will use the term norm for these functions as well[1]. Now let us define problem $\left(P_p\right)$, which regularizes system $y = \boldsymbol{A}\xi$ throught a general l_p-norm with $p > 0$:

$$\left(P_p\right): \qquad \min_{\xi} \ \|\xi\|_p^p \quad \textit{subject to} \quad y = \boldsymbol{A}\xi. \tag{3.10}$$

[1]Note that $\lim_{p\to 0} \|\xi\|_p^p = \lim_{p\to 0} \sum_{i=1}^m |\xi_i|^p = card\,(supp\,(\xi)) \triangleq \|\xi\|_0$ holds. That is why $\|\xi\|_0$ is usually refered to as the l_0-norm of vector ξ.

Interestingly it happens that l_p-norms with $0 < p \leq 1$ promote sparse solutions of the system $y = \boldsymbol{A}\xi$.



(a) $p = 2$ (b) $p = 2/3$

(c) $p = 1$ (d) $p = 1/2$

Figure 3.1: Demonstration of the geometric solution process for problem $\left(P_p\right)$ with different values of p in $\mathbb{R}^3$. In each plot, the solution of problem $\left(P_p\right)$ is the intersection between the tilted hyperplane, representing the feasible solutions set, and the l_p ball.

In order to gain an intuition of this fact let us consider how to ideally solve problem $\left(P_p\right)$ from a geometric point of view. The set of linear equations $y = \boldsymbol{A}\xi$ defines a feasible set of solutions which geometrically appears as an hyperplane of dimension $\mathbb{R}^{m-n}$ embedded in the $\mathbb{R}^m$ space. Seeking the solution of $\left(P_p\right)$ can thus be done "blowing" an l_p ball centered around the origin, and stopping its growth when it first touches the solutions hyperplane. The intersection point defines the solution of problem $\left(P_p\right)$. The result of this process is presented in Fig. 3.1 for $p = 2, 2/3, 1$, and $1/2$, in the $\mathbb{R}^3$ space, with a tilted plane representing the solutions hyperplane. Figures 3.1c and 3.1d suggest that for $p \leq 1$ the intesections among the ball and the solutions hyperplane tends to take place on the ball corners, which are located on the axis of the $\mathbb{R}^3$ space. This translates into a sparse solution of $\left(P_p\right)$, as all

points on the axis have two out of three zero cordinates. Conversely Figures 3.1a and 3.1b suggest that for $p > 1$ the intersection tends to take place in non-sparse points of $\mathbb{R}^3$, that is points with three non-zero coordinates. The intuition provided by Fig. 3.1 generalizes to higher dimensions, thus we expect that the intersection between a solutions hyperplane and an l_p ball takes place on the axis of the $\mathbb{R}^m$ space for $p \leq 1$, leading to a sparse solution.

As Fig. 3.1 shows, the shape of an l_p ball gets more tight to the axis as $p \to 0$, thus better promoting sparsity of the solution of $\left(P_p\right)$. Based on this remark it seems reasonable to replace the l_0-norm in (P_0) with some quasi-norm with a small value of p. Unfortunately every value of p in $]0, 1[$ makes $\left(P_p\right)$ a non-convex optimization problem, and this raises some difficulties in its solution. The l_1-norm instead is a convex function, and this makes (P_1) a convex optimization problem, for which very effective solving tools are available. Moreover it can be shown that problem (P_1) can be easily restated as a *Linear Programming (LP)* optimization problem, therefore classical methods such as the *Simplex Algorithm*, or the *Interior Point Algorithm*, can be used. Problem (P_1) is also known as *Basis Pursuit*:

$$(P_1): \qquad \min_{\xi} \|\xi\|_1 \quad \textit{subject to} \quad y = \boldsymbol{A}\xi. \tag{3.11}$$

Beyond the intuition provided by the previuos argumentation, theoretical results about solving (P_0) via the Basis Pursuit problem (P_1) exist. See [4] for a proof of the following Theorem.

Theorem 3. *For a system of linear equations $\boldsymbol{A}\xi = y$ ($\boldsymbol{A} \in \mathbb{R}^{n \times m}$ full-rank with $n < m$), if a solution ξ exists obeying $\|\xi\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\boldsymbol{A})}\right)$, this is both the unique solution of (P_1) and the unique solution of (P_0).*

Previous Theorem 2 claimed that, under the assumption of our target vector ξ being sparse enough, vector ξ could have been recovered as the unique solution of (P_0). However perfect recovery could be obtained theoretically only, as (P_0) was known to be NP-hard. Now, not so surprisingly because of the previous argumentation on Fig. 3.1, Theorem 3 states that, under the same hypotesis of Theorem 2 guaranteeing ξ as the unique solution of (P_0), ξ can be practically recovered via Basis Pursuit (P_1).

Altough Theorem 3 is theoretically important, its result is weak: the theorem

guarantees successful recovery of target vector ξ only when ξ is extremely sparse, and this is not the case in most practical cases. However it has been empirically noticed, by simulation, that Basis Pursuit manages to recover the sparse vector ξ, or a good approximation of it, also in situations violating the hypotesis of Theorem 3.

3.2 The noisy case

As stated at the beginning of the chapter, the noisy case deals with the underdetermined system $y = \boldsymbol{A}\xi + v$ ($\boldsymbol{A} \in \mathbb{R}^{m \times n}$ with $n < m$) and problem $\left(P_0^\epsilon\right)$:

$$(P_0^\epsilon): \qquad \min_{\xi} \|\xi\|_0 \quad subject\ to \quad \|\boldsymbol{A}\xi - y\|_2 \leq \epsilon, \tag{3.12}$$

with $\|v\|_2^2 \leq \epsilon^2$.

Here we consider how well the approximation of target vector ξ provided by $\left(P_0^\epsilon\right)$ could be, and if a practical algorithm for the recovery of this approximation exists, as $\left(P_0^\epsilon\right)$ is still an NP-hard problem.

Quite not surprisingly, in general one can no longer expect target vector ξ to be the unique solution of problem $\left(P_0^\epsilon\right)$ under some sparsity constraint, as happened in the noiseless case. This is clearly due to the error-tolerant constraint in $\left(P_0^\epsilon\right)$, and here we will try to give an intuition of this fact. Let $S \triangleq supp(\xi)$, therefore, according to our notation, let $\xi_S \in \mathbb{R}^{|S|}$ be the subvector of ξ restricted to S, and $\boldsymbol{A}_{:,S} \in \mathbb{R}^{n \times |S|}$ be the submatrix of $\boldsymbol{A}$ restricted to the columns of S. For the sake of simplicity, we will assume all vectors in the set of feasible solutions of $\left(P_0^\epsilon\right)$ have some support $S' \subseteq S$. Obviously vector ξ is a feasible solution, as it is our target vector. We can distinguish two cases.

1. Vector ξ_S is the minimizer of function $f_S(z) \triangleq \left\|\boldsymbol{A}_{:,S} z - y\right\|_2$ over $\mathbb{R}^{|S|}$ and $f_S(\xi_S) = \epsilon$ holds. Note that $f_S(z)$ has a unique minimizer due to its being strictly convex. Under these assumptions vector ξ happens to be the unique feasible solution of $\left(P_0^\epsilon\right)$, as any pertubation of its entries over support S would lead to a violation of the constraint in $\left(P_0^\epsilon\right)$. As the unique feasible solution of $\left(P_0^\epsilon\right)$, vector ξ is the unique solution of $\left(P_0^\epsilon\right)$ too.

2. Vector ξ_S, which satisfies the constraint $\left\|\boldsymbol{A}_{:,S}\xi_S - y\right\|_2 \leq \epsilon$, is no longer the minimizer of $f_S(z)$ over $\mathbb{R}^{|S|}$. This assumption allows to perturb target vector

ξ over support S in such a way that the resulting vector still satisfies the constraint in $\left(P_0^\epsilon\right)$ and has some support $S' \subseteq S$. Note that the perturbation technique allows the creation of a continuum of vectors satisfying the constraint in $\left(P_0^\epsilon\right)$ while having support included or equal to S.

The first case allows, at least theoretically, exact recovery of ξ as the unique solution of $\left(P_0^\epsilon\right)$. However it represents a very particular situation, as its assumptions are far from being met in practice. On the contrary, the assumption of the second case matches the general situation, and in this case we have no hope to exactly recover our target vector ξ, as the set of feasible solutions of $\left(P_0^\epsilon\right)$ contains a continuum of vectors whose l_0-norms are smaller or equal to the l_0-norm of ξ. As vectors in the feasible solutions set are at least as good as our target vector ξ, we have no chance to identify ξ among the other solutions. Moreover we note that our initial assumption about feasible solutions having some support $S' \subseteq S$ is a loss of generality, implying the set of feasible solutions can contain an even wider number of vectors at least as good as ξ.

Altough in the general we cannot hope to recover target vector ξ as the unique solution of $\left(P_0^\epsilon\right)$, it happens that, under some assumptions of its sparsity, a bound on the error between ξ and its approximation via $\left(P_0^\epsilon\right)$ can be claimed. See [4] for a proof of the following theorem.

Theorem 4. *Consider the instance of problem $\left(P_0^\epsilon\right)$ defined by the triplet ($\boldsymbol{A}$, y, ϵ). Suppose that a sparse vector $\xi \in \mathbb{R}^m$ is a feasible solution of $\left(P_0^\epsilon\right)$ satisfying the sparsity constraint $\|\xi\|_0 < \frac{1}{2}\left(1 + \frac{1}{\mu(\boldsymbol{A})}\right)$. Every solution ξ' of $\left(P_0^\epsilon\right)$ must obey*

$$\|\xi' - \xi\|_2^2 \leq \frac{4\epsilon^2}{1 - \mu(\boldsymbol{A})\left(2\|\xi'\|_0 - 1\right)}. \tag{3.13}$$

This result parallels that of Theorem 2, and it reduces to it when $\epsilon = 0$ holds, i.e., no noise is involved.

As in the noiseless case, the NP-hard problem $\left(P_0^\epsilon\right)$ can be attacked by relaxing the l_0-norm with the l_1 one, thus leading to problem $\left(P_1^\epsilon\right)$, known as *Basis Pursuit Denoising*:

$$(P_1^\epsilon): \qquad \min_{\xi} \|\xi\|_1 \quad \textit{subject to} \quad \|\boldsymbol{A}\xi - y\|_2 \leq \epsilon \tag{3.14}$$

Even in this case, as in the noiseless one, theoretical bounds on the solution

provided replacing the l_0-norm with the l_1 one exist. The following result parallels that of Theorem 3 for Basis Pursuit. See [4] for a proof of the following theorem.

Theorem 5. *Consider the instance of problem* $\left(P_1^\epsilon\right)$ *defined by the triplet* $(\boldsymbol{A}, y, \epsilon)$. *Suppose that a sparse vector* $\xi \in \mathbb{R}^m$ *is a feasible solution of* $\left(P_1^\epsilon\right)$ *satisfying the sparsity constraint* $\|\xi\|_0 < \frac{1}{4}\left(1 + \frac{1}{\mu(\boldsymbol{A})}\right)$. *Every solution* ξ' *of* $\left(P_1^\epsilon\right)$ *must obey*

$$\|\xi' - \xi\|_2^2 \leq \frac{4\epsilon^2}{1 - \mu(\boldsymbol{A})\left(4\|\xi'\|_0 - 1\right)}. \tag{3.15}$$

For $\epsilon = 0$ this result reduces to the noiseless case. However the result provided here shows a loss of tightness compared to the noiseless case, as it calls for half the sparsity required by Theorem 3. Moreover, assuming our target vector ξ obeys $\|\xi\|_0 < \frac{1}{4}\left(1 + \frac{1}{\mu(\boldsymbol{A})}\right)$, the bound on the difference between ξ and its approximation via Basis Pursuit Denosing is proportional to $4\epsilon^2$, that is four times the noise energy, thus suggesting Basis Pursuit Denosing performes no denoising despite its name. However, as observed for Basis Pursuit in the noiseless case, it has been empirically noticed that Basis Pursuit Denosing performs well in practice, meaning that, even in situations violating the hypothesis of Theorem 5, Basis Pursuit Denoising shows good denoising capabilities, thus leading to good approximations of ξ in general.

This clear gap between the results predicted for Basis Pursuit and Basis Pursuit Denoising by Theorems 3 and 5, respectively, and the better results they achieve in practice, is due to the kind of analysis behind these theorems. The analysis presented both in the noiseless and noisy case is a worst-case (deterministic) analysis, as it aims at providing recovery bounds holding for every signal. This obviously leads to overpessimistic bounds, as are those provided by Theorems 3 and 5. However a probabilistic analysis allowing a small recovery failure rate could lead to more optimistic bounds, better representing Basis Pursuit and Basis Pursuit Denoising results observed in practice. Altough most of the available results on sparse recovery concerns the worst-case analysis, probabilistic results have been developed too, and most of them have been developed under a quite recent branch of sparse recovery, named *Compressive Sensing* [24]. Unfortunately, but quite not surprisingly, these results call for a matrix $\boldsymbol{A}$ having some particular structure or entries distribution, thus restricting the applicability of the results.

We conclude this section showing that the noisy problem formulation, $y = \boldsymbol{A}\xi +$

v, can be used to handle the case of our original signal of interest x admiting a representation ξ which is not exactly sparse, but compressible, as it is most often the case in practice.

Assume vector ξ is compressible, thus, from Section 2.3, we know our original target signal $x \in \mathbb{R}^p$ is well approximated by vector $\boldsymbol{\Psi}\xi_k$, with $\xi_k \in \mathbb{R}^m$ a k-sparse vector containing the $k \ll p$ highest magnitude coefficients of ξ only. Once defined vectors $d \triangleq \xi - \xi_k$ and $e \triangleq \boldsymbol{\Psi}d$, the following chain of equalities holds:

$$\begin{aligned} y &= \boldsymbol{\Phi}\boldsymbol{\Psi}\xi + v \\ &= \boldsymbol{\Phi}\boldsymbol{\Psi}\xi + \boldsymbol{\Phi}\boldsymbol{\Psi}\xi_k - \boldsymbol{\Phi}\boldsymbol{\Psi}\xi_k + v \\ &= \boldsymbol{\Phi}\boldsymbol{\Psi}\xi_k + \boldsymbol{\Phi}\boldsymbol{\Psi}d + v \\ &= \boldsymbol{A}\xi_k + \boldsymbol{A}e + v. \end{aligned} \tag{3.16}$$

Since we assumed vector x to be well approximated by vector $\boldsymbol{\Psi}\xi_k$, error vector e can be assumed to have very small energy, quite negligible. However, as $\boldsymbol{A}$ is not an orthonormal matrix, vector $\boldsymbol{A}e$ could have, at least potentially, an arbitrary large energy. From linear algebra theory, for every vector $z \in \mathbb{R}^m$, the following holds:

$$\lambda_{min}\left(\boldsymbol{A}^T\boldsymbol{A}\right) \cdot \|z\|_2^2 \leq \|\boldsymbol{A}z\|_2^2 \leq \lambda_{max}\left(\boldsymbol{A}^T\boldsymbol{A}\right) \cdot \|z\|_2^2 \tag{3.17}$$

with λ_{min} and λ_{max} the smallest and largest eigenvalues of matrix $\boldsymbol{A}^T\boldsymbol{A}$ respectively. Therefore if matrix $\boldsymbol{A}$ is such that λ_{max} is not too large, as vector e is assumed to have quite negligible energy, then vector $\boldsymbol{A}e$ will have quite negligible energy too. The quite negligible energy of $\boldsymbol{A}e$ allows us to "hide" vector $\boldsymbol{A}e$ inside random noise v, leading us to $y = \boldsymbol{A}\xi_k + v$. Now we can use $\left(P_0^\epsilon\right)$, or more precisely $\left(P_1^\epsilon\right)$, to approximate ξ_k. Cleaerly one can try to adjust the parameter ϵ in order to take into account vector $\boldsymbol{A}e$, despite its negligible energy.

Obviously the assumption on λ_{max} being not too large represents a loss of generality, and thus one has to check matrix $\boldsymbol{A}$ before to procede using $\left(P_0^\epsilon\right)$. Interestingly enough, in our case study, the sparse-based demosaicing algorithm presented in Chapter 4, λ_{max} happens to be smaller than 10, therefore small enough to allow exploiting the previous expedient.

Chapter 4

A Sparse-Based Demosaicing Framework

The approach behind sparse-based demosaicing algorithms for color natural images has been introduced in Section 2.3. Here we consider a concrete case study: the demosaicing algorithm proposed by Moghadam et al. in [11].

To the best of our knowledge only three sparse-based demosaicing algorithms have been proposed up to now: algorithm in [15] by Elad et al., algorithm in [18] by Mairal et al., and the one reported here. Clearly all this algorithms are based on (P_0), but they differ by some changes applied to this optimization problem, the approach used to solve it, the way the sparsifying basis, frame, or, more generally, the sparsifying dictionary is built, and the particular way they divide the CFA image into patches. Altough in Section 2.3 we assumed to apply problem (P_0) directly to the whole image, in practice the image is divided into patches, each one leading to a different under-determined linear system, and problem (P_0) is usually applied to each one of them for their recovery.

Altough the algorithms by Elad et al. and Mairal et al. usually perform much better than non-sparse-based demosaicing algorithms, they have an important limitation which concerns their execution time. Since they use an adaptive strategy which requires to learn the sparsifying dictionary directly from the CFA image, they have to learn the dictionary at run-time. Moreover the algorithm by Mairal et al. does not learn a unique dictionary for all the patches, but learns a different dictionary for different groups of patches clustered together based on their similarity.

This adaptive strategy indeed allows to achieve high sparsification of the patches in the original image, and thus provides a high quality reconstruction of it. However online learning is usually a time-consuming procedure, which could slow down the whole reconstruction process.

In order to avoid the computational burden that online learning procedures usually call for, while achieving good sparsification of the original full-resolution image, in their algorithm Moghadam et al. propose to build offline a dictionary which explicitly takes into account the inter-pixel and inter-channel correlation that natural color images usually exibit. Interestingly the authors claim that this strategy allows to achieve state-of-the-art reconstruction results. Moreover, the algorithm by Moghadam et al. works with every CFA, while the algorithms by Elad et al. and Mairal et al. are designed solely for the Bayer filter, thus limiting their application.

Before to start presenting the algorithm by Moghadam et al., it is necessary to consider the notation we are going to use along this chapter. We will adopt exactly the notation introduced in Section 2.3, here applied to the single patches instead of the whole image, as the algorithm demosaics the single patches and not the whole image. As this notation will be used along the whole chapter, we formally restate it here.

Let us consider a square $b \times b$ patch from the original full-resolution image, numbered as the i-th patch, and let $\boldsymbol{R}^{(i)}$, $\boldsymbol{G}^{(i)}$, $\boldsymbol{B}^{(i)} \in \mathbb{R}^{b \times b}$ be the red, green, and blue channels of the patch respectively. Let $\boldsymbol{y}^{(i)} \in \mathbb{R}^{b \times b}$ be the acquired patch, thus $\boldsymbol{y}^{(i)} = \boldsymbol{\alpha}^{(i)} \odot \boldsymbol{R}^{(i)} + \boldsymbol{\beta}^{(i)} \odot \boldsymbol{G}^{(i)} + \boldsymbol{\gamma}^{(i)} \odot \boldsymbol{B}^{(i)}$ holds, with $\boldsymbol{\alpha}^{(i)}, \boldsymbol{\beta}^{(i)}, \boldsymbol{\gamma}^{(i)} \in \mathbb{R}^{b \times b}$ the red, green, and blue weights describing the portion of the whole camera CFA responsible for the acquisition of the i-th patch. Switching to the vectorized notation, as done in Section 2.3, we get

$$y^{(i)} = \left[diag\left(\alpha^{(i)}\right)\ diag\left(\beta^{(i)}\right)\ diag\left(\gamma^{(i)}\right) \right] \begin{bmatrix} R^{(i)} \\ G^{(i)} \\ B^{(i)} \end{bmatrix}. \tag{4.1}$$

Once defined the $n \times 3n$, $n = b^2$, CFA matrix $\boldsymbol{\Phi}$ as

$$\boldsymbol{\Phi}^{(i)} \triangleq \left[diag\left(\alpha^{(i)}\right)\ diag\left(\beta^{(i)}\right)\ diag\left(\gamma^{(i)}\right) \right], \tag{4.2}$$

and the vectorized form of the full-resolution patch as

$$x^{(i)} = \begin{bmatrix} R^{(i)} \\ G^{(i)} \\ B^{(i)} \end{bmatrix}, \tag{4.3}$$

equation (4.1) can be easily rewritten as

$$y^{(i)} = \mathbf{\Phi}^{(i)} x^{(i)}. \tag{4.4}$$

Once introduced the sparsifying dictionary $\mathbf{\Psi}^{(i)} \in \mathbb{R}^{3n \times p}$ for the i-th patch, $x^{(i)} = \mathbf{\Psi}^{(i)} \xi^{(i)}$ holds, with $\xi^{(i)} \in \mathbb{R}^p$ the sparse representation of $x^{(i)}$, thus equation (4.4) can be rewritten as

$$y^{(i)} = \mathbf{\Phi}^{(i)} \mathbf{\Psi}^{(i)} \xi^{(i)}. \tag{4.5}$$

Similarly to the other sparsed-based demosaicing algorithms, the algorithm by Moghadam et al. is comprised of three major stages.

1. *Dividing the CFA image into patches.* The acquired image is divided into square patches $y^{(i)}$ such that the union of all the patches covers the whole image, i.e., each pixel belongs to at least one patch. Since each patch $y^{(i)}$ will be demosaiced saparately, in order to avoid block effects in the reconstructed image usually some level of overlap among adjacent patches is adopted.

2. *Patches reconstruction.* The algorithm selects a sparsifying dictionary $\mathbf{\Psi}^{(i)}$ for patch $y^{(i)}$. Altough the sparsifying dictionary is built offline, different versions of it may exist, depending on the patch division strategy at point 1. Then the algorithm considers the under-determined system presented above, and here restated:

   $$y^{(i)} = \mathbf{\Phi}^{(i)} \mathbf{\Psi}^{(i)} \xi^{(i)}. \tag{4.6}$$

   Note that the algorithm assume no noise is involved. The $\xi^{(i)}$ representation is thus estimated via Basis Pursuit as follows:

   $$\hat{\xi}^{(i)} = \arg\min_{\xi} \|\xi\|_1 \quad \textit{subject to} \quad y^{(i)} = \mathbf{\Phi}^{(i)} \mathbf{\Psi}^{(i)} \xi. \tag{4.7}$$

   Finally the full-resolution patch $x^{(i)}$ is estimated as $\hat{x}^{(i)} = \mathbf{\Psi}^{(i)} \hat{\xi}^{(i)}$. Clearly we

would like to use the l_0-norm in problem (4.7) but, as we known from Chapter 3, the l_1-norm represents a good compromise. This step is repeated for each patch $y^{(i)}$.

3. *Post-processing.* Depending on the adopted level of overlap among patches, a number of estimates for the three colors of each pixel is available. Based on these estimates, this stage of the algorithm tries to find the best approximation for each pixel. Finally, in order to remove possible reconstruction artifacts from the whole reconstructed image, some classical post-processing, such as median filtering, is performed.

In the following sections the three basic stages of the algorithm are explained in detail. Section 4.1 deals with the core of the algorithm: the costruction of the sparsifying dictionary. Section 4.2 presents the way the CFA image is divided into patches. Altough the patch division stage precedes the employement of the sparsifying dictionary in the algorithm, here their presentation order is reversed, as the patch division strategy is motivated by a particular choice in the dictionary construction procedure. Finally, Section 4.3 explains how to exploit the available estimates of each pixel to approximate the original one. Section 4.4 concludes the chapter with an argumentation on the best CFA for the presented algorithm.

4.1 Dictionary building

The idea of Moghadam et al. about the employment of a dictionary explicitly taking into account the inter-pixel (spatial) and inter-channel (color) correlation, was developed for the first time in [12]. The dictionary employeed in the algorithm presented here represents an enhanced version.

The difference between the first version of the dictionary, that of [12], and the enhanced one, mainly concerns the way the two sparsifying components of the dictionary, the one handling the spatial correlation, and the one handling the color correlation, interact. In the first dictionary the two components are separated, while in the second one they are mixed together: for this reason the first dictionary is usually refered to as the *separable* dictionary, and the second one as the *non-separable* dictionary. We will start by presenting the separable one, as this will automatically

lead us to the non-separable case. Altough the following argumentation considers the development of a dictionary for a single $b \times b$ patch, for the sake of simplicity, we will omit the superscripts in the notation.

Let the $b \times b$ matrix $\boldsymbol{D}$ be the one dimensional *Discrete Cosine Transfom (DCT)* matrix for $\mathbb{R}^b$. By definition, the two-dimensional DCTs of matrices $\boldsymbol{R}, \boldsymbol{G}, \boldsymbol{B} \in \mathbb{R}^{b \times b}$, respectively the red, green, and blue channels of the patch considered, are

$$\hat{\boldsymbol{R}} = \boldsymbol{D}\boldsymbol{R}\boldsymbol{D}^T, \qquad \hat{\boldsymbol{G}} = \boldsymbol{D}\boldsymbol{G}\boldsymbol{D}^T, \qquad \hat{\boldsymbol{B}} = \boldsymbol{D}\boldsymbol{B}\boldsymbol{D}^T. \tag{4.8}$$

Since $\boldsymbol{D}$ is an orthonormal matrix, equations in (4.8) can be rewritten as

$$\boldsymbol{R} = \boldsymbol{D}^T\hat{\boldsymbol{R}}\boldsymbol{D}, \qquad \boldsymbol{G} = \boldsymbol{D}^T\hat{\boldsymbol{G}}\boldsymbol{D}, \qquad \boldsymbol{B} = \boldsymbol{D}^T\hat{\boldsymbol{B}}\boldsymbol{D}. \tag{4.9}$$

From linear algebra theory, the following statement about matrices and their vectorized forms holds:

$$\boldsymbol{Z} = \boldsymbol{U}\boldsymbol{V}\boldsymbol{W} \qquad \Longleftrightarrow \qquad Z = \left(\boldsymbol{W}^T \otimes \boldsymbol{U}\right) V, \tag{4.10}$$

remembering that, according to this thesis notation, Z and V represent the vectorized forms of matrices $\boldsymbol{Z}$ and $\boldsymbol{V}$ respectively.

Statement (4.10) allows us to rewrite equations in (4.9) in the vectorized form. Thus, once defined the $n \times n$ matrix $\boldsymbol{\varphi} \triangleq \left(\boldsymbol{D}^T \otimes \boldsymbol{D}^T\right)$, we have

$$R = \boldsymbol{\varphi}\hat{R}, \qquad G = \boldsymbol{\varphi}\hat{G}, \qquad B = \boldsymbol{\varphi}\hat{B}. \tag{4.11}$$

The costruction of the separable dictionary starts expressing each channel of vector x, our vectorized patch of interest, with respect to the 2D-DCT basis $\boldsymbol{\varphi}$. From (4.11), the following holds:

$$x = \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varphi} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varphi} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\varphi} \end{bmatrix} \begin{bmatrix} \hat{R} \\ \hat{G} \\ \hat{B} \end{bmatrix} = (\boldsymbol{I}_3 \otimes \varphi) \begin{bmatrix} \hat{R} \\ \hat{G} \\ \hat{B} \end{bmatrix}. \tag{4.12}$$

As the 2D-DCT basis is known to have spatial decorrelation (sparsifying) properties, vector $\left[\hat{R}^T\ \hat{G}^T\ \hat{B}^T\right]^T$ should provide a sparse representation of vector x. How-

ever, for each $i = 1, 2, \ldots, n$, the transform coefficients $\hat{R}_i$, $\hat{G}_i$, $\hat{B}_i$ still exibit color correlation, thus allowing to seek a sparse representation of vector $\left[\hat{R}_i\ \hat{G}_i\ \hat{B}_i\right]^T$, and thus a sparser representation of x.

Altough equation (4.12) is very simple, and clearly states the relationship between the original representation x and the transformed one, in order to present how to further sparsify each vector $\left[\hat{R}_i\ \hat{G}_i\ \hat{B}_i\right]^T$, we need to reformulate it. Therefore we restate equation (4.12) as follows:

$$x = \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} \boldsymbol{\varphi}_{:,\{1\}} & \mathbf{0} & \mathbf{0} & \boldsymbol{\varphi}_{:,\{2\}} & \mathbf{0} & \mathbf{0} & & \boldsymbol{\varphi}_{:,\{n\}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\varphi}_{:,\{1\}} & \mathbf{0} & \mathbf{0} & \boldsymbol{\varphi}_{:,\{2\}} & \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\varphi}_{:,\{n\}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\varphi}_{:,\{1\}} & \mathbf{0} & \mathbf{0} & \boldsymbol{\varphi}_{:,\{2\}} & & \mathbf{0} & \mathbf{0} & \boldsymbol{\varphi}_{:,\{n\}} \end{bmatrix} \begin{bmatrix} \hat{R}_1 \\ \hat{G}_1 \\ \hat{B}_1 \\ \hat{R}_2 \\ \hat{G}_2 \\ \hat{B}_2 \\ \vdots \\ \hat{R}_n \\ \hat{G}_n \\ \hat{B}_n \end{bmatrix}, \tag{4.13}$$

which can be rewritten in a more compact form as

$$x = \left[\boldsymbol{I}_3 \otimes \boldsymbol{\varphi}_{:,\{1\}}\ \boldsymbol{I}_3 \otimes \boldsymbol{\varphi}_{:,\{2\}}\ \cdots \boldsymbol{I}_3 \otimes \boldsymbol{\varphi}_{:,\{n\}}\right] \begin{bmatrix} \hat{R}_1 \\ \hat{G}_1 \\ \hat{B}_1 \\ \hat{R}_2 \\ \hat{G}_2 \\ \hat{B}_2 \\ \vdots \\ \hat{R}_n \\ \hat{G}_n \\ \hat{B}_n \end{bmatrix}. \tag{4.14}$$

Altough equation (4.14) is not as intuitive as equation (4.12), one can easily show that it has been obtained from (4.12) via a proper permutation of the columns of

matrix $\boldsymbol{I}_3 \otimes \boldsymbol{\varphi}$.

In order to try removing the color correlation involved in each vector $\left[\hat{R}_i\ \hat{G}_i\ \hat{B}_i\right]^T$, in the separable dictionary Moghadam et al. propose to express $\left[\hat{R}_i\ \hat{G}_i\ \hat{B}_i\right]^T$ with respect to the frame $\boldsymbol{\theta} \triangleq [\boldsymbol{\theta}_{ETF}\ \boldsymbol{\theta}_{YUV}]$, with $\boldsymbol{\theta}_{ETF}$ the *Equiangular Tight Frame (ETF)* of $\mathbb{R}^3$, and $\boldsymbol{\theta}_{YUV}$ the *YUV* basis. Therefore we get the following relationship:

$$\begin{bmatrix} \hat{R}_i \\ \hat{G}_i \\ \hat{B}_i \end{bmatrix} = \boldsymbol{\theta}\xi_{(i)}, \tag{4.15}$$

with $\xi_{(i)}$ the representation of vector $\left[\hat{R}_i\ \hat{G}_i\ \hat{B}_i\right]^T$ with respect to the frame $\boldsymbol{\theta}$. For completeness we provide the ETF and the YUV basis:

$$\boldsymbol{\theta}_{ETF} = \frac{1}{\sqrt{1+\lambda^2}} \begin{bmatrix} 0 & 0 & 1 & 1 & \lambda & -\lambda \\ 1 & 1 & \lambda & -\lambda & 0 & 0 \\ \lambda & -\lambda & 0 & 0 & 1 & 1 \end{bmatrix}, \qquad \lambda = \frac{1+\sqrt{5}}{2}, \tag{4.16}$$

$$\boldsymbol{\theta}_{YUV} = \begin{bmatrix} 1 & 0 & 1.139 \\ 1 & -0.394 & -0.580 \\ 1 & 2.032 & 0 \end{bmatrix}. \tag{4.17}$$

From the expressions of $\boldsymbol{\theta}_{ETF}$ and $\boldsymbol{\theta}_{YUV}$, frame $\boldsymbol{\theta}$ is revealed to be a 3×9 real matrix and thus $\xi_{(i)}$ a vector in $\mathbb{R}^9$. Applying equation (4.15) to all the n vectors $\left[\hat{R}_i\ \hat{G}_i\ \hat{B}_i\right]^T$, we get the following equality:

$$\begin{bmatrix} \hat{R}_1 \\ \hat{G}_1 \\ \hat{B}_1 \\ \hat{R}_2 \\ \hat{G}_2 \\ \hat{B}_2 \\ \vdots \\ \hat{R}_n \\ \hat{G}_n \\ \hat{B}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{\theta} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\theta} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\theta} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\theta} \end{bmatrix} \begin{bmatrix} \xi_{(1)} \\ \xi_{(2)} \\ \xi_{(3)} \\ \vdots \\ \xi_{(n)} \end{bmatrix}. \tag{4.18}$$

Now, using (4.14) and (4.18), we can express x as

$$\begin{aligned} x &= \left[\boldsymbol{I}_3 \otimes \boldsymbol{\varphi}_{:,\{1\}}\ \boldsymbol{I}_3 \otimes \boldsymbol{\varphi}_{:,\{2\}}\ \cdots\ \boldsymbol{I}_3 \otimes \boldsymbol{\varphi}_{:,\{n\}}\right] \begin{bmatrix} \boldsymbol{\theta} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\theta} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\theta} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\theta} \end{bmatrix} \xi \\ &= \left[\boldsymbol{\theta} \otimes \boldsymbol{\varphi}_{:,\{1\}}\ \boldsymbol{\theta} \otimes \boldsymbol{\varphi}_{:,\{2\}}\ \cdots\ \boldsymbol{\theta} \otimes \boldsymbol{\varphi}_{:,\{n\}}\right] \xi, \end{aligned} \tag{4.19}$$

with $\xi \triangleq \left[\xi_{(1)}^T\ \xi_{(2)}^T\ \xi_{(3)}^T\ \ldots\ \xi_{(n)}^T\right]^T$ the sought sparse representation of vector x. Therefore the separable sparsifying dictionary for x, the vectorized form of our $b \times b$ full-resolution patch of interest, is defined as follows:

$$\boldsymbol{\Psi} \triangleq \left[\boldsymbol{I}_3 \otimes \boldsymbol{\varphi}_{:,\{1\}}\ \boldsymbol{I}_3 \otimes \boldsymbol{\varphi}_{:,\{2\}}\ \cdots\ \boldsymbol{I}_3 \otimes \boldsymbol{\varphi}_{:,\{n\}}\right] \begin{bmatrix} \boldsymbol{\theta} & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\theta} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\theta} & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\theta} \end{bmatrix}. \tag{4.20}$$

Once fixed the sparse vector ξ in equation (4.19), one can show that a proper permutation of the columns of $\boldsymbol{\Psi}$ allows to restate equation (4.19) as

$$x = (\boldsymbol{\theta} \otimes \boldsymbol{\varphi})\, \eta, \tag{4.21}$$

with $\boldsymbol{\theta} \otimes \boldsymbol{\varphi}$ the permuted version of $\boldsymbol{\Psi}$, and vector η the consequent permuted version of ξ. Altough dictionaries $\boldsymbol{\Psi}$ and $\boldsymbol{\theta} \otimes \boldsymbol{\varphi}$ are different, they can be used interchangeably, as they share the same columns (usually refered to as *atoms*), and thus they lead to the same sparse representation of x, up to a permutation. As in dictionary $\boldsymbol{\theta} \otimes \boldsymbol{\varphi}$ there is a clear distinction between the component of the dictionary responsible for the inter-pixel decorrelation, $\boldsymbol{\varphi}$, and the one responsible for the inter-channel decorrelation, $\boldsymbol{\theta}$, dictionary $\boldsymbol{\theta} \otimes \boldsymbol{\varphi}$, or dictionary $\boldsymbol{\Psi}$ equivalently, is refered to as the separable dictionary.

The separable ditionary employes the same frame $\boldsymbol{\theta}$ for each vector $\left[\hat{R}_i\ \hat{G}_i\ \hat{B}_i\right]^T$, i.e., for each spatial frequency of the 2D-DCT. Altough it is possible to use a dif-

ferent frame instead of that employing the ETF and the YUV basis, designing a frame providing good sparsification for all frequencies is really difficult, maybe practically infeasible. That is why the non-separable dictionary, as opposed to the separable one, considers the use of different frames for different spatial frequencies, thus increasing the chance to sparsely represent vector x. This strategy leads to re-define dictionary $\mathbf{\Psi}$ as follows:

$$\mathbf{\Psi} \triangleq \begin{bmatrix} \boldsymbol{I}_3 \otimes \boldsymbol{\varphi}_{:,\{1\}} & \boldsymbol{I}_3 \otimes \boldsymbol{\varphi}_{:,\{2\}} & \cdots & \boldsymbol{I}_3 \otimes \boldsymbol{\varphi}_{:,\{n\}} \end{bmatrix} \begin{bmatrix} \boldsymbol{\theta}_1 & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\theta}_2 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\theta}_3 & \cdots & \mathbf{0} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \cdots & \boldsymbol{\theta}_n \end{bmatrix}, \tag{4.22}$$

with $\boldsymbol{\theta}_i \in \mathbb{R}^{3 \times q_i}$ the sparsifying frame for vector $\begin{bmatrix} \hat{R}_i & \hat{G}_i & \hat{B}_i \end{bmatrix}^T$. More compactly, dictionary $\mathbf{\Psi}$ can be rewritten as

$$\mathbf{\Psi} \triangleq \begin{bmatrix} \boldsymbol{\theta}_1 \otimes \boldsymbol{\varphi}_{:,\{1\}} & \boldsymbol{\theta}_2 \otimes \boldsymbol{\varphi}_{:,\{2\}} & \cdots & \boldsymbol{\theta}_n \otimes \boldsymbol{\varphi}_{:,\{n\}} \end{bmatrix}, \tag{4.23}$$

which shows how the component of the dictionary responsible for inter-pixel decorrelation, and the one responsible for inter-channel decorrelation, this time are mixed together, thus explaining the definition of non-separable dictionary.

Altough different strategies can be adopted to design frames $\boldsymbol{\theta}_i$, hereafter refered to as *color frames*, Moghadam et al. propose to learn them from a training set. The training procedure for the costruction of color frames, one for each 2D-DCT spatial frequency, is presented in Algorithm 4.1. The set $\left\{ \boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(d)} \right\}$ in input contains a collection of $b \times b$ full-resolution patches, with $\boldsymbol{R}^{(k)}$, $\boldsymbol{G}^{(k)}$, $\boldsymbol{B}^{(k)} \in \mathbb{R}^{b \times b}$ respectively the red, green, and blue channels of patch $\boldsymbol{x}^{(k)}$, $k \in [d]$. Moreover the algorithm receives, for each $i \in [n]$, a pair (l_i, u_i) denoting respectively the lower and upper bounds for the number of atoms in the i-th color frame. Note that Algorithm 4.1 is run offline, therefore, as opposed to the algorithms by Elad et al. and Mairal et al., here the demosaicing algorithm only requires to solve one Basis Pursuit problem for each patch, dictionary $\mathbf{\Psi}$ being already available.

Altough color frames are learnt offline, in order to reduce the training time, one can chose to partition the set of the 2D-DCT spatial frequencies indexes $\{1, 2, \ldots, n\}$

Algorithm 4.1 Color Frames Training Procedure

Input: $\left\{\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \ldots, \boldsymbol{x}^{(d)}\right\}$, $\epsilon \geq 0$, and $l, u \in \mathbb{R}^n$.
Output: $\{\boldsymbol{\theta}_i,\ i = 1, 2, \ldots, n\}$.
For $i = 1$ *to* n

$$\boldsymbol{Z} = \begin{bmatrix} \left(\boldsymbol{\varphi}_{:,\{i\}}\right)^T R^{(1)} & \left(\boldsymbol{\varphi}_{:,\{i\}}\right)^T R^{(2)} & \cdots & \left(\boldsymbol{\varphi}_{:,\{i\}}\right)^T R^{(d)} \\ \left(\boldsymbol{\varphi}_{:,\{i\}}\right)^T G^{(1)} & \left(\boldsymbol{\varphi}_{:,\{i\}}\right)^T G^{(2)} & \cdots & \left(\boldsymbol{\varphi}_{:,\{i\}}\right)^T G^{(d)} \\ \left(\boldsymbol{\varphi}_{:,\{i\}}\right)^T B^{(1)} & \left(\boldsymbol{\varphi}_{:,\{i\}}\right)^T B^{(2)} & \cdots & \left(\boldsymbol{\varphi}_{:,\{i\}}\right)^T B^{(d)} \end{bmatrix};$$

For $j = l_i$ *to* u_i

$$L_j = \min_{\boldsymbol{T}_j} \left\|\boldsymbol{\alpha}_{:,\{1\}}\right\|_1 + \left\|\boldsymbol{\alpha}_{:,\{2\}}\right\|_1 + \ldots + \left\|\boldsymbol{\alpha}_{:,\{d\}}\right\|_1$$
$$s.t. \quad \left\|\boldsymbol{Z}_{:,\{k\}} - \boldsymbol{T}_j \boldsymbol{\alpha}_{:,\{k\}}\right\|_2 \leq \epsilon, \quad \forall k = 1, 2, \ldots, d;$$

End

Let a be the index such that $L_a \leq L_j\ \forall j = l_i, l_i + 1, \ldots, u_i$;

$\boldsymbol{\theta}_i = \boldsymbol{T}_a$;

End

into K subsets, and train a color frame for each subset. Interestingly, this allows to obtain an equivalent expression for dictionary $\boldsymbol{\Psi}$, but a little more compact. Let us denote the subsets with $v_i \subseteq [n]$, $i \in [K]$: as they are chosen to be a partition, for every i and j in $[K]$, $\bigcup_{i \in [K]} v_i = [n]$ and $v_i \bigcap v_j = \emptyset$ hold. It is easily seen that properly permuting the columns of $\boldsymbol{\Psi}$ in equation (4.23) one can obtain the following equivalent formulation of $\boldsymbol{\Psi}$:

$$\boldsymbol{\Psi} \triangleq \left[\boldsymbol{\theta}_1 \otimes \boldsymbol{\varphi}_{:,v_1}\ \boldsymbol{\theta}_2 \otimes \boldsymbol{\varphi}_{:,v_2}\ \cdots\ \boldsymbol{\theta}_k \otimes \boldsymbol{\varphi}_{:,v_k}\right], \tag{4.24}$$

with $\boldsymbol{\theta}_i$ the color frame learned for the i-th subset, $i \in [K]$. Note that dictionary in (4.24) reduces to the separable dictionary $\boldsymbol{\theta} \otimes \boldsymbol{\varphi}$ when the partition contains only one set, the whole set of the 2D-DCT spatial frequencies indexes $[n]$.

4.2 Patch division strategy

Moghadam et al. consider two patch divions strategies: the first very simple, the second a little more involved but providing much better results on average.

- *Fixed-size strategy.* The CFA image is divided into $b \times b$ patches, usually $b = 8$

or $b = 16$, with each patch exibiting a vertical overlap of l pixels both with the patch above and that below respectively, and an horizontal overlap of l pixels both with the patch at its left and that at its right respectively. Parameter l is allowed to range in $\{0, 1, 2, \ldots, b-1\}$.

- *Adaptive strategy.* The CFA image is initially divided into $16x16$ patches, with some level of overlap, as in the fixe-size strategy. Then a high pass filter with diagonal detection capability is applied to each 16×16 patch, and the energy of the filter output is measured[1]. Based on the measured values exceeding or not an empirically chosen treshold, the patch is marked as containing diagonal details or not, respectively. Every patch marked as containing diagonal details is left untouched, while those marked as not containing diagonal details are further divided into 8×8 overlapping patches.

The patches produced via the first or the second strategy are directly demosaiced via Basis Pursuit.

As one expects, the fixed-size strategy actually does not hide any strategy at all. Some regions of the image may be better demosaiced with patches of size 8×8, others with patches of size 16×16, but the fixed-size strategy does not take care of it. However the fixed-size strategy is easily implementable and does not introduce overheads adaptive strategies usually call for.

On the contrary, the adaptive strategy is based on two observations.

- In the fixed-size configuration, Moghadam et al. observed that, as the patch size increases, the estimation process becomes stable. More specifically, a larger block size usually leads to demosaiced images with an acceptable level of error (neither very high nor very low quality and usually color washed patches), while chosing a smaller patch size usually generates either a noisy patch or one without any visible artifact.

- In the proposed dictionary, spatial decorrelation is attacked via the 2D-DCT basis. The 2D-DCT basis contains atoms with vertical and horizontal strips, which allow to represents a patch with vertical and horizontal edges very

[1] It is empirically known that the CFA image looks like the luminance of the original image with mosaic effects. Therefore applying a filter directly to the CFA image gives a sense of the actual output we would obtain by applying the filter to the original image.

effectivelly. However the 2D-DCT basis contains no atoms with diagonal strips, thus it is much less effective in representing patches with diagonal edges, and this directly translates into a less sparse representation.

Based on the second observation, if a small patch size is adopted for demosaicing a region with diagonal edges, the patches covering that region will hardly have a sparse representation. This translates into less chances to recover the original full-resolution patches, and thus in the introduction of visible artifacts in the region considered. On the other hand, from the first observation, we know that adopting a larger block size for demosaicing the same region will spread the reconstruction error in the whole region, thus leading to less visible artifacts.

In their simulations over the Kodak dataset, Moghadam et al. built the high pass filter with diagonal detection capabilities via a learning procedure. They extracted all the possible 16×16 patches from the Kodak dataset, and among these patches they identified 5000 patches wich were best to be demosaiced directly, and 5000 wich were best to be further decomposed into 8×8 patches before being demosaiced. The filter learning was then performed using a *Support Vector Machine (SVM)* over the two sets. In their simulations Moghadam et al. observed significant visual improvement and notably PSNR boost of the images demosaiced with the adaptive strategy over that demosaiced with the fixed-size one (see Chapter 5).

4.3 Best pixel estimation

Due to the overlap among adjacent patches, multiple estimates of a single pixel are available once all the patches it belongs have been demosaiced[2]. This raises the problem of how to best estimate the pixel from the available estimates.

Let us consider a single pixel of the image and denote it as the i-th pixel. Now let us assume the i-th pixel belongs to $u \in \mathbb{N}$ patches, and let $\bar{r}_i = \{r_1, r_2, \ldots, r_u\}$, $\bar{g}_i = \{g_1, g_2, \ldots, g_u\}$, and $\bar{b}_i = \{b_1, b_2, \ldots, b_u\}$, be the sets of the available estimates for the i-th pixel, with r_j, g_j, b_j ($j \in [u]$) the estimates provided by the j-th patch for the red, green, and blue channels, respectively. The simplest way to estimate the i-th pixel would be to estimate its red, green, and blue channels, as the mean of the values in $\bar{r}_i$, $\bar{g}_i$, and $\bar{b}_i$, respectively. However this strategy attributes to each available estimate

[2]Pixels around the image borders clearly represent an exception.

triplet $\left(r_j, g_j, b_j\right)$ the same importance, not taking into account that some patches among the u considered may have been recovered better than others, and thus may provide better estimates fot the i-th pixel. This suggest to adopt a weigthed mean, where the weight of each triplet would be proportional to its quality. However the original full-resolution versions of the u patches are not available, therefore we can't assess their respective qualities from their reconstruction errors.

Moghadam et al. propose to evaluate the patches quality from their sparse or compressible representation $\hat{\xi}^{(1)}$, $\hat{\xi}^{(2)}$, ... , $\hat{\xi}^{(u)}$, obtained via Basis Pursuit at stage 2 of the algorithm. From Chapter 3 we know that our chances to recover a signal heavily depends on the level of sparsity or compressibility of its representation with respect to the sparsifying dictionary. Therefore a representation $\hat{\xi}^{(j)}$ exibiting high sparsity or compressibility may suggest that a good approximation of the original full-resolution patch has been found, while a non-sparse or non-compressible vector $\hat{\xi}^{(j)}$ may warn about a possible failure in the recovery procedure.

Formally Moghadam et al. propose to introduce function $G(\xi, a)$, defined for every vector $\xi \in \mathbb{R}^p$ and real number $a \geq 0$ as

$$G(\xi, a) \triangleq \frac{card\left(\{k \in [p] : \ |\xi_i| > a\}\right)}{p}, \tag{4.25}$$

and to estimate $\hat{r}_i$, $\hat{g}_i$, $\hat{b}_i$, respectively the red, green, and blue components of the i-th pixel, as

$$\hat{r}_i = \frac{\sum_{j=1}^{u} w_j r_j}{\sum_{j=1}^{u} w_j}, \qquad \hat{g}_i = \frac{\sum_{j=1}^{u} w_j g_j}{\sum_{j=1}^{u} w_j}, \qquad \hat{b}_i = \frac{\sum_{j=1}^{u} w_j b_j}{\sum_{j=1}^{u} w_j}, \tag{4.26}$$

with

$$w_j \triangleq \frac{1}{G\left(\hat{\xi}^{(j)}, a\right)}. \tag{4.27}$$

Function $G\left(\hat{\xi}^{(j)}, a\right)$ mesures the ratio of the number of entries of vector $\hat{\xi}^{(j)}$ with magnitude larger than a to the length of the vector. A proper tuning of parameter a allows $G\left(\hat{\xi}^{(j)}, a\right)$ to measure the level of compressibility of vector $\hat{\xi}^{(j)}$, therefore weighting each estimate j of the i-th pixel with $1/G\left(\hat{\xi}^{(j)}, a\right)$ allows estimate j to have a weight which increases when the compressibility of its patch increases, and vice versa. Note that, for $a = 0$, function $G\left(\hat{\xi}^{(j)}, a\right)$ measures exactly the sparsity of

$\hat{\xi}^{(j)}$, but this may be a too strict choice, as vector $\hat{\xi}^{(j)}$ is not exactly sparse in general.

4.4 Color filter arrays

The studies in the area of demosaicing has led to believe that a CFA which is optimal overall probably does not exist: instead one can seek a CFA which is optimal for a given demosaicing algorithm. The work by Hirakawa and Wolfe reported in Section 2.2 is a good example, as the the optimal CFAs they propose are inteded for frequency-based demosaicing algorithms.

Seeking an optimal CFA for a sparse-based demosaicing algorithm happens to be not straightfoward at all. Let us recall the general formulation of a sparse-based demosaicing algorithm provided in Section 2.3, where, for the sake of simplicity, we assumed no patch division. Thus matrices $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are the CFA matrix and the sparsifying one, respectively. We start by observing that the quality of the demosaiced image not only depends on the CFA matrix $\mathbf{\Phi}$, but on the projection matrix $\boldsymbol{A} = \mathbf{\Phi}\mathbf{\Psi}$, as any sparse recovery algorithm (Basis Pursuit or greedy ones) is oblivious to $\mathbf{\Phi}$ and $\mathbf{\Psi}$. Therefore, once matrix $\mathbf{\Psi}$ has been chosen, the CFA design procedure has to take $\mathbf{\Psi}$ into account, as matrix $\mathbf{\Phi}$ may exibit some of the good recovery properties seen in Chapter 3, such as high spark or low coherence, but matrix $\boldsymbol{A} = \mathbf{\Phi}\mathbf{\Psi}$ may not. Moreover we note that the CFA matrix has a particular structure, exactly N rows, as the CFA acquires a value for each pixel of the camera, an only three non-zero values per row, summing to one, as it linearly combines the red, green, and blue values at each pixel. Therefore, if we assume the vectorized notation, matrix $\mathbf{\Phi}$ has to be searched in a particular set, that of the admissible CFAs Ω, defined as follows (see eq. (2.14)):

$$\Omega \triangleq \left\{ \mathbf{\Phi} \in \mathbb{R}^{N \times 3N} : \mathbf{\Phi}_{i,i} + \mathbf{\Phi}_{i,N+i} + \mathbf{\Phi}_{i,2N+i} = 1 \ \forall i \in [N], \mathbf{\Phi}_{i,j} = 0 \ \textit{elsewhere} \right\}. \tag{4.28}$$

Finally, the search for the optimal CFA calls for an order function $\Xi : \mathbb{R}^{a \times b} \to \mathbb{R}^{+}$ for all possible projection matrices of size $a \times b$, such that, if $\Xi(\boldsymbol{A}) < \Xi(\boldsymbol{B})$ then matrix $\boldsymbol{A}$ is preferred over $\boldsymbol{B}$ as a projection matrix. The definition of this function

would allow to formalize the search for the optimal CFA as follows:

$$\boldsymbol{\Phi}^* = \arg\min_{\boldsymbol{\Phi}\in\Omega} \Xi\left(\boldsymbol{\Phi}\boldsymbol{\Psi}\right). \tag{4.29}$$

However this raises the problem of how to chose function Ξ. One may suggest to define $\Xi\left(\boldsymbol{A}\right) \triangleq -spark\left(\boldsymbol{A}\right)$, as we know from Chapter 3 that a high spark value is a desiderable property in Sparse Recovery theory, but from Chapter 3 we also know that computing the spark of a matrix is as hard as solving (P_0), therefore spark cannot be used in practice. Then one may suggest to define $\Xi\left(\boldsymbol{A}\right) \triangleq \mu\left(\boldsymbol{A}\right)$, as also a small mutual-coherence value is a desiderable property in Sparse Recovery theory, and coherence is very simple to compute. However coherence is know to be a worst case measure (as spark is after all), therefore it may be a misleading measure, not reflecting the actual behaviour of a Sparse Recovery algorithm on matrix $\boldsymbol{A}$.

Inspired by the work of Elad in [19], in [13] Moghadam et al. proposed to use the *Coherence Cumulative Distribution Function (CCDF)* of matrix $\boldsymbol{A}$ as a realistic measure of its optimality as a projection matrix. Denoting the k-th column of $\boldsymbol{A}$ by a_k, the CCDF of matrix $\boldsymbol{A}$ is defined as follows:

$$F_C\left(\boldsymbol{A}, g\right) \triangleq Pr\left(\frac{\left|a_i^T a_j\right|}{\left\|a_i\right\|_2 \left\|a_j\right\|_2} \geq g\right), \quad i \neq j,\ g \in [0,1], \tag{4.30}$$

with Pr denoting the probability function. Both mutual-coherence and the CCDF calls for a normalization of the columns of $\boldsymbol{A}$ and the computation of its Gramian G. However, while $\mu\left(\boldsymbol{A}\right)$ takes only the maximum among the absolute values of the off-diagonal entries of G, function $F_C\left(\boldsymbol{A}, g\right)$ considers their distribution. For an optimal projection matrix $\boldsymbol{A}$, Moghadam et al. expected $F_C\left(\boldsymbol{A}, g\right)$ to decay fast. Conversely they expected that a long-tailed $F_C\left(\boldsymbol{A}, g\right)$, i.e., a matrix $\boldsymbol{A}$ with a large number of high absolute value off-diagonal entries, should have warned about an increase in the chance of erroneus recovery. Their sparse-based demosaicing experiments in [13] confirmed their intuition. Unfortunately, CCDF is not a valid order function Ξ, as it does not provide a non-negative scalar, but a whole function in the range $[0, 1]$.

Fig. 4.1 compares three CCDFs: $F_C\left(\boldsymbol{\Phi}_B\boldsymbol{\Psi}, g\right)$, $F_C\left(\boldsymbol{\Phi}_R\boldsymbol{\Psi}, g\right)$, and $F_C\left(\boldsymbol{\Phi}_{SG}\boldsymbol{\Psi}, g\right)$, with $\boldsymbol{\Phi}_B$ the Bayer CFA, $\boldsymbol{\Phi}_R$ a (periodic) random panchromatic CFA, $\boldsymbol{\Phi}_{SG}$ one of the

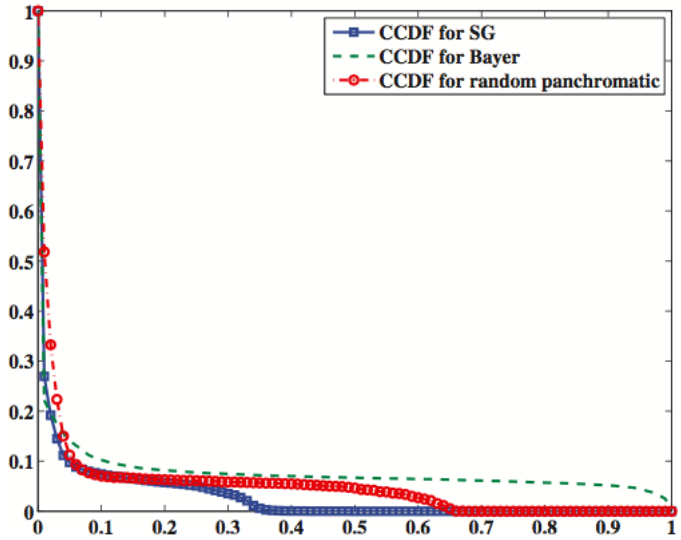


Figure 4.1: CCDF for three different CFA types.

CFAs by Hirakawa and Wolfe [9], hereafter refered to as *Second Generation (SG) CFA*s, and $\mathbf{\Psi}$ the non-separable sparsifying dictionary of Section 4.1 for patches of size 8×8. As shown in Fig. 4.1, the CCDF related to the SG CFA decays faster than the other ones, therefore we expect the SG CFA to be preferable over the random panchromatic CFA and the Bayer CFA when employed in the sparse-based demosaicing algorithm presented here. One may argue that the result in Fig. 4.1 may not be general. The algorithm by Moghadam et al. adopts a patch division strategy, therefore each $b \times b$ patch is equipped with a $b \times b$ portion of the whole CFA matrix, let say a sub-CFA. Altough the CFAs considered in Fig. 4.1 are periodic, a mismatch between the size of the patch and the CFA period, or simply the overlap among patches adopted in the algorithm, may cause each patch to be equipped with a different sub-CFA, thus making Fig. 4.1 dependent on a particular sub-CFA. Interestingly, it happens that, when employing a given CFA, the projection matrices related to each patch my change from patch to patch, but the related CCDFs exibit the same decay.

The simulations performed by Moghadam et al. over the Kodak dataset, employing the Bayer CFA and the SG one, seems to be in agreement with Fig. 4.1. In particular, altough the PSNR of images demosaiced with the SG CFA happens to be not much higher than those demosaiced with the Bayer CFA on average, the SG CFA usually leads to images with less visual artifacts, and thus more pleasant to

see. The results of the simulations by Abdolreza et al. are presented in Chapter 5, together with the results we obtained paralleling their work.

Chapter 5

Experimental Results

According to almost all the litterature on demosaicing, Moghadam et al. tested their framework on the Kodak dataset [42], which is shown in Fig. 5.1. The settings adopted in their tests are listed below.

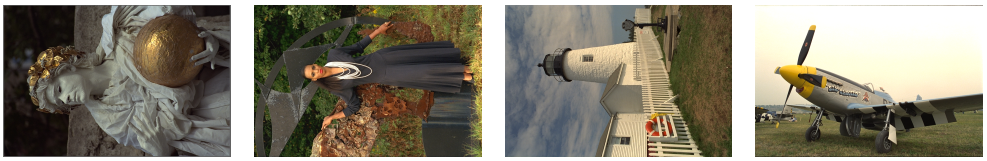
- In the sensing stage two CFAs were considered: the Bayer CFA and the SG one.
- For the patch division strategy, three scenarios were considered: the fixed-size strategy, in the 8×8 and 16×16 configurations, and the adaptive strategy.
- For the color frames, the 2D-DCT spatial frequencies were partitioned into four bands: DC, Low Frequencies (LF), Middle Frequencies (MF), and High Frequencies (HF). Each band covered one or more adjacent diagonals of the 2D-DCT spectrum, where diagonals refer to the zig-zag ordering of the 2D-DCT frequencies. In [11] Moghadam et al. does not exactly specify how many diagonals each band was comprised of. Let us denote by $\boldsymbol{\theta}_{DC}$, $\boldsymbol{\theta}_{LF}$, $\boldsymbol{\theta}_{MF}$, and $\boldsymbol{\theta}_{HF}$, the color frames assigned to each band. Moghadam et al. observed that as the spatial frequency increases toward high frequencies, a decrease on the number of atoms in the corresponding color frame usually led to higher quality demosaiced images, both in PSNR and visually. Moreover they noticed that after few diagonals in the 2D-DCT spectrum, the optimal color frame happened to be comprised of atom $\left[\frac{1}{\sqrt{3}}\ \frac{1}{\sqrt{3}}\ \frac{1}{\sqrt{3}}\right]^T$ only, which is a vector along the luminance axis (see eq. (4.17)). Based on these observations, Moghadam et al. built $\boldsymbol{\theta}_{DC}$, $\boldsymbol{\theta}_{LF}$, and $\boldsymbol{\theta}_{MF}$, via Algorithm 4.1, constraining

(a) Image 1 (b) Image 2 (c) Image 3 (d) Image 4

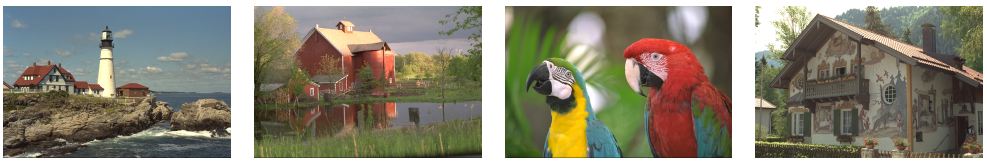
(e) Image 5 (f) Image 6 (g) Image 7 (h) Image 8

(i) Image 9 (j) Image 10 (k) Image 11 (l) Image 12

(m) Image 13 (n) Image 14 (o) Image 15 (p) Image 16



(q) Image 17 (r) Image 18 (s) Image 19 (t) Image 20



(u) Image 21 (v) Image 22 (w) Image 23 (x) Image 24

Figure 5.1: The Kodak dataset.

their number of atoms to 30, 10, and 4, respectively, while they directly set $\boldsymbol{\theta}_{HF} = \left[\frac{1}{\sqrt{3}} \ \frac{1}{\sqrt{3}} \ \frac{1}{\sqrt{3}}\right]^T$.

- Color frames $\boldsymbol{\theta}_{DC} \in \mathbb{R}^{3\times30}$, $\boldsymbol{\theta}_{LF} \in \mathbb{R}^{3\times10}$, and $\boldsymbol{\theta}_{MF} \in \mathbb{R}^{3\times4}$, were trained on a set containing a few thousand patches picked at random from the set of all the possible Kodak patches [39].

- Algorithm 4.1 was implemented via the `mexTrainDL` function of the *SPArse Modeling Software (SPAMS)* library [40].

- The Basis Pursuit Problem, employed in the reconstruction of each single patch, was solved via the `mexLasso` function of SPAMS, which implements the LARS algorithm [23].

Moghadam et al. compared their simulation results with those of three leading demosaicing algorithms of the litterature: DL [34], LPA [35], and LSSC [18]. While DL and LPA are non-sparse-based demosaicing algorithms, LSSC is a sparse-based one, in particular it is the algorithm by Mairal et al. cited in Chapter 4. The PSNRs of the Kodak images demosaiced via the framework by Moghadam et al. are reported in Table 5.1, together with the PSNRs of the other demosaicing algorithms. In order to guarantee a fair comparison with the other algorithms, a border of 15 pixels was excluded when computing the PSNRs. For the sake of simplicity, hereafter we will refer to the framework configuration employing the fixed-size patch division strategy and the adaptive one, as the "standard" configuration and the "filtering" one, respectively.

Let us start by concentrating on the standard configuration of the framework. As we expected from the argumentation in Section 4.2, employing the SG CFA instead of the Bayer one, on average guarantees higher PSNRs, regardless of the patch size. Moreover, as Fig. 5.2 shows, the Bayer CFA can lead to some visual artifacts the SG CFA usually avoids. Table 5.1 also shows that a slightly higher average PSNR is achieved when employing the 16×16 patch size rather than the 8×8 one, regardless of the CFA. Based on this observation one may argue that the 16×16 patch size is always preferable over the 8×8 one. However we believe that, both with the Bayer CFA and the SG one, the distance between the average PSNR of the 16×16 patch size configuration and that of the 8×8 one is too small to argue that a patch size is preferable over the other one.

Table 5.1: Comparing the PSNRs of the images demosaiced via algorithms DL, LPA, LSSC, and the algorithm by Moghadam et al. in all its configurations. Here, M denotes the algorithm by Moghadam et al., while B and SG denote that the Bayer CFA and the SG one were employed at the sensing stage, respectively. Finally, 8, 16, and FI, denote that the 8×8 standard configuration, the 16×16 standard configuration, and the filtering one, were adopted respectively.

Img.	LSSC	DL	LPA	M-B-8	M-B-16	M-SG-8	M-SG-16	M-SG-FI
1	41.36	38.46	39.45	37.68	37.71	38.28	41.30	41.51
2	42.24	40.89	41.36	38.54	38.52	40.92	40.52	41.27
3	44.24	42.66	43.47	36.96	37.30	41.54	39.59	42.56
4	42.45	40.49	40.84	40.94	41.07	39.85	39.43	40.86
5	39.45	38.07	37.51	37.62	37.81	37.64	36.59	38.57
6	41.71	40.19	40.92	39.10	39.27	39.32	41.18	41.71
7	44.06	42.35	43.06	41.44	41.60	42.09	40.93	42.57
8	37.57	35.58	37.13	35.20	35.41	36.76	38.64	38.75
9	43.83	43.05	43.50	42.19	42.45	42.48	42.37	43.43
10	43.33	42.54	42.77	42.19	42.25	41.87	41.75	42.91
11	41.51	40.01	40.51	38.99	39.25	39.33	40.44	40.80
12	44.90	43.45	44.01	43.36	43.50	43.63	43.07	44.52
13	36.35	34.75	36.08	33.91	34.11	32.97	36.41	36.43
14	38.77	36.91	36.86	35.86	36.17	36.09	34.29	36.52
15	41.74	39.82	40.09	39.13	39.02	38.86	33.88	39.77
16	44.91	43.75	44.02	42.54	42.75	42.44	44.65	44.73
17	41.98	41.68	41.75	40.78	41.07	41.06	42.28	42.33
18	38.38	37.64	37.59	36.93	36.78	36.40	37.94	38.46
19	42.31	41.01	41.55	39.97	40.13	41.30	42.21	42.65
20	42.27	41.24	41.48	40.42	40.57	41.04	41.52	42.37
21	40.65	39.10	39.61	38.25	38.48	38.74	40.71	40.98
22	39.24	38.37	38.44	37.90	38.16	38.57	38.32	39.03
23	44.34	43.22	43.92	38.70	38.69	42.55	39.18	42.95
24	35.89	35.55	35.44	34.74	36.74	35.13	36.68	36.80
Avg.	41.40	40.03	40.47	38.89	39.03	39.54	39.75	40.94

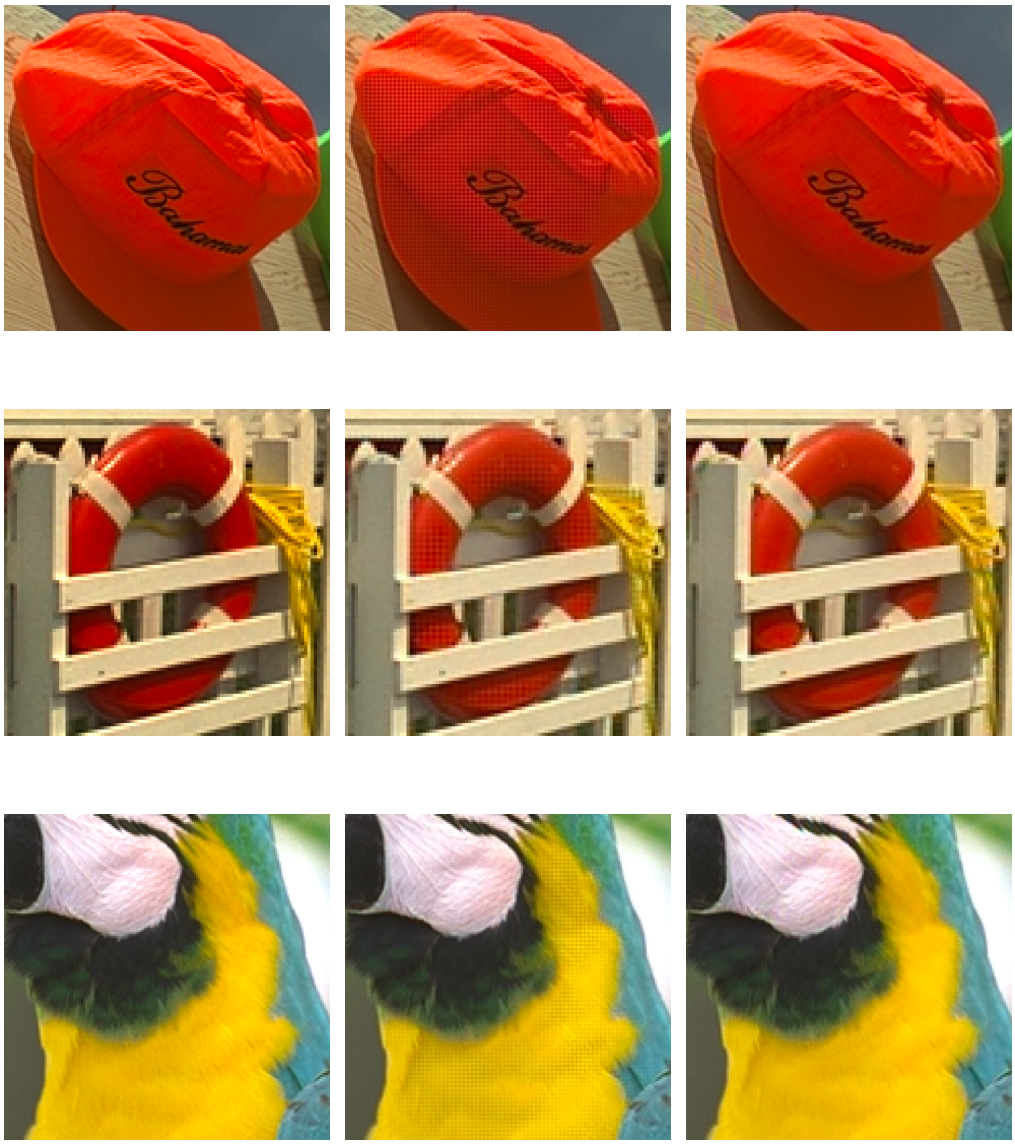


Figure 5.2: Effect of the CFA on the demosaiced image quality. Demosaicing via the 8×8 standard configuration of the framework by Moghadam et al. is considered. The top, central, and bottom rows refer to details of Kodak images 3, 19, and 23, respectively. For each row, on the left is the original detail, while in the middle and on the right are the details from the demosaiced images related to the employement of the Bayer CFA and the SG one, respectively.

Table 5.1 shows that the filtering configuration of the framework (together with the SG CFA) happens to outperform the standard one, suggesting that adopting a different patch size for different regions of the image can be an effective strategy. Moreover, Table 5.1 shows that, on average, the filtering configuration of the framework outperforms DL and LPA algorithms too, but does not manage to outperform LSSC, which exibit a fair average PSNR margin over the other algorithms. The higher PSNRs of LSSC are due to its optimal dictionary, learnt directly on the CFA image during the demosaicing process. However, as observed in Chapter 4, this is a time-consuming strategy, therefore Moghadam et al. argued that the PSNRs of the LSSC algorithm should be considered only as upper-bounds. Interestingly we note that the framework by Moghadam et al., in its filtering configuration, manages to get quite close to the LSSC average PSNR, up to half a decibel, and manages to outperform it on nine images, some with an interesting margin. Moreover the framework by Moghadam et al. is biased toward improving the visual quality of the demosaiced image, rather than its PSNR value. The choice of employing, in each color frame of the HF band, only the vector along the luminance axis goes toward this direction. This choice leads to a sparsifying dictionary $\mathbf{\Psi}$ which, at high spatial frequencies, is able to span only the luminance component, but not the chrominace ones. Clearly this translates into a considerable error in regions of the image with fast color changes. On the other hand, this regions usually happen to be really difficult to reconstruct without introducing relevant color artifacts. Therefore algorithms which do not take this problem into account may produce images exibiting low error in this region, but noticeable color artifacts. On the other hand, the human visual system is known to be much less sensitive to fast color changes, therefore the considerable error introduced by the framework happens not to be visible. Fig. 5.3-5.9 present some instances where the image demosaiced via the framework by Moghadam et al. exibit less visual artifacts, thus higher visual quality, when compared to DL, LPA, and LSSC, regardless of the PSNR measure. Each figure contains six images: three on the top row and three on the bottom one. Each image is a detail from one of the Kodak images. In the top row, on the left is the original image, in the middle is the image demosaiced via DL, and on the right the image demosaiced via LPA. In the bottom row, on the left is the image demosaiced via LSSC, in the middle is the image demosaiced via the framework by Moghadam et

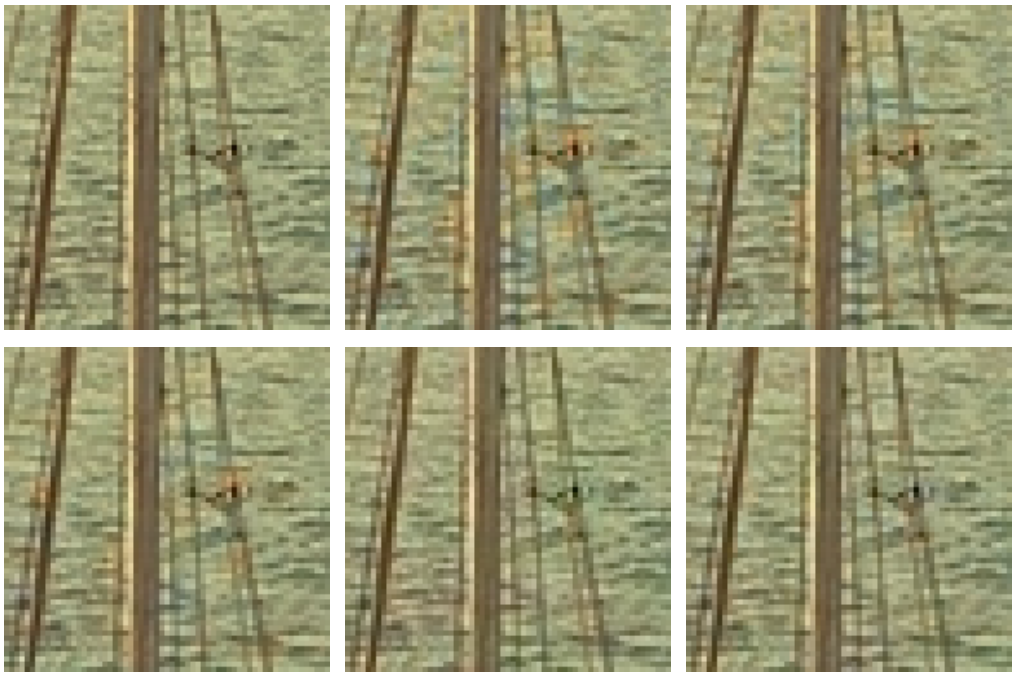


Figure 5.3: Comparing the visual quality provided by different demosacing algorithm on a detail of Kodak image 6.

al. in its filtering configuration, and on the right is the image demosaiced by our implementation of the 8×8 standard configuration of the framework by Moghadam et al. Note that, for both the images demosaiced via the framework by Moghadam et al., the SG CFA was employed at the sensing stage, while for the other algorithms the Bayer CFA was employed.

Equipped with our implementation of the framework, we tried to repeat the above tests. We focused on the standard configuration of the framework, i.e., that employing the fixed-size patch division strategy. The reason of this choice will be explained later in this section. Repeating the above tests required to choose some implementative details that Moghadame et al. did not specify in [11]. We list our implementative choices below.

- In the 8×8 patch size configuration, we assigned the first two diagonals after the DC component to the LF band, the next four diagonals to the MF band, and the remaining ones to the HF band.

- In the 16×16 patch size configuration, we assigned the first five diagonals

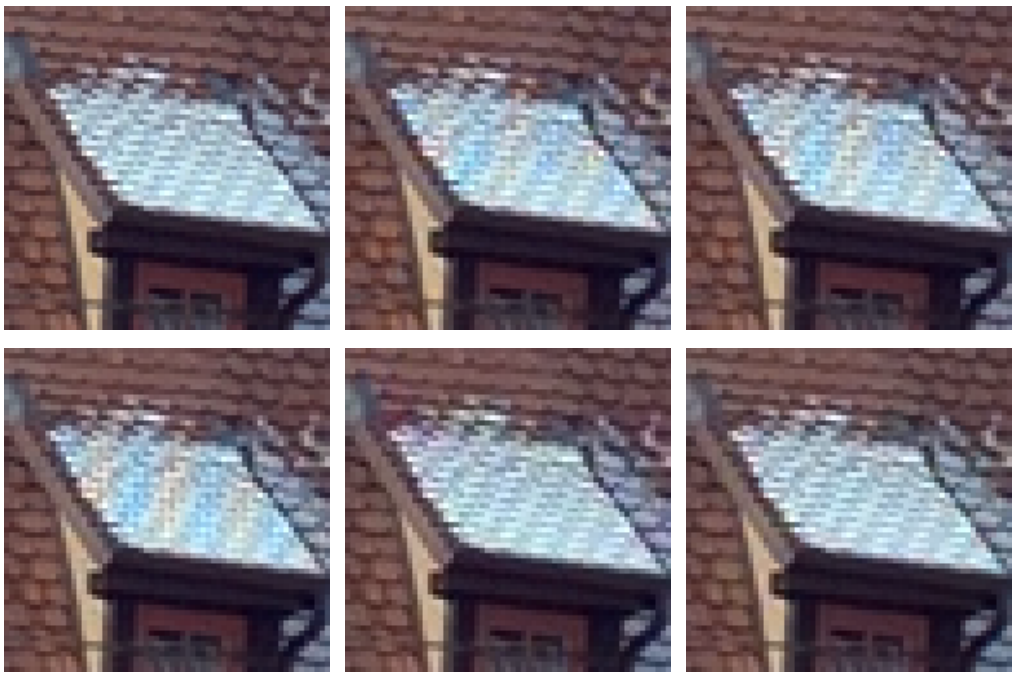


Figure 5.4: Comparing the visual quality provided by different demosacing algorithm on a detail of Kodak image 8.

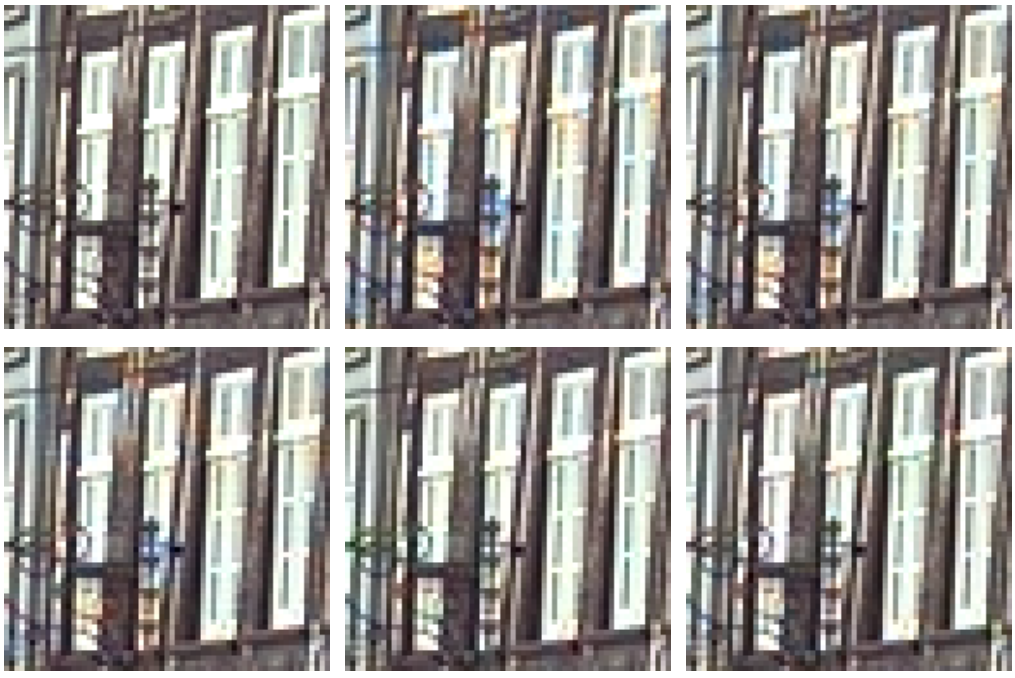


Figure 5.5: Comparing the visual quality provided by different demosacing algorithm on a detail of Kodak image 8.

Figure 5.6: Comparing the visual quality provided by different demosacing algorithm on a detail of Kodak image 19.

Figure 5.7: Comparing the visual quality provided by different demosacing algorithm on a detail of Kodak image 21.

Figure 5.8: Comparing the visual quality provided by different demosacing algorithm on a detail of Kodak image 21.

Figure 5.9: Comparing the visual quality provided by different demosacing algorithm on a detail of Kodak image 24.

after the DC components to the LF band, the next nine diagonals to the MF band, and the remaining ones to the HF band.

- As opposed to Moghadam et al., in our simulations we did not trained the color frames $\boldsymbol{\theta}_{DC}$, $\boldsymbol{\theta}_{LF}$, and $\boldsymbol{\theta}_{MF}$, on a set of a few thousand patches picked at random from the set of all the possible Kodak patches. Despite the small cardinality of the set they considered, if compared to the set of all possible Kodak patches, we believed it was not a fair choice. Therefore, when demosaicing the i-th Kodak image, we employed a triplet $(\boldsymbol{\theta}_{DC}, \boldsymbol{\theta}_{LF}, \boldsymbol{\theta}_{MF})$ trained on a set of patches picked from all the possible Kodak patches, except those coming from the i-th image.

- Regardelss to the patch size, we added to $\boldsymbol{\theta}_{DC}$ the canonical basis of $\mathbb{R}^3$. This choice raised from observing that when employing the Bayer CFA at the sensing stage, the demosaiced image may still exibit the CFA pattern in some of its regions. This can be noted in the central column of Fig. 5.2, whose images were all sensed with the Bayer CFA and then demosaiced. This phenomenon was observed when demosacing a uniform region characterized by a very bright or very saturated color, which did not appear in any of the patches used to train $\boldsymbol{\theta}_{DC}$. A uniform region is mainly characterized by the DC component of its red, green, and blue channels, therefore its reconstruction heavely depends on $\boldsymbol{\theta}_{DC}$. We observed that introducing the canonical basis of $\mathbb{R}^3$ in $\boldsymbol{\theta}_{DC}$ allowed the Basis Pursuit solver (`mexLasso` in our case) to better approximate the DC component of the three channels, thus avoiding the Bayer CFA pattern appearing in the demosaiced image. Fig. 5.10 compares the demosaiced images in the central column of Fig. 5.2 with those obtained still employing the Bayer CFA at the sensing stage, but including the canonical basis.

- The SG CFA adopted by Moghadam et al., the one that in Section 4.4 was told to exibit a fast decay CCDF when combined with the non-separable sparsifying dictionary, was not provided in [11]. In our simulations we tried all the four SG CFAs in [9], but no one led us to a decay similar to the one described by Moghadam et al. This made us believe that probably the matrix $\boldsymbol{\Phi}_{SG}\boldsymbol{\Psi}$ employed by Moghadam et al., with $\boldsymbol{\Phi}_{SG}$ the SG CFA matrix, and $\boldsymbol{\Psi}$ the sparsifying non-separable dictionary, undergone some post-processing

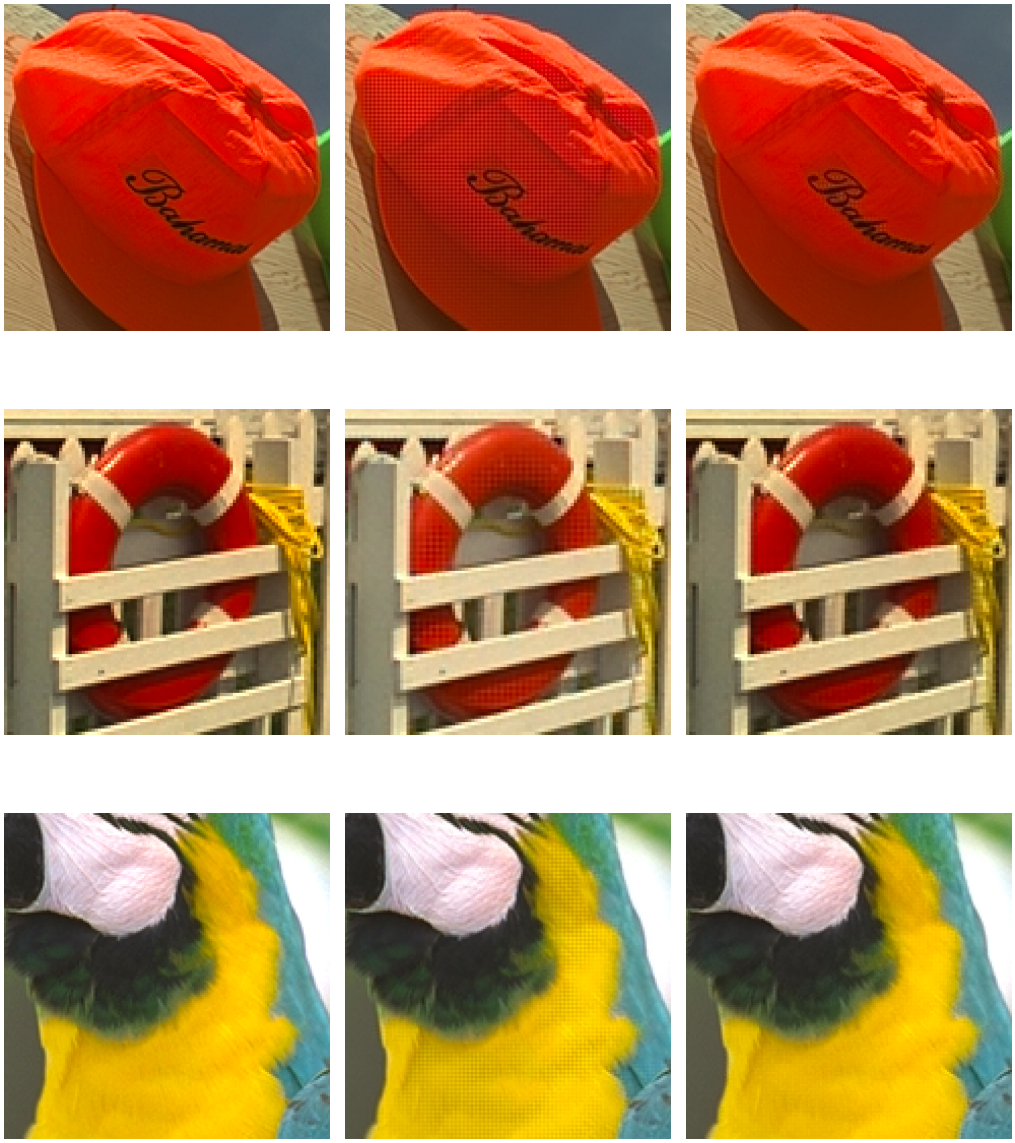


Figure 5.10: Effect of the introduction of the canonical basis of $\mathbb{R}^3$ in $\boldsymbol{\theta}_{DC}$. The left column contains the original images. The central column is the same central column of Fig. 5.2. The column on the right contains the images demosaiced via our implementation of the 8×8 standard configuration of the framework by Moghadam et al., which employes the canonical basis. All images were sensed with the Bayer CFA.

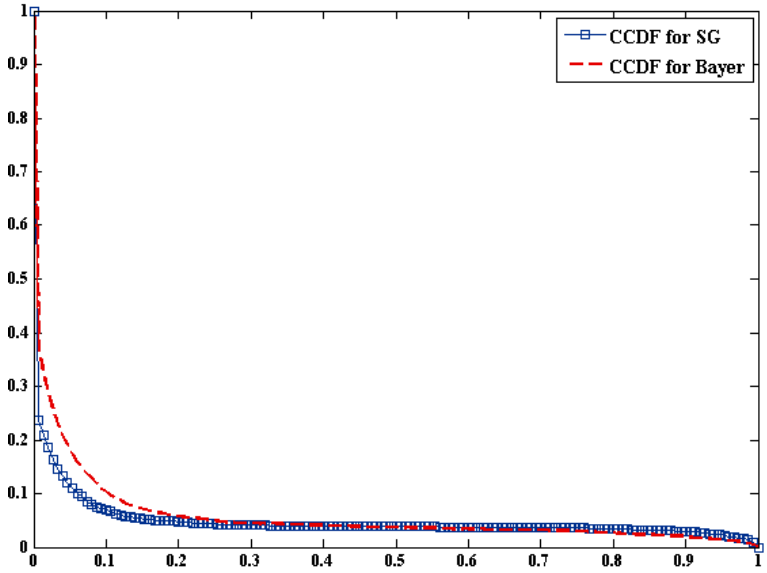


Figure 5.11: CCDFs obtained in our tests for the Bayer and SG CFAs.

in order to accelerate its CCDF decay, regardless of the SG CFA employed. Therefore, in our tests, we consider the SG CFA reported in (2.11), as this led us to the best results among the four SG CFAs in [9]. The CCDFs obtained with our simulations are represented in Fig. 5.11.

- In the 8×8 and 16×16 patch size configurations, we adopted an overlap of six and twelve pixels among adjacent patches, respectively. This choice was oriented toward a fair comparison of the two configurations, as it guaranteed that both in the 8×8 configuration and in the 16×16 one, each pixel was approximated based on the same number of estimates (i.e., eight).

- After each patch was demosaiced and the whole image reassembled, we applied the median filter described in [33].

Table 5.2 compares the results provided by Moghadam et al. with those we obtained in our tests. In the Bayer case, on average, our implementation achieves higher PSNRs than Moghadam et al., regardless of the patch size. We believe this is mainly due to our expedient concerning the introduction of the canonical basis. In the SG case instead, our implementation achieves, on average, lower PSNRs than Moghadam et al., again regardless of the patch size. We believe this is strictly connected to our SG CCDF, which happens to exibit the same decay of the Bayer one,

Table 5.2: Comparing the PSNRs of the Kodak images demosaiced via the standard configuration of the framework by Moghadam et al. Here, M denotes the implementation by Moghadam et al., while M* denotes our one. B and SG denote that the Bayer CFA and the SG one were employed at the sensing stage, respectively. Finally, 8 and 16, denote that the 8×8 standard configuration and the 16×16 one, were adopted respectively.

Img.	M-B-8	M*-B-8	M-B-16	M*-B-16	M-SG-8	M*-SG-8	M-SG-16	M*-SG-16
1	37.68	38.08	37.71	38.42	38.28	40.09	41.30	40.14
2	38.54	39.71	38.52	39.54	40.92	38.29	40.52	38.09
3	36.96	41.11	37.30	40.86	41.54	39.77	39.59	39.71
4	40.94	40.04	41.07	40.26	39.85	39.36	39.43	39.54
5	37.62	36.63	37.81	36.56	37.64	36.28	36.59	35.16
6	39.10	39.31	39.27	39.94	39.32	41.12	41.18	41.05
7	41.44	41.20	41.60	40.58	42.09	40.56	40.93	39.55
8	35.20	35.19	35.41	35.01	36.76	37.82	38.64	37.57
9	42.19	41.75	42.45	41.84	42.48	41.99	42.37	41.15
10	42.19	41.69	42.25	41.82	41.87	42.48	41.75	41.88
11	38.99	39.05	39.25	39.17	39.33	39.87	40.44	39.23
12	43.36	42.16	43.50	41.91	43.63	42.68	43.07	41.87
13	33.91	35.00	34.11	35.23	32.97	35.89	36.41	35.96
14	35.86	35.74	36.17	35.39	36.09	34.39	34.29	32.85
15	39.13	39.06	39.02	39.14	38.86	38.56	33.88	38.71
16	42.54	42.91	42.75	43.09	42.44	44.19	44.65	43.88
17	40.78	41.02	41.07	41.33	41.06	41.09	42.28	41.32
18	36.93	37.28	36.78	37.28	36.40	37.38	37.94	37.03
19	39.97	39.43	40.13	40.31	41.30	41.06	42.21	41.10
20	40.42	40.88	40.57	40.87	41.04	40.93	41.52	39.92
21	38.25	38.85	38.48	39.00	38.74	40.20	40.71	39.82
22	37.90	37.27	38.16	37.15	38.57	38.19	38.32	37.68
23	38.70	41.91	38.69	41.63	42.55	36.99	39.18	37.40
24	34.74	35.56	36.74	35.79	35.13	36.63	36.68	36.68
Avg.	38.89	39.20	39.03	39.26	39.54	39.41	39.75	39.05

or even worse. In Fig. 5.11 one can note that, when the Bayer CCDF decays to zero, its plot is slightly behind that of the SG CCDF. In our implementation, on average, the SG case exibits higher PSNRs than the Bayer one, when the 8×8 patch size is considered, while the converse happens with the 16×16 patch size. However, employing the SG CFA rather than the Bayer one, usually results in notably less color artifacts in the demosaiced images.

Results in Table 5.2 do not allow to establish if a given patch size is preferable over the other one. In the Bayer case, the 16×16 patch size allows to achieve slightly higher PSNRs, on average. Instead, in the SG case, the converse happens, as the 8×8 patch size seems to allow achieving higher PSNRs, on average. However, in both cases the average PSNR margin among the two patch size configurations is small. We observed the same fact in Table 5.1, with the results provided by Moghadam et al. This suggests, again, that adopting an adaptive patch division strategy mixing together the 8×8 and the 16×16 patch sizes may be a more effective choice. Interestingly, when the SG CFA is employed at the sensing stage, our implementation of the framework, which does not adopt any patch division strategy, provides demosaiced image whose visual quality is at least as high as that achieved by Moghadam et al. with their filtering configuration. This can be observed in Fig. 5.3-5.9. This result holds regardless of the patch size adopted in our implemention. Moreover, in some of the few cases where the filtering configuration by Moghadam et al. shows visible artifacts, our implementation does not. This can be observed in Fig. 5.12, which is organized exactly as Figures 5.3-5.9. We believe that the high visual quality provided by our implementation lies in the introduction of the canonical basis. Altough this expedient was studied for a particular problem involving only the Bayer CFA, in all our tests it showed to generally increase both the PSNR and the visual quality of the demosaiced images.

The results we obtained can be further improved, at least for how concerns the PSNR measure, increasing the overlap among adjacent patches. Clearly, an increase in the level of overlap directly translates into an increase in the number of patches to demosaic, and thus into an increasing of the amount of time required for demosaicing the image. In Table 5.3 we report the PSNRs obtained when applying an overlap of seven pixels in the 8×8 patch size configuration, and we compare them with the results of Table 5.2. Results in Table 5.3 confirm that increasing the level

Figure 5.12: Comparing the visual quality provided by different demosacing algorithm on a detail of Kodak image 8.

of overlap happens to be an effective choice, as it allows an important increase of the PSNR measure.

As we already claimed, we believe that an adaptive patch division strategy could be an effective choice in further improving the framework demosaicing quality. The results Moghadam et al. obtained with their filtering configuration of the framework suggest it. However, we believe that the success of their filtering configuration was mainly due to its implementation. More precisely, we believe that their results were mainly due to the procedure used to train the filter, rather than on the actual filter capability to detect diagonal edges. Moghadam et al. trained the filter on 5000 patches of 16×16 patch size which were better to be demosaiced directly, and 5000 patches of 16×16 patch size which were better to be further decomposed before being demosaiced. Obviously, each patch was classified via an automatic procedure, demosaicing each patch in both the ways, and then classifying the patch based on the PSNRs of the two reconstructions obtained. However this classification procedure does not guarantee that the patches in one set were characterized by diagonal details, while those in the other one were not. Therefore, the trained filter did not

Table 5.3: Comparing the PSNRs of the Kodak images demosaiced via our implementation of the framework by Moghadam et al. with two different levels of overlap. The 8×8 configuration is considered. Here, ov6 and ov7 denote the configuration employing an overlap of 6 and 7 pixels, respectively.

Img.	M*-B-8-ov6	M*-B-8-ov7	M*-SG-8-ov6	M*-SG-8-ov7
1	38.08	38.56	40.09	40.51
2	39.71	40.01	38.29	38.32
3	41.11	41.79	39.77	40.15
4	40.04	40.36	39.36	39.50
5	36.63	37.17	36.28	36.57
6	39.31	39.63	41.12	41.35
7	41.20	41.74	40.56	40.84
8	35.19	35.47	37.82	38.21
9	41.75	42.35	41.99	42.26
10	41.69	42.12	42.48	42.70
11	39.05	39.45	39.87	40.07
12	42.16	42.71	42.68	43.03
13	35.00	35.29	35.89	36.17
14	35.74	36.12	34.39	34.47
15	39.06	39.39	38.56	38.60
16	42.91	43.30	44.19	44.40
17	41.02	41.33	41.09	41.31
18	37.28	37.56	37.38	37.52
19	39.43	40.00	41.06	41.32
20	40.88	41.25	40.93	41.18
21	38.85	39.24	40.20	40.49
22	37.27	37.58	38.19	38.39
23	41.91	42.60	36.99	37.39
24	35.56	35.79	36.63	36.77
Avg.	39.20	39.62	39.41	39.65

necessary had diagonal detection capabilities. Nevertheless, the results provided by Moghadam et al. suggest that the trained filter predicted very well if a patch was better to be demosaiced directly, or after a further decomposition. This is not a contraddiction, as the filter was trained for this scope, and moreover it was trained on patches from the Kodak dataset, the same dataset whose images were later submitted to the filter during the demosaicing process. This reasoning convinced us that a different patch division strategy should be developed. That is why we decided not to develop the filtering configuration proposed by Moghadam et al.

We conclude this section by noting that quite all the demosaicing algorithms proposed in the litterature have been developed aiming at achieving high quality results on the Kodak dataset. However the results a demosaicing algorithm exibits on the Kodak dataset do not necessary reflect the actual behaviour of the algorithm on images captured by a digital camera. The reasons are mainly two. First, Kodak images were not acquired via a digital color camera. Instead, Kodak images where acquired via a film camera, and then the camera film was scanned to obtain a 2048×3072 digitalized version of each image. Second, the Kodak dataset usually employed, whose images size is 512×768, was obtained as a subsampled version of the original Kodak dataset obtained via the scanning procedure. Recently, in [37] Andriani et al. proposed a new data set comprised of images acquired via a professional digital camera developed by ARRI [41]. Thanks to the particular architecture of the employed camera, for each image in the proposed dataset is provided exactly the data acquired by the camera sensor (which is equipped with a Bayer CFA), without any post-processing. Our implementation of the demosaicing algorithm by Moghadam et al. was tested on this new dataset too. Moreover, the resulting demosaiced images were compared with those obtained via *ADA-3 (ARRI Debayering Algorithm version 3)*, the demosaicing algorithm currently employed at ARRI. Cleary the comparison was performed visually, as the original red, green, and blue full-resolution channels of the images in the dataset are not available. Again, the algorithm by Moghadam et al. happened to show notably less visual artifacts, and moreover, to exibit noise removal properties. This is well illustrated in Fig. 5.13, where the demosaiced image obtained via ADA-3 shows some color artifacts when compared with that obtained via the framework by Moghadam et al.

Figure 5.13: Testing our implementation of the framework by Moghadam et al. on an image from the dataset by Andriani et al. [37]. Above is the full image demosaiced by our implementation of the framework, in the middle is a detail from the previous image, and below is the same detail when the ADA3 demosaicing algorithm is employed.

Chapter 6

Conclusions

In this thesis recent sparse-based demosaicing algorithms has been considered, with particular emphasis to the algorithm proposed by Moghadam et al. in [11]. In sparse-based demosaicing algorithms, a major role is played by the sparsifying dictionary. Beyond the theoretical recovery guarantees provided by Sparse Recovery Theory, which happen to be overpessimistic in practice, it has been empirically noticed that the quality of the reconstruction provided by a sparse-based demosaicing image mostly depends on the selected sparsifying dictionary. Training the dictionary directly on the image to demosaic, therefore online, proved to be an effective strategy. That is way sparse-based demosaicing algorithms usually employ this strategy. However, the effectiveness of this adaptive strategy does not come for free, as the online training is a time consuming procedure which may slow down the whole demosaicing process. The algorithm by Moghadam et al. avoids the computational burden an online learning calls for, by building offline a dictionary which esplicitly takes into account the spatial and color correlation typical of the natural images involved in the demosaicing process. The results provided by Moghadam et al., together with the results we obtained developing our implementation of the framework, showed the effectiveness of this approach. Beyond the PSNR measure, the algorithm by Moghadam et al. happens to visually outperform state of the art demosaicing algorithms.

Developing our implementation of the algorithm by Moghadam et al. led us to believe that the algorithm can be further enhanced. Our introduction of the canonical basis of $\mathbb{R}^3$, which shown to be very effective in removing some artifacts due

to the Bayer CFA, is a clear example of it. The introduction of the adaptive patch division strategy by Moghadam et al. was motivated by the 2D-DCT being not too much effective in representing patches with diagonal edges. Moghadam et al. suggested to adopt a larger patch size in order to spread the error produced by this lack in the 2D-DCT on a wider area. However, a different strategy may consider the replacement of the 2D-DCT with another spatial transform, such as some kind of wavelet. One may even consider to train a spatial transform on a proper data set of ad hoc built images. Moreover, it would be very interesting to investigate a training set allowing to learn the color frames once for all, for example by including a palette. Interestingly, or unfortunately, the choice of the spatial transform, the choice of the set of color frames, and also the choice of the CFA to employ at the sensing stage, are all interellated, and this makes the enhancement of the algorithm a hard challenge.

Bibliography

[1] X. Li, B. K. Gunturk, and L. Zhang, "Image demosaicing: a systematic survey," in *Proc. SPIE-IS&T Electronic Imaging*, *Visual Communications and Image Processing*, vol. 6822, no. 1, Jan. 2008.

[2] P. M. Hubel, J. Liu, R. J. Guttosch, "Spatial frequency response of color image sensors: Bayer color filters and Foveon X3," *Technical report ID 6502*, *Foveon*, San Antonio, Texas, Mar. 2002.

[3] B. E. Bayer, "Color imaging array," *U.S. Patent* 3 971 065, July 1976.

[4] M. Elad, "Sparse and redundant representations: from theory to applications in signal and image processing," *Springer*, 2010.

[5] D. Alleyson, S. Süsstrunk, J. Hérault, "Estimating luminance and opponent chromatic signals in the Fourier domain," in *Proc. IS&T/SID 10th Color Imaging Conference*, vol. 10, pp. 331-336, Nov. 2002.

[6] E. Dubois, "Frequency-domain methods for demosaicking of Bayer-sampled color images," *IEEE Signal Processing Letters*, vol. 12, no. 12, pp. 847-850, Dec. 2005.

[7] B. Leung, G. Jeon, E. Dubois, "Least-squares luma-chroma demultiplexing algorithm for Bayer demosaicking," *IEEE Trans. on Image Processing*, vol. 20, no. 7, pp. 1885-1894, July 2011.

[8] G. Jeon, E. Dubois, "Demosaicking of noisy Bayer-sampled color images with least-square luma-chroma demoultiplexing and noise level estimation," *IEEE Trans. on Image Processing*, vol. 22, no. 1, pp. 146-156, Jan. 2013.

[9] K. Hirakawa and P. J. Wolfe, "Saptio-spectral color filter array design for optimal image recovery," *IEEE Trans. on Image Processing*, vol. 17, no. 10, pp. 1876-1890, Oct. 2008.

[10] D. L. Donoho, M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via l_1 minimization," in *Proc. of the National Academy of Sciences*, vol. 100, no. 5, pp. 2197-2002, 2003.

[11] A. A. Moghadam, M. Aghagolzadeh, M. Kumar, H. Radha, "Compressive framework for demosaicing of natural images," *IEEE Trans. on Image Processing*, vol. 22, no. 6, June 2013.

[12] A. A. Moghadam, M. Aghagolzadeh, H. Radha, M. Kumar, "Compressive demosaicing," in *Proc. IEEE Int. Workshop Multimedia Signal Processing*, pp. 105-110, Oct. 2010.

[13] M. Aghagolzadeh, A. A. Moghadam, H. Radha, M. Kumar, "Bayer and panchromatic color filter array demosaicing by sparse recovery," *SPIE*, *Digital Photography*, vol.7876, pp. 787603-1-787603-11, Jan. 2011.

[14] A. A. Moghadam, M. Aghagolzadeh, H. Radha, M. Kumar, "Incoherent color frames for compressive demosaicing," in *Proc. IEEE Acoustic, Speech Signal Processing*, pp. 5984-5987, May 2011.

[15] J. Mairal, M. Elad, G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Processing*, vol. 17, no. 1, pp. 53-69, Jan 2008.

[16] M. Aharon, M.Elad, A. M. Bruckstein, "The K-SVD: an algorithm for designing of overcomplete dictionaries for sparse representations," *IEEE Trans. on Image Processing*, vol. 54, no. 11, pp. 4311-4322, Nov. 2006.

[17] M. Elad, M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. on Image Processing*, vol. 15, no. 12, pp. 3736-3745, Dec. 2006.

[18] J. Mairal, F. Bach, J. Ponce, G. Sapiro, A. Zisserman, "Non-local sparse models for image restoration," in *Proc. IEEE 12th Int. Conf. Computer Vision*, pp. 2272-2279, Sep./Oct. 2009.

[19] M. Elad, "Optimized projections for compressed sensing," *IEEE Trans. on Signal Processing*, vol. 5, no. 12, pp. 5695-5702, Dec. 2007.

[20] S. Mallat, Z. Zhang, "Matching pursuit in a time-frequency dictionary," *IEEE Trans. on Signal Processing*, vol 41, no. 12, pp. 3397-3415, Dec. 1993.

[21] Y. C. Pati, R. Rezaiifar, P. S. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proc. of the 27th Annual Asilomar Conference on Signals, Systems, and Computers*, Nov. 1993.

[22] S. S. Chen, D. L. Donoho, M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing*, vol. 43, no. 1, pp. 129-159, 2001.

[23] B. Efron, T. Hastie, I. M. Johnstone, R. Tibshirani, "Least angle regression," *The Annals of Statistics*, vol. 32, no. 2, pp. 407-499, 2004.

[24] D. L. Donoho, "Compressed Sensing," *IEEE Trans. on Information Theory*, vol. 54, no. 6, pp. 1289-1306, April 2006.

[25] E. Candès, J. Romberg, T. Tao, "Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. on Information Theory*, vol. 52, no. 2, pp. 489-509, Feb. 2006.

[26] E. Candès, T. Tao, "Near optimal signal recovery from random projection: universal encoding strategies?," *IEEE Trans. on Information Theory*, vol. 52, no. 12, pp. 5406-5425, Dec. 2006.

[27] E. Candès, J. Romberg, T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Communications on Pure and Applied Mathematics*, vol. 59, no. 8, pp. 1207-1223, Aug. 2006.

[28] E. Candès, T. Tao, "Decoding by linear programming," *IEEE Trans. on Information Theory*, vol. 51, no. 12, 4203-4215, Dec. 2005.

[29] R. Baraniuk, M. Davenport, R. DeVore, M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, no. 3, pp. 253-263, Dec. 2008.

[30] A. Cohen, W. Dahmen, R. DeVore, "Compressed sensing and best k-term approximation," *Journal of the American Mathematical Society*, vol. 22, no. 1, pp. 211-231, Jan. 2009.

[31] Y. C. Eldar, G. Kutyniok, "Compressed sensing: theory and applications," *Cambridge University Press*, 2012.

[32] M. Duarte, M. Davenport, D. Takhar, J. Laska, T. Sun, K. Kelly, R. Baraniuk, "Single-pixel imaging via compressive sampling," *IEEE Signal Processing Magazine*, col. 25, no. 2, pp. 83-91, March 2008.

[33] K. Hirakawa, T. W. Parks, "Adaptive homogeneity-directed demosaicing algorithm," *IEEE Trans. on Image Processing*, vol. 14, no. 3, pp. 360-369, Mar. 2005.

[34] L. Zhang, X. Wu, "Color demosaicking via directional linear minimun square-error estimation," *IEEE Trans. Image Processing*, vol. 14, no.12, pp. 2167-2178, Dec. 2005.

[35] D. Paliy, V. Katkovnik, R. Bilcu, S. Alenius, 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, Oct. 2007.

[36] M. Fornasier, "Theoretical foundations and numerical methods for sparse recovery," *De Gruyter*, *Radon Series on Computational and Applied Mathematics 9*, July 2010.

[37] S. Andriani, H. Brendel, T. Seybold, J. Goldstone, "Beyond the KODAK image set: a new reference set of color image sequences," in *Proc. of the IEEE International Conference on Image Processing*, Melbourne, Australia, Sep. 2013.

[38] D. Menon, S. Andriani, G. Calvagno, "Demosaicing with directional filtering and a posteriori decision," *IEEE Trans. on Image Processing*, vol.16, no. 1, pp. 132-141, Jan. 2007.

[39] Private communication with the authors of [11].

[40] J. Mairal, F. Back, J. Ponce, G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. of the 26th International Annual Conference on Machine Learning*, Montreal, Canada, 2009.

[41] ARRI Group: <*http://www.arri.com*/>.

[42] KODAK dataset: <*http://www.cipr.rpi.edu/resource/stills/kodak.html*>.