A compressive framework for demosaicing of natural images

Abdolreza Abdolhosseini Moghadam[1], Mohammad Aghagolzadeh, Mrityunjay Kumar and Hayder Radha, *IEEE Fellow*.

***Abstract*— Typical consumer digital cameras sense only one out of three color components per image pixel. The problem of demosaicing deals with interpolating those missing color components. In this paper, we present Compressive Demosaicing (CD), a framework for demosaicing natural images based on the theory of Compressed Sensing (CS). Given sensed samples of an image, CD employs a CS solver to find the sparse representation of that image under a fixed sparsifying dictionary Ψ. As opposed to state of the art CS-based demosaicing approaches, we consider a clear distinction between the inter channel (color) and inter pixel correlations of natural images. Utilizing some well-known facts about the human visual system, those two types of correlations are utilized in a non-separable format to construct the sparsifying transform Ψ. Our simulation results verify that CD performs better (both visually and in terms of PSNR) than leading demosaicing approaches when applied to the majority of standard test images.**

***Index Terms*—image demosaicing, sparse coding, compressed sensing, compressive demosaicing.**

I. Introduction

In order to reduce the cost and size of consumer digital cameras, typically, only one photo sensor is utilized per pixel; and this sensor must capture the intensity of a single color of that particular pixel [1] [2]. Thus, color filters are placed over the photo-sensors to filter the captured light. Each color filter, which is associated with a single pixel and a corresponding photo sensor of that pixel, passes a specific narrow range of color wavelengths centered around a particular desired color (e.g., red, green or blue). The collection of all color filters utilized in a digital camera forms a

[1] This work was supported in part by Eastman Kodak company and National Science Foundation grant.
A. A. Moghadam, M. Aghagolzadeh and H. Radha are with the department of Electrical and Computer Engineering, Michigan State University, East Lansing, MI 48824, U.S.A. (email: abdolhos@msu.edu; aghagol1@msu.edu; radha@msu.edu). M. Kumar is with Eastman Kodak company, 1999 Lake Avenue, Rochester, 14650 NY (email: mrityunjay.kumar@kodak.com).

Color Filter Array (CFA). Therefore, the captured image is a mosaic of single-color pixels each with a color dictated by the CFA color pattern. The CFA-based camera architecture and the captured mosaic images ("CFA images") made it necessary to develop *demosaicing* algorithms that recover the original three color channels for each image pixel. Due to its impact on millions of consumer cameras, the area of demosaicing has received a great deal of attention over the past decade [3][16]. Several recent papers on image demosaicing provide an excellent overview of leading approaches and their classification (e.g. spatial versus frequency domains) [3][6].

It can be shown that the problem of demosaicing corresponds to an under-determined system of linear equations $y = \mathbf{\Phi}x$, where y is the (vector of) CFA samples sensed by the camera, $\mathbf{\Phi}$ represents the camera CFA, and the unknown vector x represents the image in full colors. The theory of Compressed Sensing (CS) [17][18] has proved that the solution x of an under-determined system of linear equations $y = \mathbf{\Phi}x$ is unique and can be found tractably, if (a) the solution vector x has a sufficiently sparse (or compressible) representation ξ under some dictionary $\mathbf{\Psi}$: $x = \mathbf{\Psi}\xi$ and (b) the matrix $\mathbf{\Phi\Psi}$, which in the CS literature referred as the projection matrix, adheres to certain conditions [17][18]. Hence a fundamental design issue for solving the demosaicing problem under a compressed sensing framework is the choice of the sparsifying dictionary $\mathbf{\Psi}$ for a given CFA matrix $\mathbf{\Phi}$, such that the two aforementioned conditions are met simultaneously.

Few novel attempts have been made to solve the problem of demosaicing based on solutions developed under the area of compressed sensing [32][37][36][15][34][38]. Despite the relative success of these approaches, each of them has its own limitations. For instance, the method in [38] (in its best profile) requires to "learn", in real-time, the sparsifying dictionary $\mathbf{\Psi}$ for a group of similar patches in the image, which might be prohibitively expensive in terms of computational complexity. For another example, many of the those approaches (e.g. [15][38][34]) are designed solely for the Bayer CFA. Although the Bayer CFA is the most common type of color filter arrays, this limits applicability of the aforementioned methods on other types of CFAs. Furthermore, as we show in this paper, Bayer CFA inherently introduces certain types of artifacts in the demosaiced images, which lowers the visual qualities.

With more emphasis on visual qualities of the demosaiced images, in this paper we present Compressive Demosaicing (CD), a CS-based framework for solving the problem of demosaicing. Similar to other CS based approaches, the proposed CD framework has three stages: (1) it divides CFA samples into a number of patches then (2) an under-determined system of linear equations for each patch would be solved by employing a CS solver and

finally (3) some post processing on the reconstructed image will be performed. However, unlike most of other CS-based approaches [15][16], we make no assumption about the utilized CFA in the sensing process in our design. Consequently, CD can work with any type of CFA. Arguably, what distinguishes CD the most from other CS-based approaches is how spatial and color correlations are utilized for the demosaicing process. As opposed to other approaches which are inherently oblivious to the distinction between these two types of correlations, CD is mindful of them. In other words, we utilize different sparsifying dictionaries for each domain, i.e. $\boldsymbol{\psi}$ and $\boldsymbol{\Theta}$ for spatial and color domains respectively. The final sparsifying dictionary $\boldsymbol{\Psi}$ will be formed by fusing these two dictionaries together. We show that employing only one frame for $\boldsymbol{\Theta}$ (corresponding to the separable formulation$\boldsymbol{\Theta} \otimes \boldsymbol{\psi}$**)** is very constraining and usually does not lead to a "good" quality sparsifying dictionary. This is due to the fact that the existence of color correlation in the pixel domain has different implications on different transform coefficients of the original image in the $\boldsymbol{\psi}$ domain. Thus, we extend the scope of $\boldsymbol{\Theta}$ from a single frame to a series of frames $\{\boldsymbol{\Theta}_i\}_i$**,** which we refer to as Incoherent Color Frames (ICF). Each of these frames $\boldsymbol{\Theta}_i$ captures the interaction of color correlation on a specific range of transform coefficients in the $\boldsymbol{\psi}$ domain. By incorporating some known facts about the human visual system during fusion of $\{\boldsymbol{\Theta}_i\}_i$ and $\boldsymbol{\psi}$ into the sparsifying dictionary $\boldsymbol{\Psi}$, one can achieve state of the art demosaicing even by applying non-adaptive sparsifying dictionaries. These concepts shall be explained with much more details in the following sections.

This paper is structured as follows: in section II, we present an overview of the Compressive Demosaicing framework and then investigate each stage of the framework with details. Section III is devoted to simulation results and finally section IV concludes this paper.

A. An overview of the paper contributions

The proposed Compressive Demosaicing framework is based on utilizing a fixed sparsifying dictionary $\boldsymbol{\Psi}$ that can work with any given CFA matrix $\boldsymbol{\Phi}$. This represents one departure from recent CS-based approaches, which rely solely on the Bayer CFA and adaptive learned dictionaries to solve the demosaicing problem. The proposed compressive demosaicing framework is characterized by the following contributions.

1) Explicit distinction between spatial and color sparsifying dictionaries

A viable CS-based demosaicing solution needs to consider the sparsification of the color image in both (a) the spatial (pixel) domain through an inter-pixel decorrelation transform and (b) the 3D space of the color channels through

inter-color channel (or simply "inter-channel") decorrelation. This sparsification process in both domains can be achieved in a separable manner or by a non-separable framework. Under CD, we develop both separable and non-separable frameworks. For now, assume that we can employ two separable sparsification dictionaries: $\boldsymbol{\psi}$ for the spatial (pixel) domain and $\boldsymbol{\Theta}$ for the 3D color space. Hence, under a separable CD framework, we model the demosaicing problem using an overall sparsifying dictionary $\boldsymbol{\Psi} = \boldsymbol{\Theta} \otimes \boldsymbol{\psi}$, where $\otimes$ is the tensor Kronecker product. Hence, under CD, we have $y = \boldsymbol{\Phi}\boldsymbol{\Psi}\xi = \boldsymbol{\Phi}(\boldsymbol{\Theta} \otimes \boldsymbol{\psi})\xi$, where the pixel domain and color domain sparsification dictionaries are explicitly distinct from each other. It should be clear that for the pixel sparsification dictionary $\boldsymbol{\psi}$, a wide range of popular transforms such as DCT or wavelets, or even over-complete spatial dictionaries could be employed. Hence, a key question under such framework is what dictionary $\boldsymbol{\Theta}$ one should employ for the color channels.

2) Incoherent Color Frames (ICF) for compressive demosaicing

Our initial work on compressive demosaicing [32] was based on designing a single color dictionary $\boldsymbol{\Theta}$ by projecting the traditional three color channels onto an Equiangular Tight Frame (ETF) [7][31]. This opened the door to a new paradigm for solving the demosaicing problem. In a nutshell, applying an ETF within the 3D color space enables further sparsification, and hence enhances the probability of a CS solver to find the original three-color signal. Nevertheless, and despite being maximally incoherent, an ETF based approach for demosaicing suffers from a major shortcoming where only a single non-trivial ETF frame is known to exist in 3D. This single non-trivial ETF in 3D has only six basis vectors; and hence the level of sparsity that can be achieved is rather limited. Consequently, there is a clear need for developing a more general framework than ETF for representing and sparsifying color channels in their inherently 3D space. Such framework should have the following objective: It should exploit the notion of projecting the three traditional color channels onto a sparsifying color dictionary using an optimal number of basis vectors while considering key features of the color space and its relationship with the spatial frequency domain. In this paper, we introduce a new frame design that we refer to as Incoherent Color Frames (ICF) which addresses the aforementioned objective.

3) Non-separable ICF for compressive demosaicing

One option for designing a color dictionary is to have an ICF that is a function of the pixel-domain spatial-frequency. Such strategy can be expressed using a more compact compressive-demosaicing framework: $y = \Phi S(\boldsymbol{\psi}, \{\boldsymbol{\Theta}_i\}_i)\xi$,

where in this case, the non-separable incoherent color frames $\{\boldsymbol{\Theta}_i\}_i$ are a function of the particular pixel-domain spatial frequency i and $S(\boldsymbol{\psi}, \{\boldsymbol{\Theta}_i\}_i)$ is a function that "mixes" $\{\boldsymbol{\Theta}_i\}$ with the spatial dictionary $\boldsymbol{\psi}$. The non-separable ICF approach takes into consideration key features of the color space, its relationship with the spatial frequency domain and important attributes of the human visual system (e.g., its low-sensitivity to the high frequency color components). We show that this non-separable strategy can outperform its separable counterpart significantly.

4) Construction algorithms for ICF frames

It is important to note that employing well known "optimization" methods for constructing tight frames or for achieving minimum coherence projection matrices [33] are not suitable (not even applicable) in the context of the proposed ICF frames since a set of ICF constraints (outlined later) lead to a highly non-convex problem and any alternating projection algorithm may diverge after a few iterations. Nevertheless, we show that finding optimum ICFs translates into finding "good" sparsifying dictionaries (or transforms) for triplets of $(\hat{R}_i, \hat{G}_i, \hat{B}_i)$ where $\hat{R}_i$, $\hat{G}_i$ and $\hat{B}_i$ are respectively, the i-th transform coefficient of red, green and blue color planes of the original image in the employed spatial dictionary $\boldsymbol{\psi}$. Therefore, one can utilize available learning methods and optimization techniques to find and construct ICFs. In our design, we also consider attributes of the human vision system to achieve good sparsifying results. Our simulation results show that the proposed ICF design provides visibly improved quality for demosaiced images when compared to leading approaches.

II. The framework of Compressive Demosaicing

In this section, we present the framework of "Compressive Demosaicing" (CD), a framework for demosaicing of natural images based on the theory of compressed sensing. Similar to other CS-based demosaicing algorithms [15][32][38] CD is comprised of the following three major stages:

- *Dividing the image into patches*: The algorithm divides sensed CFA samples y into groups $\{y^{(j)}\}$ (where $\bigcup_j y^{(j)} = y$) and each group $y^{(j)}$ would undergo stages 2-3. Note that, these groups of samples correspond to patches of the original image which would be demosaiced. To avoid the blocking effects, usually some overlaps among these blocks are assumed in practice.
- *Solving a system of linear equations*: The algorithm selects a sparsifying transform (dictionary) $\boldsymbol{\Psi}^{(j)}$ for the input group of samples $y^{(j)}$. This dictionary can range from a fixed dictionary for all blocks of the image $\forall j: \boldsymbol{\Psi}^{(j)} = \boldsymbol{\Psi}$ to

an adaptive one which we choose based on the image patch. Then, the algorithm forms an under-determined system of linear equations $y^{(j)} = \mathbf{\Phi}^{(j)}\mathbf{\Psi}^{(j)}\xi^{(j)}$, where $y^{(j)}$ is the input group of samples, $\mathbf{\Phi}^{(j)}$ is a function of the camera and the part of the CFA affecting the sample values $y^{(j)}$. The unknown vector $\xi^{(j)}$ is essentially the representation of the patch of the original image (corresponding to samples $y^{(j)}$) in the $\mathbf{\Psi}^{(j)}$ domain. The representation $\xi^{(j)}$ will be estimated by a CS decoder to $\hat{\xi}^{(j)}$. Finally, that patch in full color would be approximated by $\mathbf{\Psi}^{(j)}\hat{\xi}^{(j)}$.

- *Post processing*: Depending on the amount of overlaps, a number of estimates for the colors of a pixel would be available. The last stage of CD finds the best approximation of the colors of the image from the available estimates $\{\mathbf{\Psi}^{(j)}\hat{\xi}^{(j)}\}$. Some classical post-processing such as median filtering might be also required to remove certain types of color artifacts.

Although all CS-based demosaicing methods follow the same high level algorithm (outlined above), their performances and qualities of recovered images could vary significantly. For instance, the authors in [15] used their own modified version of the greedy Orthogonal Matching Pursuit (OMP) [22] for CS-solver in stage two, while Basis Pursuit (BP) [23][29] was utilized in the proposed approach in [32]. Although the choice of the CS-solver affects the computational complexity of an algorithm, the quality of demosaiced images mostly depends on the quality of the selected sparsifying dictionary $\mathbf{\Psi}^{(.)}$.

A. Notations

Before formulating the problem of demosaicing in the following, let us present the notations used throughout this paper. For a natural number $q \in \mathbb{N}$, define $[q] \coloneqq \{1,2,\dots,q\}$. The cardinality of a set s is denoted by $Card\{s\}$. The l-th entry of the vector ξ is denoted by ξ_l. Matrices (or 2D images) would be represented by bold size fonts (e.g. $\mathbf{\Phi}$) while vectors and scalars shall be represented by normal font size (e.g. x). We often use MATLAB notation for identifying submatrices of a matrix. For instance, for $\boldsymbol{A} \in \mathbb{R}^{n_1 \times n_2}$, $s_1 \in [n_1]$ and $s_2 \in [n_2]$, by $\boldsymbol{A}_{s_1,s_2}$ we mean the submatrix of $\boldsymbol{A}$ restricted to rows of s_1 and columns of s_2. Similarly we use the notion of ":" for instance $\boldsymbol{A}_{s_1,:}$ (or $\boldsymbol{A}_{:,s_2}$) to keep only rows (columns) indexed by s_1 (s_2). The transpose of $\boldsymbol{A}$ is denoted by $\boldsymbol{A}^T$. To simplify equations and without loss of generality, we often deal with the *vectorized* form of images in this paper. By vectorized form of an image (or a 2D matrix) $\boldsymbol{A}$, we mean that we stack the columns of that image on top of each others. For instance,

the vectorized form of a $n_1 \times n_2$ pixel image $\boldsymbol{A} \in \mathbb{R}^{n_1 \times n_2}$ is a $n_1 n_2 \times 1$ vector $A := Vec(\boldsymbol{A})$ where $A_l = \boldsymbol{A}_{i,j}$ holds true for $l = (j-1)n_1 + i$.

The expected value of a random variable x will be denoted by $E(x)$. Finally $\otimes$, $\odot$ and $*$ respectively denote Kronecker tensor product operator, the Hadamard (point wise) product and 2D-convolution.

B. The problem formulation

In this sub-section, we model the sensing process and link the demosaicing problem to the CS problem [17][18]. Consider $\boldsymbol{X}$, an $n_1 \times n_2$ pixel color image, composed of three primary color planes red $\boldsymbol{R} \in \mathbb{R}^{n_1 \times n_2}$, green $\boldsymbol{G} \in \mathbb{R}^{n_1 \times n_2}$ and blue $\boldsymbol{B} \in \mathbb{R}^{n_1 \times n_2}$. Then, the color of the l-th pixel (in the vectorized form) is in the form of (R_i, G_i, B_i) in the RGB color system where $i \in [N]$ and $N = n_1 n_2$ is the total number of pixels in $\boldsymbol{X}$. Furthermore, suppose that the deployed CFA, at the i-th pixel passes red, green and blue wavelengths by factors of α_i, β_i and γ_i respectively where[2] $\|[\alpha_i \ \beta_i \ \gamma_i]\|_1 = \alpha_i + \beta_i + \gamma_i = 1$ [5]. Here the value of one indicates a complete pass (no attenuation) and the value of zero represents total blockage (no passing). Hence y_i, the i-th sensed sample is simply:

$$\forall i \in [N]: y_i = \alpha_i R_i + \beta_i G_i + \gamma_i B_i \tag{1}$$

Note that, this formulation is valid only for "single color per pixel sensing". Nevertheless in this paper, we adhere only to this case i.e. ordinary (and not CS) cameras[3].

Let us define $N \times N$ diagonal matrices $\overline{\boldsymbol{\alpha}}$, $\overline{\boldsymbol{\beta}}$ and $\overline{\boldsymbol{\gamma}}$ as:

$$\overline{\boldsymbol{\alpha}}_{i,i} = \alpha_i, \ \ \overline{\boldsymbol{\beta}}_{i,i} = \beta_i, \ \ \overline{\boldsymbol{\gamma}}_{i,i} = \gamma_i \tag{2}$$

and zero otherwise. Also define the *CFA matrix* $\boldsymbol{\Phi} \in \mathbb{R}^{N \times 3N}$ by:

$$\boldsymbol{\Phi} = [\overline{\boldsymbol{\alpha}} \quad \overline{\boldsymbol{\beta}} \quad \overline{\boldsymbol{\gamma}}] \tag{3}$$

Then we can extend (1) to y, the vector of all CFA samples through:

$$y = \boldsymbol{\Phi} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4}$$

[2] This constraint considers the fact that sensors have an upper limit of sensing of light intensities.

[3] The formulation can be easily extended to the case of single pixel cameras [28]. In this case, the passing parameters would be matrices (and not vectors). More specifically, let $\boldsymbol{\alpha}_{i,j}$, $\boldsymbol{\beta}_{i,j}$ and $\boldsymbol{\gamma}_{i,j}$ represent how much intensities of red, green and blue wavelengths respectively at pixel index j (in the vectorized form) have been attenuated in the sensing of the i-th sample. Then, (1) turns into:

$$y_i = \sum_{j=1}^{N} (\boldsymbol{\alpha}_{i,j} R_j + \boldsymbol{\beta}_{i,j} G_j + \boldsymbol{\gamma}_{i,j} B_j)$$

Consequently, the matrices $\overline{\boldsymbol{\alpha}}$, $\overline{\boldsymbol{\beta}}$ and $\overline{\boldsymbol{\gamma}}$ in (2) will not be diagonal in this case.

The equation (4) can explain the problem of demosaicing in a compact form; that is: given sensed samples y and the CFA (matrix) $\mathbf{\Phi}$, estimate the primary color planes of the underlying image (i.e. vector of $[R^T\ G^T\ B^T]^T$). Since the number of samples ($Card\{y\} = N$) is less than the size of the unknown vector $[R^T\ G^T\ B^T]^T$ (which is $3N$), hence the problem of (4) is an under-determined system of linear equations. From basic linear algebra, it has been known that such a system of equations has an infinite number of solutions and thus any algebraic approach to solve the demosaicing problem sounds inconceivable. However, the recently developed theory of Compressed Sensing (CS) [17][18][19] has proved that an under-determined system of linear equations under certain conditions (outlined in those references) has a unique solution which can be found tractably, if that solution is sufficiently sparse (mostly zero) or compressible. So one might think that, the samples and the CFA matrix could be input to a CS decoder to find the RGB color planes of the image. However, this approach more likely will fail, since the most important assumption of the CS framework, i.e. sparsity of the solution vector is not satisfied in this approach. More specifically, usually for natural images, RGB color planes are very similar to (the luminance of) the image itself and hence neither of them is sparse. On the other hand, it has been widely known that natural color images have strong inter-pixel and inter-channel (color) correlations. High inter-pixel and inter-color correlations translate to existence of some sparsifying frames where these natural images are sparse with respect to them. This suggests that we demosaic color images in transform domains other than RGB pixel domain.

Assume $[R^T\ G^T\ B^T]^T$ has a sparse representation $\xi \in \mathbb{R}^{qN}$ in $\mathbf{\Psi} \in \mathbb{R}^{3N\times qN}$ domain, where $q/3 \geq 1$ determines the redundancy factor of such transform. This can be expressed by:

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \mathbf{\Psi}\xi \tag{5}$$

Then, merging (4) and (5) yields:

$$y = \mathbf{\Phi\Psi}\xi \tag{6}$$

If the transform $\mathbf{\Psi}$ captures inter-pixel and inter-channel correlations effectively, then ξ would be sufficiently sparse or compressible:

$$k = \|\xi\|_0 \coloneqq Card\{i: \xi_i \neq 0\} \ll 3N \tag{7}$$

Consequently, generic CS-decoders such as Lasso [24], greedy pursuit algorithms (e.g. OMP, Cosamp, Stomp, etc) [21][22][20] or Basis Pursuit (BP) [23], can estimate the unknown vector ξ from samples y by solving the following convex optimization problem:

$$\xi^* = \arg\min_{\hat{\xi}} \left\|\hat{\xi}\right\|_1 : y = \boldsymbol{\Phi}\boldsymbol{\Psi}\,\hat{\xi} \tag{8}$$

Having ξ^*, one can use (5) to estimate RGB color planes of the original image by $\boldsymbol{\Psi}\xi^*$.

Before introducing configurations which we adapted in our simulations, let us present the effect of deployed CFA on the quality of the demosaiced images in the next subsection.

C. The Color Filter Array

We begin presenting CD, by investigating the effect of CFA on quality of demosaiced images for two reasons: (a) CFA is the first physical component in a digital camera which affects the quality of demosaiced images and (b) it controls the information that samples capture. Hence, utilizing an improper CFA in the sensing stage can lead to demosaiced images with degraded qualities even if an optimal demosaicing method is deployed.

Arguably, Bayer CFA is the most common type of color filter arrays used in the digital cameras where its CFA matrix ($\boldsymbol{\Phi}_{\mathrm{B}}$) for a block of 2×2 is in the form of:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{9}$$

Recall that human eyes are more sensitive to green wavelength than red and blue. In the design of Bayer pattern, this fact is considered by allocating more green color filters. In fact, most of traditional demosaicing algorithms benefit such property of Bayer pattern by following a sequential process where first, the green color plane (with more available samples) is interpolated and then the other two difference (or ratio) color planes (e.g. R-G and B-G) would be interpolated [3]. On the other hand, a CS-based demosaicing algorithm (usually) cannot follow such sequential approach, since it recovers the sparse representation of RGB color planes of a block (in some sparsifying transform) simultaneously. One might think that Bayer CFA is the optimal design, even for CS-based demosaicing approaches since the method [38] with the highest reported PSNRs of demosaiced images on Kodak test set, is designed upon Bayer CFA. Broadly speaking, this method [38] attempts to find similar patches of the image and then finds an adaptive sparsifying transform for these similar patches. This task is achieved by exploiting the “pure” sampling

property of Bayer CFA where, each sample is from only one color plane (and not a mixture of different color planes). Despite the extensive exploitation of Bayer CFA in detecting similar patches of an image, it can be argued that, this method benefits mainly from non-local means filtering [39] and employing adaptive dictionary design in a joint sparsity model [38]. Furthermore, it has been proved in some other works (e.g. [5][41]) that panchromatic CFAs must be deployed to achieve better modulations in terms of separability (i.e. lower chance of aliasing) of color channels in the Fourier domain. This leaves us with an important question to answer, and that is: which CFA is optimum and what characteristics does it have? In the rest of this sub-section, we present our observations and also some guidelines in design of optimal CFAs for the CD framework. Indeed there does not exist a globally optimum CFA and it might be the case that, a CFA is optimum for a demosaicing algorithm and not for another one. The optimal CFA design for classical (non CS) demosaicing approaches have been studied extensively (e.g. in [5][41]) but not for recently developed CS-based demosaicing methods. It is of great importance to note that under a CS-based demosaicing approach, the optimality of a CFA $\boldsymbol{\Phi}$ depends on the employed sparsifying transform $\boldsymbol{\Psi}$, since the CS solver only operates on $\boldsymbol{P} = \boldsymbol{\Phi}\boldsymbol{\Psi}$ and is oblivious to $\boldsymbol{\Psi}$ and $\boldsymbol{\Phi}$. For now assume that there is an ordering function $\Xi: \mathbb{R}^{a\times b} \to \mathbb{R}^{+}$ for all possible projection matrices of size $a \times b$. That is, if $\Xi(\boldsymbol{A}) < \Xi(\boldsymbol{B})$ then $\boldsymbol{A}$ is preferred over $\boldsymbol{B}$ as a projection matrix for a CS decoder. For a given number of pixels N let us define the class of "admissible CFA matrices" $\mho$ as follows:

$$\mho = \{\boldsymbol{\Phi} \in \mathbb{R}^{N\times 3N}: \boldsymbol{\Phi}_{i,j} = 0, \boldsymbol{\Phi}_{i,i} + \boldsymbol{\Phi}_{i,N+i} + \boldsymbol{\Phi}_{i,2N+i} = 1\}$$

for all $i, j \in [N]$ and $j \neq i, N+i, 2N+i$. Note that this class of matrices corresponds to all valid CFAs which can be employed in the sensing stage. For a given ordering Ξ and sparsifying dictionary $\boldsymbol{\Psi}$, the problem of optimal design of CFA under a CS framework can be casted as:

$$\boldsymbol{\Phi}^{*} = \arg\min_{\boldsymbol{\Phi}} \Xi(\boldsymbol{\Phi}\boldsymbol{\Psi}) : \boldsymbol{\Phi} \in \mho \tag{10}$$

Note that (depending on Ξ) this optimization problem might not be convex. For now, let us focus on the ordering function $\Xi(.)$. Recall that a CS-decoder estimates the unique sparse/compressible solution ξ in (8) with the minimum error, if the effective projection matrix $\boldsymbol{P} = \boldsymbol{\Phi}\boldsymbol{\Psi}$ has a small value of "Restricted Isometry Constant" [18] or RIC[4]. Back to the CFA matrix of Bayer in (9), it can be clearly observed that there are exactly eight zero columns in $\boldsymbol{\Phi}_B$ indicating a high RIC for that matrix. This undesirable property can be propagated from the CFA matrix $\boldsymbol{\Phi}_B$ to the

[4] Broadly speaking, RIC of a matrix is a measure, indicating how much sub-matrices of that matrix deviate from an orthogonal projection.

effective projection matrix $\boldsymbol{P}_B = \boldsymbol{\Phi}_B \boldsymbol{\Psi}$. In words, the Bayer might not be the optimal CFA under the measure of RIC for Ξ. This prevision was in complete agreement with our simulation results where we have observed that the Bayer CFA does not lead to demosaiced images with highest PSNRs (see Figure 1 for an example). Further results, showing the effect of CFA on the quality of demosaicing are available in our website [44].

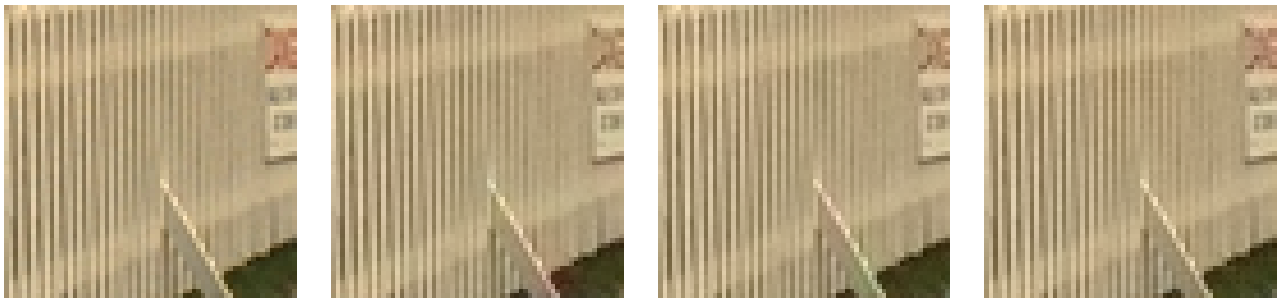


Figure 1 The effect of CFA on the quality of the demosaiced images. From left to right: (a) The original patch (b) random panchromatic CFA, PSNR=41.88db, (c) Bayer CFA, PSNR=41.24db, (d) Second generation CFA, PSNR=42.58db.

Despite the ability of RIC in predicting the behaviour of generic CS-solvers during the sparse recovery, it cannot be employed in Ξ since computing the RIC of a matrix is combinatorially hard and cannot be done in polynomial time. Thus the community proposed the alternative measure "mutual coherency" which can be computed very efficiently and yet it provides some (weak) bounds for RIC of a matrix [17][21][33]. The mutual coherency of a matrix $\boldsymbol{A} \in \mathbb{R}^{n_1 \times n_2}$ is denoted by $\mu(\boldsymbol{A})$ and is the largest magnitude normalized inner product between different columns of that matrix. This can be expressed in the form of:

$$\mu(\boldsymbol{A}) = \max_{i \neq j} \frac{|\boldsymbol{A}_{:,i}^T \boldsymbol{A}_{:,j}|}{\|\boldsymbol{A}_{:,i}\| \|\boldsymbol{A}_{:,j}\|} \tag{11}$$

A low measure of mutual coherency is highly desirable for the success of many CS-decoders belonging to the class of greedy pursuit algorithms (e.g. [20][21][22]). On the other hand, mutual coherency is a worst case measure and may not reflect the actual behaviour of a CS solver in practice. Thus, it has been suggested in several works (e.g. [33][36]) to use the distribution of the sequence $\left|\boldsymbol{A}_{:,i}^T \boldsymbol{A}_{:,j}^T\right| / (\|\boldsymbol{A}_{:,i}\| \|\boldsymbol{A}_{:,j}\|)$ as a more realistic measure for optimality of the matrix $\boldsymbol{A}$ as the projection matrix in a CS framework. For such measure, the authors in [36] proposed using the exceedance (or Complementary Cumulative Distribution) function:

$$F_c(\boldsymbol{A}, g) = \Pr\left(\frac{|\boldsymbol{A}_{:,i}^T \boldsymbol{A}_{:,j}|}{\|\boldsymbol{A}_{:,i}\| \|\boldsymbol{A}_{:,j}\|} \geq g \right) \tag{12}$$

where Pr denotes the probability function and g is a real valued number in the range of $[0,1]$. Thus for an optimal projection matrix $\boldsymbol{P}$, one expects that $F_c(\boldsymbol{P},.)$ decays fast. Figure 2 compares the corresponding exceedance functions of the effective projection matrices for three types of CFAs: Bayer $F_c(\boldsymbol{\Phi}_B\boldsymbol{\Psi},g)$, Second Generation (SG) [5] $F_c(\boldsymbol{\Phi}_S\boldsymbol{\Psi},g)$ and a random panchromatic CFA $F_c(\boldsymbol{\Phi}_R\boldsymbol{\Psi},g)$ where $\boldsymbol{\Psi}$ is our deployed sparsifying dictionary (to be presented in the following sections) for patches of size 8×8 pixels. As clearly illustrated, the exceedance function of a SG CFA decays fastest when compared with the other two CFAs. On the other hand, a Bayer CFA has the heaviest tail. Thus one might expect that under CD framework, a SG CFA is preferable over the other two CFAs. In Figure 1, we have demosaiced a patch from the famous "lighthouse" test image under CD framework with the aforementioned three CFAs. Note that the resulted PSNRs of demosaiced images are in complete agreement with our proposed measure. Unfortunately, the proposed exceedance function is not a valid ordering Ξ since $F_c(\boldsymbol{A},g)$ does not give us a non-negative scalar and instead it yields a continuous function over the range of $[0,1]$ ($F_c(\boldsymbol{A},g):\mathbb{R}^{a\times b}\times[0,1]\rightarrow[0,1]$). Designing a valid ordering from the exceedance function and solving (10) are subjects of our future work. Due to the availability and desirable properties of SG CFAs, in this paper we have adapted them in our simulations.

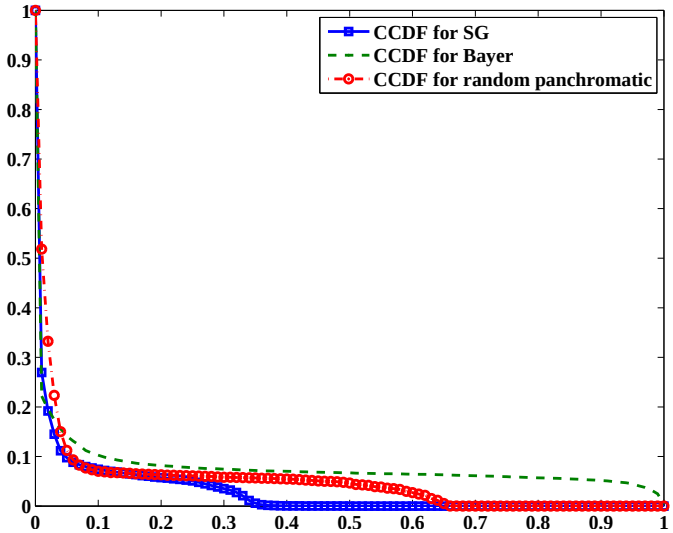


Figure 2. Complementary Cumulative Distribution Function (CCDF) for three different CFA types.

D. Dividing the image into patches

To employ CS in the problem of demosaicing, we have to demosaic blocks (as opposed to individual pixels) of images, due to several reasons, few of which we list here. First, in a CS framework, we traditionally sense $m>1$ linear samples from the same signal; however in digital cameras, we are sensing only one compressive sample ($m=1$) from three unknowns (the color components of a specific pixel). In other words, for each pixel we have a (seemingly independent) system of only one equation and three unknowns and these unknowns might not be sparse

in the RGB color coordinate and CS does not apply to this type of problem. Second, in practice, virtually all popular CS decoders have optimum operating regions [40], where outside these regions even if a theoretically sufficient number of samples are provided to the solver, there is no guarantee that the underlying CS decoder would find the solution. Usually, this optimal working region does not include extremely short length signals. Thus, if we attempt to recover the color components of each pixel individually, even if the pixel has a sparse representation in the RGB domain, there is no guarantee that the underlying CS decoder would find that solution. Therefore, a block-wise processing would be inevitable if CS is being utilized in the demosaicing framework. Assume that an N pixel image is divided into Q blocks. In this paper, we denote the pixel indices belonging to the i-th block by S_i. That is:

$$\forall i \in [Q]: S_i \subseteq [N]: \bigcup_i S_i = [N] \tag{13}$$

Then we have:

$$y^{(i)} = y_{S_i} = \alpha_{S_i} \odot R_{S_i} + \beta_{S_i} \odot G_{S_i} + \gamma_{S_i} \odot B_{S_i} \tag{14}$$

Optimal configuration of block sizes $Card\{S_i\}$ and also the algorithm which assigns pixel indices to these sets $\{S_i\}$, significantly improve the quality of demosaiced images. Similar to (virtually all) classical image processing algorithms, these optimal configurations are highly dependent on the domain where the demosaicing process takes place ($\boldsymbol{\Psi}$) and more importantly on the image itself. For instance and as reported in [15][38], a larger block size should be adapted for demosaicing of smooth and texture regions while a smaller block size leads to demosaiced patches with fewer artifacts in non-smooth regions. Another observation in case of CS-based demosaicing is that, as the size of the block increases, the estimation process becomes more stable. More specifically, a larger block size leads to demosaiced images with an acceptable level of error (neither very high nor very low quality and usually a color-washed patch) while choosing a smaller block size generates either a noisy patch or one without any visible artifacts. These effects are clearly illustrated in Figure 3.

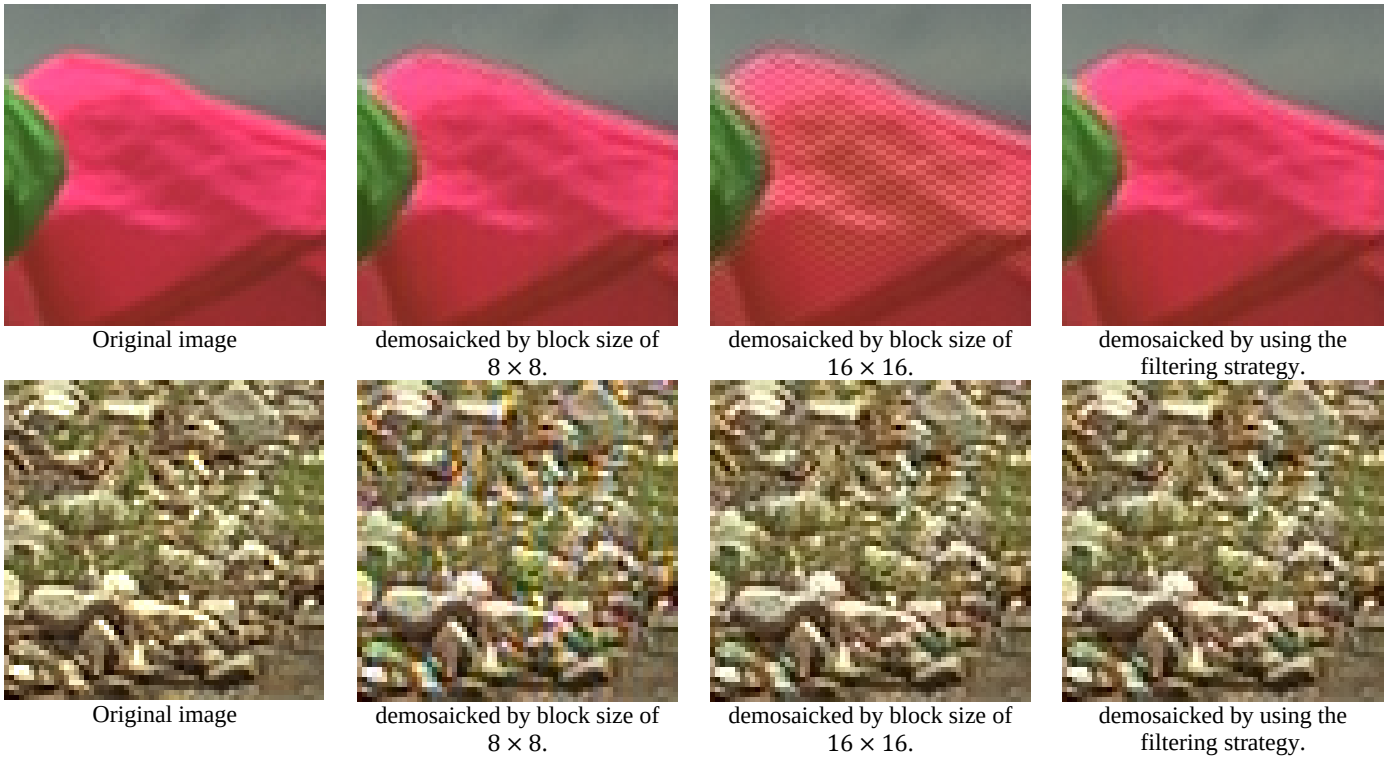


Figure 3 The effect of block size on the quality of the demosaiced images. Smaller block sizes are more suitable for singular regions (the first row), while larger block sizes should be selected for texture and smooth regions (the second row).

For the brevity, let us use the term "block division strategy" for the algorithm which selects the block sizes and assigns pixel indices to the blocks. Three notable block division policies have been proposed so far for a CS-based demosaicing scheme:

- Fixed block size: Similar to classical block processing algorithms, this strategy slides a window of fixed size ($\forall i,j: Card\{S_i\} = Card\{S_j\}$) over the image (e.g. in [15][32][36]).
- Using two sliding windows of different sizes: An image would be demosaiced using two block sizes (e.g. 8×8 and 16×16). Then two reconstructed images would be fused by a weighted average (e.g. the proposition of [15]).
- Similar blocks of the same size: In this strategy, the image would be demosaiced using a fixed block size. However similar blocks would be demosaiced using a joint sparsity model (see [38]).

Each of these three policies has its own advantages and disadvantages. For instance the first strategy is very simple and demands the least of computations at the cost of lower qualities of demosaiced images. The second strategy is computationally light and leads to improvement on the qualities of reconstructed images (compared to the first strategy). However this strategy implies two runs of the demosaicing algorithm. Furthermore, it does not benefit

good properties of different block sizes to the maximum extent possible[5]. Although the third strategy is utilized in a top performer (in terms of PSNRs) [38], it is the most complex one among the three policies and ignores the benefits of utilizing different block sizes.

In the remainder of this subsection, we present an efficient strategy for dividing the image into patches with proper block sizes that we used in our simulations.

1) Filtering

Here, we propose a low complexity strategy that does not demand two iterations of the demosaicing process. The first stage of such strategy is dividing the CFA samples into macro-blocks. Each macro-block would contain a number of blocks and all these blocks would have the same size. However, different macro-blocks might be demosaiced using different block sizes. Let us use Ω_i to denote the block indices belonging to $\bar{S}_i$, the i-th macro-block. Then we have:

$$\forall a, b \in \Omega_i : S_a \subseteq \bar{S}_i \, , Card\{S_a\} = Card\{S_b\} \tag{15}$$

In our simulations we form these macro-blocks by partitioning an image into patches of size 16×16. Indeed more effective algorithms can be used for forming macro-blocks (e.g. adaptive macro-blocks with varying sizes). However to keep the strategy complexity low, we follow such basic scheme. Then a high pass filter with diagonal edge detection capability would be applied directly to the CFA samples in that macro-block. This can be written as:

$$\bar{y}_{\bar{S}_i} = y_{\bar{S}_i} * Z \tag{16}$$

where $\bar{y}_{\bar{S}_i}$ is the filtered macroblock, Z is the high pass kernel. Now if the energy of the filtered CFA image in that macro-block ($\left\| \bar{y}_{\bar{S}_i} \right\|$) is more than a threshold δ, then that macro-block would be demosaiced using large block sizes. Otherwise, a smaller block size would be adapted. It should be clear that over-estimating and under-estimating the value of δ lead to false adaptions of smaller and larger block sizes respectively. In other words, over-estimating δ makes the demosaiced image sharper and reduces the computational time and at the same time might introduce artifacts in the recovered image. On the other hand, under-estimating δ leads to higher PSNRs but also causes color-washed patches and increases the computational time.

[5] Since a weighted averaging scheme increases the lower bound and meanwhile decreases the upper bound of the achievable PSNRs.

Let us briefly justify this strategy in here. As presented in the following subsection, our sparsifying dictionary $\boldsymbol{\Psi}$ is based on 2D-DCT. There exist atoms in 2D-DCT dictionary in the shape of vertical and horizontal strips which can express vertical and horizontal edges very effectively, even in small block sizes. On the other hand, it is known that DCT is much less effective in sparsely expressing diagonal edges [26][27][25][30]. Thus in case of a DCT based sparsifying dictionary, a CS-decoder would less likely converge to the correct solution in the regions with diagonal edges. Now if a small block size is adapted for demosaicing a region with diagonal edges, then visible artifacts would be introduced in that region (see Figure 3). On the other hand, adapting a larger block size for demosaicing of the same region, would spread the error in the whole region which is less visible. Now, it has been known (and can be verified) that, CFA samples of a scene look like the luminance of the same scene with mosaic effects. Thus applying a filter to CFA samples directly gives a sense of the actual output of the original image when filtered using the same filter. The role of the threshold δ is to detect real diagonal edges in the image and ignoring the mosaic patterns of the CFA samples. In practice, we have observed that this strategy leads to significant visual improvement and considerably large PSNR boost (approximately 1db on average for all Kodak test images) to the demosaiced images. Algorithm 1 summarizes the proposed strategy.

Input: $y, \bar{S}, b_1, b_2, \delta, Z$

Output: S

$S = \emptyset$

$\bar{y} = y * Z$

For $i = 1$ *to* $Card\{\bar{S}\}$

  If $\|\bar{y}_{\bar{S}_i}\| \geq \delta$ *then*

    $S \leftarrow S \cup \{all\ blocks\ of\ size\ b_1 \times b_1\ in\ \bar{S}_i\}$

  Else

    $S \leftarrow S \cup \{all\ blocks\ of\ size\ b_2 \times b_2\ in\ \bar{S}_i\}$

  End

End

Algorithm 1: The proposed block division strategy.

For our simulations, we have trained a filter to detect diagonal edges from CFA samples. More specifically, among almost nine million possible patches of size 16×16 in the Kodak image set, we identified $5{,}000$ CFA patches which

are best to be demosaiced in blocks of size 16×16 and assigned label -1 to all these patches. Similarly, we found 5,000 CFA patches of size 16×16 which are better to be demosaiced in blocks of size 8×8 and assigned label $+1$ to these patches. After making those CFA samples zero mean and unit norm, we input such training data to an off-the-shelf linear SVM trainer [43] to find such filter from the CFA samples. The derived filter that we also utilized in our simulations is presented in Figure 4. Note that the trained filter is indeed a simple high pass filter, as we claimed before in this sub-section. Also in our simulations, we set the value of δ to $\|B * Z\|$ where B has the same size of $\bar{y}_{\bar{S}_i}$ and all of its entries equal to $E(\bar{y}_{\bar{S}_i})$. It is straightforward to verify that by using this value of δ, the derived filter would serve as a support vector, separating CFA samples that are better to be demosaiced in blocks of 8×8 from those that are better to be demosaiced in blocks of 16×16.

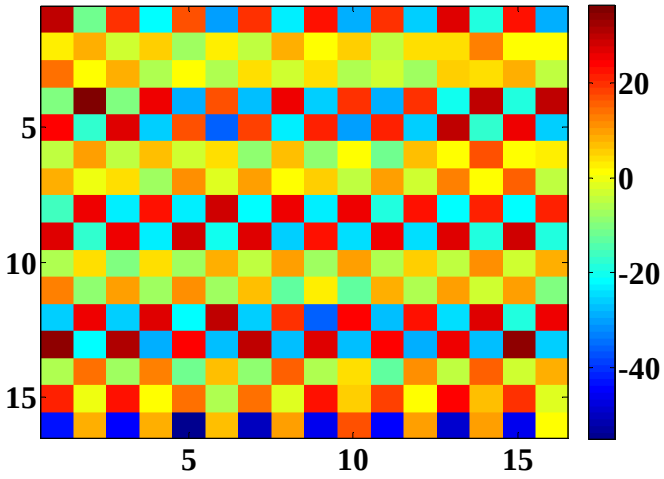


Figure 4 A high pass filter, trained for identifying the proper block size in the demosaicing process.

Before concluding this subsection, let us discuss the effect of overlap among patches on the quality of the demosaiced images and also on the complexity of CD. As stated before, we need to allow some overlap among patches during the demosaicing process to avoid blocking artifacts. Clearly, as we increase the overlap amount, the quality of the demosaiced image and also the computational time will increase. Figure 5 illustrates this effect. In Figure 5-a and Figure 5-b, we investigated the effect of increasing the overlap (among CFA patches) on the quality of demosaicing and also on the required time to demosaic Kodak test image 23 when the block division strategy is fixed 16×16. Figure 5-c and Figure 5-d show the same functions when Kodak test image 3 is demosaiced by adapting a block division strategy of fixed 8×8. For these plots, we chose these two images since among all test images, Kodak test images 3 and 23 would take the longest time to be demosaiced by block sizes 8×8 and 16×16

respectively. Note that after a sufficient amount of overlaps among patches (e.g. 3 for block size 8×8 and 6 for 16×16), the quality of demosaicing will saturate. In other words, no significant improvement on the quality of demosaiced images would be observed after those amounts of overlaps while computational time increases non-linearly. Figure 5 shows that it would take approximately 14 seconds in the worst case to demosaic any Kodak test image by using our CD implementation in Matlab to get virtually the best result when the block division strategy is fixed 8×8. On the other hand, it would take less than 100 seconds to demosaic any of Kodak test images to get nearly best quality, if all patches are demosaiced in block sizes of 16×16. Meanwhile, and as shown in the simulation result section, adapting a fixed block size is not the optimal strategy and adapting different block sizes would be beneficial in improving the quality of demosaicing. For instance, the introduced high pass filter (illustrated in Figure 4) predicted that out of 1536 non-overlapping patches of 16×16 in Kodak test image 23, only 666 of them need to be demosaiced by block size of 16×16 and the rest should be demosaiced by blocks of size 8×8 (leading to an effective running time of less than one minute to get nearly the best result).

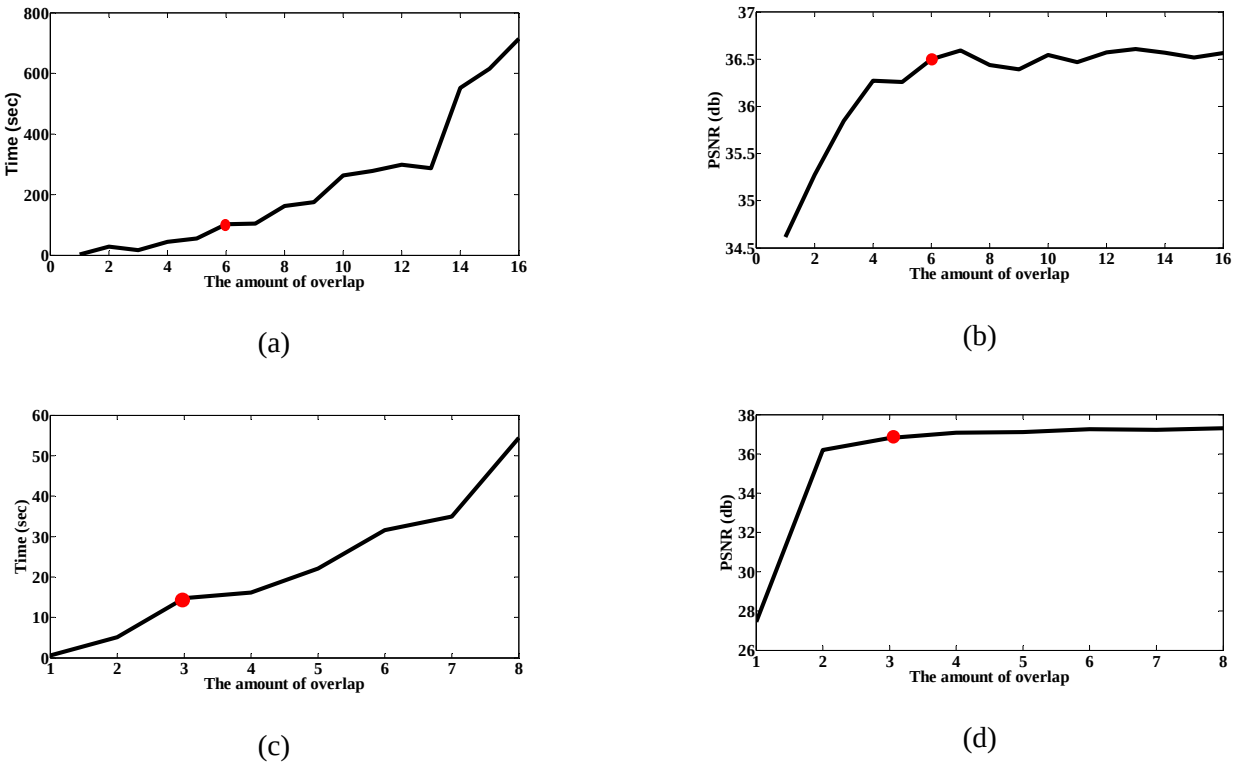


Figure 5 (a) The required time to demosaic Kodak test image 23 as a function of the amount of overlap among patches, when the block division strategy is fixed 16×16. (b) PSNR of demosaiced Kodak test image 23 as a function of the amount of overlap among patches, when the block division strategy is fixed 16×16. (a) Required time to demosaic Kodak test image 3 as a function of the amount of overlap among patches when the block division strategy is fixed 8×8. (b) PSNR of demosaiced Kodak test image 3 as a function of the amount of overlap among patches when the block division strategy is fixed 8×8.

E. Solving a system of linear equations

Without loss of generality and for the simplicity of notations, suppose that we divide the image $X = [R^T\ G^T\ B^T]$ into only one patch (block), i.e. $Q = 1$, $y = y^{(1)}$ and $\mathbf{\Psi} = \mathbf{\Psi}^{(1)}$. As before, assume that X has the representation of ξ in $\mathbf{\Psi}$ domain: $X = \mathbf{\Psi}\xi$. Inputing CFA samples $y = \mathbf{\Phi}X$ to any generic CS decoder, the representation ξ would be approximated as ξ^* by using the optimization problem (8). Now the quality of the demosaiced image is completely determined by the estimation error of $\xi - \xi^*$. Such error would be minimal if ξ is sufficiently sparse (guaranteeing the CS decoder to converge to the correct solution). This can be achieved if $\mathbf{\Psi}$ effectively captures the spatial and color correlations of that image patch.

Broadly speaking, there are two approaches for designing the sparsifying transform $\mathbf{\Psi}$ for natural color images: (a) methods which are inherently oblivious to the distinction between the spatial and color domains (e.g. [15][38]); and (b) approaches that are mindful of the distinction between the two types of correlations (spatial and color-channel) [32][34][36]. A prime example of the first approach is the one introduced in [38] where adaptive dictionaries were employed to sparsify color images. Despite of its superior results, such approach relies on image-dependent transforms which they learn during the demosaicing process and hence can suffer from high computational complexities. In this paper, we focus on the second approach (differentiating spatial and color correlations) and utilize the concept of fixed *non-separable frames*, introduced by authors in [37]. Before presenting the utilized sparsifying transform, let us review the concept of non-separable color frames in here.

Consider the signal $x = (x_1, x_2, \dots, x_l)$ where transform $\mathbf{\Psi}^{(i)}$ de-correlates $x_i \in \mathbb{R}^{d_i}$. Then, one can use the *separable* transform $\mathbf{\Psi}^{(1)} \otimes \mathbf{\Psi}^{(2)} \otimes \dots \otimes \mathbf{\Psi}^{(l)}$ to de-correlate the vectorized form of x. Some prior works [32][34][36] in CS-based demosaicing utilized inter-channel and inter-pixel correlations in this separable form. More specifically in those approaches [32][34][36], the sparsifying transform $\mathbf{\Psi}$ is formed by: $\mathbf{\Psi} = \mathbf{\Theta} \otimes \boldsymbol{\psi}$, where $\boldsymbol{\psi}$ and $\mathbf{\Theta}$ are sparsifying dictionaries for spatial and color domains, respectively. This separable formulation, which is similar to some traditional demosaicing methods (e.g.[3]) attempts to find spatial transform (e.g., DCT or Fourier) coefficients of the image in fixed color spaces (e.g., YUV). In [32], we utilized Equiangular Tight Frame (ETF) along with the luminance (Y) axis in a separable form for $\mathbf{\Theta}$ and 2D-DCT for the spatial transform $\boldsymbol{\psi}$. At first glance, this choice seems to be an optimal one since an ETF is the most incoherent redundant frame for a given number of atoms.

Although the ETF based solution may work well for image regions with limited colors, a demosaiced image can still have visible color shifts in other regions. To find the source of those artifacts for the separable case of $\boldsymbol{\Psi} = \boldsymbol{\Theta} \otimes \boldsymbol{\psi}$, let $\boldsymbol{\psi} \in \mathbb{R}^{N\times M}$, and define the permutation matrix $\boldsymbol{\Pi} \in \mathbb{R}^{3M\times 3M}$ as:

$$\boldsymbol{\Pi}_{i,j} = \begin{cases} 1 & j = M\big((i-1)\ mod\ 3\big) + \left\lfloor \frac{i}{3} \right\rfloor + 1 \\ 0 & \text{otherwise} \end{cases} \tag{17}$$

Hence, $\boldsymbol{\Pi}$ is in the form of:

$$\begin{bmatrix} \overbrace{\mathbf{1}\ \mathbf{0}\ \dots}^{M} & \overbrace{\mathbf{0}\ \dots\ \dots}^{M} & \overbrace{\mathbf{0}\ \dots\ \dots}^{M} \\ \mathbf{0}\ \mathbf{0}\ \dots & \mathbf{1}\ \mathbf{0}\ \dots & \mathbf{0}\ \dots\ \dots \\ \mathbf{0}\ \dots\ \dots & \mathbf{0}\ \dots\ \dots & \mathbf{1}\ \mathbf{0}\ \dots \\ \mathbf{0}\ \mathbf{1}\ \dots & \mathbf{0}\ \dots\ \dots & \mathbf{0}\ \dots\ \dots \\ \mathbf{0}\ \mathbf{0}\ \dots & \mathbf{0}\ \mathbf{1}\ \dots & \mathbf{0}\ \dots\ \dots \\ \mathbf{0}\ \dots\ \dots & \mathbf{0}\ \dots\ \dots & \mathbf{0}\ \mathbf{1}\ \dots \\ \ddots & \ddots & \ddots \end{bmatrix} \tag{18}$$

Now let us compare the solution of η^* in:

$$\begin{gathered} \eta^* = \arg\min_{\hat{\eta}} \|\hat{\eta}\|_1\ , s.t. \\ y = \boldsymbol{\Phi}[\boldsymbol{\Theta} \otimes \boldsymbol{\psi}_{:,1} \quad \boldsymbol{\Theta} \otimes \boldsymbol{\psi}_{:,2} \quad ...]\ \hat{\eta} \end{gathered} \tag{19}$$

to the solution of ξ^* in the original problem of (8). Note that, by expanding (8) in case of the separable sparsifying dictionary $\boldsymbol{\Psi} = \boldsymbol{\Theta} \otimes \boldsymbol{\psi}$, we have:

$$\begin{gathered} \xi^* = \arg\min_{\hat{\xi}} \left\|\hat{\xi}\right\|_1\ , s.t. \\ y = \boldsymbol{\Phi} \begin{bmatrix} \boldsymbol{\Theta_{1,1}\psi} & \boldsymbol{\Theta_{1,2}\psi} & \dots. \\ \boldsymbol{\Theta_{2,1}\psi} & \boldsymbol{\Theta_{2,2}\psi} & \dots \\ \boldsymbol{\Theta_{3,1}\psi} & \boldsymbol{\Theta_{3,2}\psi} & \dots \end{bmatrix} \hat{\xi} \end{gathered} \tag{20}$$

It is straightforward to verify that $\xi^* = \boldsymbol{\Pi^{-1}}\eta^*$ and hence $\|\eta^*\|_p = \|\xi^*\|_p$ for any $p \geq 0$. In words, η^* is simply a permuted (re-ordered) version of ξ^*. Consequently, solving (20) is equivalent to solving the original formulation of (8). On the other hand, studying the implications of color dictionary $\boldsymbol{\Theta}$ in (20) is easier when compared to that one in the original formulation (8). Hence, let us focus on constraints of (20) in here.

Let $\boldsymbol{\Theta} \in \mathbb{R}^{\mathbf{3}\times\boldsymbol{q}}$ and define $\eta^{(i)} \in \mathbb{R}^q$ as:

$$\eta^{(i)} = \begin{bmatrix} \eta_1^{(i)} \\ \eta_2^{(i)} \\ \vdots \\ \eta_q^{(i)} \end{bmatrix} = \begin{bmatrix} \eta^*_{(i-1)q+1} \\ \eta^*_{(i-1)q+2} \\ \vdots \\ \eta^*_{iq} \end{bmatrix} \tag{21}$$

Then the following holds true:

$$\begin{bmatrix}\hat{R}_i\\ \hat{G}_i\\ \hat{B}_i\end{bmatrix} = \boldsymbol{\Theta}\eta^{(i)} = \boldsymbol{\Theta}\begin{bmatrix}\eta^*_{(i-1)q+1}\\ \eta^*_{(i-1)q+2}\\ \vdots\\ \eta^*_{iq}\end{bmatrix} \tag{22}$$

where $\hat{R}$, $\hat{G}$ and $\hat{B}$ are representations of color planes red, green and blue of the original image X, in the $\boldsymbol{\psi}$ domain, i.e.: $R = \boldsymbol{\psi}\hat{R}$, $G = \boldsymbol{\psi}\hat{G}$ and $B = \boldsymbol{\psi}\hat{B}$. Extending (22) to the whole vector of η we get:

$$\begin{bmatrix}\hat{R}_1\\ \hat{G}_1\\ \hat{B}_1\\ \hat{R}_2\\ \hat{G}_2\\ \hat{B}_2\\ \vdots\end{bmatrix} = \begin{bmatrix}\boldsymbol{\Theta} & \mathbf{0} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\Theta} & \mathbf{0} & \mathbf{0}\\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Theta} & \mathbf{0}\\ \mathbf{0} & \dots & \dots & \ddots\end{bmatrix}\eta^* \tag{23}$$

In this formulation, the role of $\boldsymbol{\Theta}$ is revealed where it can be considered as a frame to de-correlate (sparsify) the triplets $\{(\hat{R}_i, \hat{G}_i, \hat{B}_i)\}$. Unfortunately, designing a $\boldsymbol{\Theta}$ which can sparsify $\{(\hat{R}_i, \hat{G}_i, \hat{B}_i)\}$ for all frequencies i, is cumbersome and practically infeasible. This is due the fact that if an image patch has more than one color (which is most likely the case), then the optimum $\boldsymbol{\Theta}$ changes for different frequencies as well. Consequently η^* (and equivalently ξ^*) might not be maximally sparse. This makes the CS-solver to diverge from the correct solution and hence introduces some artifacts in the demosaicing process. In summary, adapting a separable formulation translates into using a non-flexible scheme where only a single frame $\boldsymbol{\Theta}$ (which is not optimal for all spatial frequencies) is deployed for designing the sparisyfing transform.

The authors in [37] proposed utilizing non-separable Incoherent Color Frames (ICF) as a remedy for inflexible separable formulations. In a nutshell, in a non-separable ICF, (23) is replaced by:

$$\begin{bmatrix}\hat{R}_i\\ \hat{G}_i\\ \hat{B}_i\end{bmatrix} = \boldsymbol{\Theta_i}\eta^{(i)} \tag{24}$$

where $\boldsymbol{\Theta_i} \in \mathbb{R}^{\mathbf{3}\times \boldsymbol{q_i}}$. This consequently changes (24) as follows:

$$\begin{bmatrix} \hat{R}_1 \\ \hat{G}_1 \\ \hat{B}_1 \\ \hat{R}_2 \\ \hat{G}_2 \\ \hat{B}_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Theta_1} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{\Theta_2} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{\Theta_3} & \mathbf{0} \\ \mathbf{0} & \dots & \dots & \ddots \end{bmatrix} \eta^* \tag{25}$$

Clearly the flexibility of using different color frames for different spatial frequencies increases the chance to sparsely express η^* (and hence ξ^*). Consequently, it would be more likely that CS solver converges to a solution with a smaller error, (i.e. with higher quality of demosaicing). In this paper, we follow such approach where frequency indices $[N] = \{1,2, \dots, N\}$ would be partitioned into K groups $\nu_i \subseteq [N]$, i.e.: $i, j \in [K], \cup \nu_i = [N]$ and $\nu_i \cap \nu_j = \emptyset$ for $i \neq j$. Then for the i-th frequency group, an ICF $\boldsymbol{\Theta}_i \in \mathbb{R}^{3 \times q_i}$ that has q_i color atoms would be utilized. That is: $\forall a, b \in \nu_i$: $\boldsymbol{\Theta}_a = \boldsymbol{\Theta}_b$. Thus, given the frequency groups $\{\nu_i\}_i$, spatial dictionary $\boldsymbol{\psi}$ and the color coordinates $\{\boldsymbol{\Theta}_i\}$, the final sparsifying dictionary is formed by:

$$\boldsymbol{\Psi} = [\boldsymbol{\Theta}_1 \otimes \boldsymbol{\psi}_{:,\nu_1} \quad \dots \quad \boldsymbol{\Theta}_K \otimes \boldsymbol{\psi}_{:,\nu_K}] \tag{26}$$

Note that in an extreme case, where all $\boldsymbol{\Theta}_i$ are the same, then the non-separable formulation (25) turns into the separable format. Another extreme case might be the case where we design a distinct color frame for each frequency, i.e. $K = N$. For a fixed block size $b \times b$, Algorithm.2 summarizes the training step for $K = N$, where $\{\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \dots, \boldsymbol{X}^{(d)}\}$ is the collection of d training full color patches where $X^{(i)} = Vec(\boldsymbol{X}^{(i)}) = [R^{(i)}\ G^{(i)}\ B^{(i)}]$. Pair of (u_j, l_j) denotes the lower and upper bounds for the number of atoms in the corresponding color frame $\boldsymbol{\Theta}_j$. Providing bounds u and l is usually mandatory, since virtually none of dictionary learning algorithms can find the optimal number of atoms for a dictionary. Also note that, this training process is performed off-line and once the optimal color frames are found, the demosaicing process is very simple. We have observed that as the spatial frequency increases, a decrease on the number of atoms in the corresponding color frame leads to higher quantitative and qualitative measures for the demosaiced images. In fact, after a few (e.g. only two) diagonals in the zig-zag ordering, it seems that $\boldsymbol{\Theta}_i = [1/\sqrt{3} \ \ 1/\sqrt{3} \ \ 1/\sqrt{3}\ ]$ which is a vector along the luma axis, is the optimum color frame (for high spatial frequency atoms). The same result (although heuristically) has been reported by authors in [37]. This can be justified by considering the well-known fact that the human visual system is much less sensitive to high frequency chrominance changes. Hence for simplicity of design and by choosing $[1/\sqrt{3} \ \ 1/\sqrt{3} \ \ 1/\sqrt{3}\ ]$ as the

only color coordinate for high frequency atoms, we enforce the demosaiced image to have the same high frequency coefficients for luma and chroma. This also prevents the CS-decoder to choose high frequency chroma atoms that might lead to color artifacts. Finally, the color frame learning process has been done using SPAMS software [42].

Input: $\{\boldsymbol{X}^{(1)}, \boldsymbol{X}^{(2)}, \dots, \boldsymbol{X}^{(d)}\}, \epsilon \geq 0,$ and $l, u \in \mathbb{N}^{b^2}$

Output: $\{\boldsymbol{\Theta}_i\}$

For $i = 1$ *to* b^2

$$\boldsymbol{Z} = \begin{bmatrix} \boldsymbol{\psi}_{i,:}R^{(1)} & \boldsymbol{\psi}_{i,:}R^{(2)} & \dots & \boldsymbol{\psi}_{i,:}R^{(d)} \\ \boldsymbol{\psi}_{i,:}G^{(1)} & \boldsymbol{\psi}_{i,:}G^{(1)} & \dots & \boldsymbol{\psi}_{i,:}G^{(1)} \\ \boldsymbol{\psi}_{i,:}B^{(1)} & \boldsymbol{\psi}_{i,:}B^{(1)} & \dots & \boldsymbol{\psi}_{i,:}B^{(1)} \end{bmatrix}$$

For $j = l_i$ *to* u_i

$L_j = \min_{\boldsymbol{T}_j} \|\alpha_{:,i}\|_1$ *s.t.* $\boldsymbol{T}_j \in \mathbb{R}^{3\times j}$ *and*

$\forall c \in [d]: \|\boldsymbol{Z}_{:,c} - \boldsymbol{T}_j \alpha_{:,c}\| \leq \epsilon$

End

Let a be the index where L gets its minimum.

$\boldsymbol{\Theta}_i \leftarrow \boldsymbol{T}_a$

End

Algorithm 2 The algorithm for learning the color frames.

F. Finding the best estimate

As stated before, a CS-based demosaicing algorithm is usually run on overlapped blocks of the image to avoid the blocking effects. Hence, except the pixels around the image borders, a number of "votes" for colors of each pixel is available. Let us use $\bar{r}_i = \{r_1, r_2, \dots, r_u\}$, $\bar{g}_i = \{g_1, g_2, \dots, g_u\}$ and $\bar{b}_i = \{b_1, b_2, \dots, b_u\}$ to denote the available u votes for respectively the red, green and blue values of the i-th pixel. Also, define $Z_i = (\bar{r}_i, \bar{g}_i, \bar{b}_i)$. Then a generic demosaicing method estimates the colors of that pixel in the RGB domain by:

$$\left(\hat{R}_i, \hat{G}_i, \hat{B}_i\right) = \left(F_r(Z_i), F_r(Z_i), B_r(Z_i)\right) \tag{27}$$

where $F_r(.), F_g(.), F_b(.): \mathbb{R}^{3u} \rightarrow \mathbb{R}$. In practice, $F_r(.)$, $F_g(.)$ and $F_b(.)$ only operate on $\bar{r}_i$, $\bar{g}_i$ and $\bar{b}_i$ respectively (i.e. $F_r(.), F_g(.), F_b(.): \mathbb{R}^u \rightarrow \mathbb{R}$). Common functions for these operators are median and average (e.g. $F_r(\bar{r}_i) = \sum_j r_j/u$). Now, recall that the success of a block based demosaicing approach highly dependens on the block size and also the

block position. More specifically, since we have used (shift-variant) DCT in design of the sparsifying transform, then it might be the case that some blocks do not have a sparse representation with respect to the utilized sparsifying dictionary due to bad block positioning. Thus for those blocks, the CS-solver would not converge to the correct solution, leading to demosaiced blocks with artifacts. Therefore, the voting functions $F_r(.)$, $F_g(.)$ and $F_b(.)$ would be affected particularly in those regions. This consequently reduces the quality of the reconstructed image. Hence, it would be of great importance to prevent this event by suppressing the effect of erroneous estimates. However, we are facing a challenge that we do not have access to the original image itself to distinguish erroneous estimates. On the other hand, in our simulations, we observed that there are measures indicating the quality of a demosaiced block. This measure is simply the compressibility of a demosaiced block in the selected sparsifying transform for that block. More specifically let us define the function $G(x,a)$ as follows:

$$G(x,a)=\frac{Card\{i:|x_i|>a\}}{Card\{x\}} \tag{28}$$

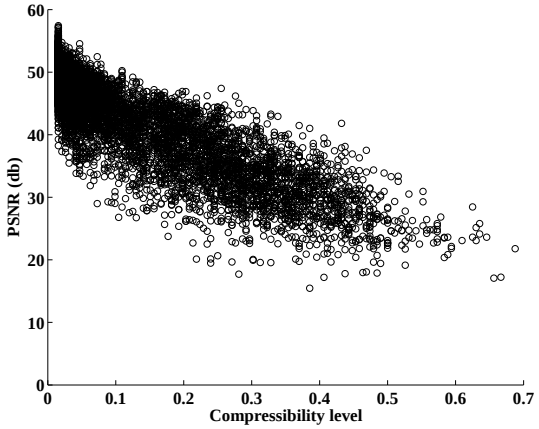
In words, for a signal x, $G(x,a)$ measures the ratio of the number of elements in x with magnitudes larger than a to the length of that signal. Note that, smaller values of $G(x,a)$ translate into a more compressible (or sparse) signal and for the special case of $z=0$, $G(x,a)$ would be simply the sparsity ratio of that signal. Proper tuning of parameter a in (28) might provide us a reliable indicator function for identifying demosaiced blocks with artifacts. This can be justified by recalling that, the CS-decoder uses (8) to estimate ξ, the true representation of an image patch in the selected sparsifying transform $\mathbf{\Psi}$. The estimated value ξ^* would be compressible if the dictionary $\mathbf{\Psi}$ effectively captures color and spatial correlations (i.e. ξ is sufficiently sparse or compressible). On the other hand, a non-compressible estimation ξ^* possibly warns about the event when the CS-decoder has diverged from the correct solution ξ due to unsuitable sparsifying transforms or bad block positioning where several objects are inside that block.

To investigate the effectiveness of the measure $G(x,a)$ in identifying erroneous estimates, we have computed representations ξ^* for $10{,}000$ random patches of a Kodak test image (Kodim3). Furthermore, to study effects of the block size on that measure, we repeated the same process for block sizes 8×8 and 16×16. In Figure 6, the PSNRs of the demosaiced blocks ($\mathbf{\Psi}\xi^*$) as a function of $G(\xi^*,10)$ are plotted. Interestingly, it seems that a linear function is governing the relation between the PSNR and the proposed simple function. This suggests us that, given a set of

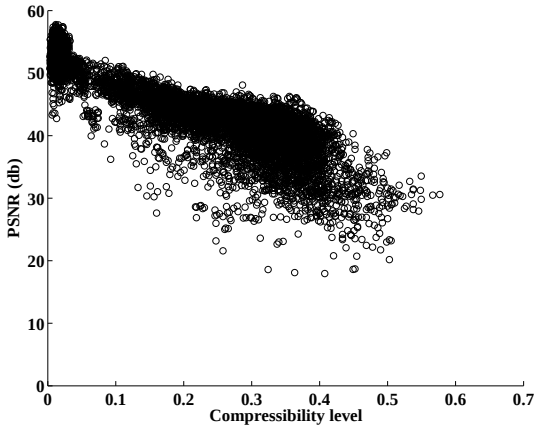
votes, instead of adapting simple average or median for voting functions $F_r(.)$, $F_g(.)$ and $F_b(.)$ in (27), a more efficient strategy such as weighted mean could be employed in estimating the colors of a pixel. More specifically, assume that in the first step of CD (dividing the image into patches), the i-th pixel falls into u patches with indices $I_i = \{I_{i,1}, I_{i,2}, \dots, I_{i,u}\}$, that is: $\forall j \in [u]: i \in S_{I_{i,j}}$, where S_j denotes the pixel indices belonging to the j-th block. Thus, we would have u approximations for the colors of that pixel: $\bar{r}_i = \{r_1, r_2, \dots, r_u\}$, $\bar{g}_i = \{g_1, g_2, \dots, g_u\}$ and $\bar{b}_i = \{b_1, b_2, \dots, b_u\}$. Then we estimate the colors of that pixel by the triplet: $\left(F_r(\bar{r}_i), F_g(\bar{g}_i), F_b\left(\bar{b}_i\right)\right)$ in the RGB color coordinate where:

$$F_r(\bar{r}_i) = \frac{\sum_{j=1}^{u} \omega_j r_j}{\sum_{j=1}^{u} \omega_j}, F_g(\bar{g}_i) = \frac{\sum_{j=1}^{u} \omega_j g_j}{\sum_{j=1}^{u} \omega_j}, F_b\left(\bar{b}_i\right) = \frac{\sum_{j=1}^{u} \omega_j b_j}{\sum_{j=1}^{u} \omega_j} \tag{29}$$

are the weighted means of the estimated values in the respective color plane, and the weighting coefficient for the j-th block (which i-th pixel belongs to) is inversely proportional to the compressibility level of that block: $\omega_j = 1/G(\xi^*_{(I_{i,j})}, a)$ where $\xi^*_{(t)}$ is the representation of t-th block in $\mathbf{\Psi}$, the selected sparsifying dictionary for that block.



(a) 10,000 random patches demosaiced by block size of 8×8.



(b) 10,000 random patches demosaiced by block size of 16×16.

Figure 6 The measure of compressibility of a block might be a good indicator function for identifying erroneous estimates. Also demosaicing of larger block sizes on average leads to a lower quality but stable reconstructions.

For a fixed block size $b \times b$ and a compressibility level $G(\xi^*, a)$, the PSNRs of the demoasiced patches can be modeled as a random variable with mean μ_b and the variance of σ_b^2. In our simulations, we have observed that, if $b_1 > b_2$ then $\mu_{b_1} < \mu_{b_2}$ and $\sigma_{b_1}^2 < \sigma_{b_2}^2$. In other words, as the block size increases, the expected quality of a demosaiced patch would reduce, however at the same time, it would be unlikely to have a reconstructed patch with too many artifacts. Meanwhile, adapting a smaller size block leads to either very low quality demosaiced patches or virtually noise free ones. This claim is clearly illustrated in Figure 6, where the plot associated with the block size of 8×8 is thicker. We refer to this phenomenon as the "stability of approximations" for larger block sizes. Despite the relative effectiveness of the measure of compressibility level in identifying erroneous estimates, such measure is not optimal and could be improved upon furthermore. Designing an optimal measure for detecting erroneous estimates is also subject of our future work. Finally, a post processing step often enhances the quality of the demosaiced images. In our simulations, a median filter has been utilized to reduce random color artifacts [8].

III. Simulation Results

The number of demosaicing methods, proposed in the community is so large that we can present only some of the leading approaches to compare them with the proposed CD framework. To that end, we have selected the method of DL [35], LPA [13] and LSSC [38] as benchmarks for evaluating the performance of the proposed CD framework. Note that the first two methods (DL and LPA) follow a classical (non CS) approach to demosaicing, while CD and LSSC are CS-based demosaicing frameworks. Consequently, the first two methods are much faster compared to their CS-based counterparts. For instance, DL and LPA on average take 30 and 7 seconds, respectively, to demosaic any image in our test platform. On the other hand and as stated before, LSSC (in its optimum setting) demands an online sparsifying dictionary learning which is computationally expensive. This learning process during the task of demosaicing makes this algorithm to achieve the highest PSNRs reported (with a fairly significant margin) for standard test images. On the other hand, the proposed CD is a much simpler algorithm (compared to LSSC) that uses fixed, non-adaptive sparsifying dictionary in the demosaicing process. For example, the available implementation of LSSC (which is mainly in C) took slightly more than one minute on average[6] to demosaic each test image by using a block size of 8×8. On the other hand and as shown before in Figure 4, our implementation of CD (which is all in Matlab scripts except the mex file from LSSC framework, implementing the algorithm of Lasso) could achieve nearly the best results possible in approximately 14 seconds, if images are demosaiced by blocks of size 8×8.

Although the CD framework, even in its non-optimized setting (to be discussed shortly), outperforms (in terms of PSNR) such complex demosaicing algorithm in some test images, we considered the reported PSNRs for LSSC as arguably the maximum PSNR that a CS-based algorithm can achieve. From that prospective, the performance of LSSC, and due to its optimized dictionary learning strategy during the real-time demosaicing process, can be considered as a performance-bound in terms of PSNR values. It is important to note though that our design of the CD framework is biased toward improving visual/subjective qualities rather than the quantitative measure of PSNR. As we show later in this section, there are many instances where images demosaiced by CD have higher visual qualities and at the same time they have lower PSNRs compared to other methods. This can be explained by noting that in our current implementation, the color coordinates corresponding to high spatial frequencies only

[6] All simulations were run in an Intel 3.4 GHz (64 bit architecture) machine with Linux-Matlab 2011b and 4 GB of memory.

have an atom along the luma axis $\left[\frac{1}{\sqrt{3}}\frac{1}{\sqrt{3}}\frac{1}{\sqrt{3}}\right]$. In other words, the utilized sparsifying dictionary $\boldsymbol{\Psi}$ in high spatial frequencies, can only span the luma (and not chroma) components. Thus, the reconstructed images by our current implementation of CD might have considerable errors in high frequency parts of chroma channels. On the other hand, the human visual system is much less sensitive to this range of frequencies and thus the error is not even visible to the human eye.

We have demosaiced Kodak test images using the following settings:

- In the sensing stage, both Second Generation CFA [5] and Bayer CFA are investigated.
- For the block division strategy we considered four scenarios: (a) fixed block size of 8×8 (b) fixed block size of 16×16 (c) an adaptive block size, where the trained filter shown in Figure 4 selects the block size to be adapted for each macro-block of 16×16 and (d) an oracle setting where the image is partitioned into macroblocks of 16×16 and assuming that there exists a filter which can select the optimal block size between two options of 8×8 and 16×16 for that macroblock.
- For training the color coordinates $\boldsymbol{\Theta}_i$, we have considered four spatial bands in our simulations: DC, Low Frequency (LF), Middle Frequency (MF) and High Frequency (HF). For each spatial frequency, we trained (off-line) an ICF. However, the number of atoms of ICFs in each band is assumed to be equal. In particular (regardless of the block size), the number of atoms of ICFs in DC, LF, MF and HF are respectively 30, 10, 4 and 1 in our simulations.

The PSNRs of demosaiced images are reported in Table 1. Similar to other available comparisons [38] and to provide fairness, a border of 15-pixels has been excluded in computing the PSNRs. As clearly illustrated, not only the proposed CD even in the non-optimal setting can outperform DL and LPA in most test images, it also outperforms the complex method of LSSC in some test images. Also by comparing the oracle results of CD with those from filtering strategy (the last two columns of Table 1), we observe that the trained filter for CD is quite precise in selecting the appropriate block size under the block division strategy.

Finally in Figure 7**,** we have presented a few instances where CD reconstructed images with less visible artifacts, proving our claim that CD targets improving visual qualities rather than focusing on improving the PSNR measure. For further results (including those from different block strategy division and CFAs), we refer the readers to our website [44] where all demosaiced images are available.

Here we would like to note that relative to performance of the configuration of CD with second generation CFA along with the filtering strategy, which is sub-optimal and can be tuned, the qualities of demosaiced images can be improved further under a more general and more optimal CD setting. Ideally, to find the optimal setting for the CD framework one would need to find (a) optimal CFAs (b) optimal sparsifying spatial dictionaries (c) the corresponding optimal color dictionaries for the deployed spatial dictionary, and we need to tune (d) parameters of the CS solver (e.g. regularization parameters and so on). Naturally, all of these combinations of optimal settings are interrelated, and hence, finding the truly optimal CD solution is not feasible in practice. For instance, designing the color dictionary depends on finding the optimal spatial sparsifying dictionaries which itself is an active area of research. Furthermore, the degrees of freedom in designing a color dictionary (e.g. how to divide the spatial atoms into frequency bands, what is the number of atoms used in each band, and so on) are so high that it makes an exhaustive search infeasible. For instance, using Algorithm 2 with a block size of $b \times b$ and K spatial frequency groups $\nu_1, \dots, \nu_K$, we require

$$\sum_{i=1}^{K} (u_i - l_i) Card\{\nu_i\} \tag{30}$$

runs of dictionary learning, where $l_i \leq |\boldsymbol{\Theta}_i| \leq u_i$ for $i \in [K]$. Consequently, if we do not divide the spatial frequencies into bands and let the number of atoms of each color coordinate to be independent, then $K = b^2$ and the training algorithm would be very time consuming. In words, we have chosen the optimum ICFs among a limited number of configurations in our current implementation of CD. Clearly, it is very unlikely that the employed ICFs are truly the globally optimum ones. Consequently, the design of an optimal color dictionary for demosaicing might require a newly designed algorithm, which could be the subject of another paper.

In summary, we leave finding the optimal configuration of CD for future works. Nevertheless, we briefly highlight some possible directions that might lead to improvements on our results. We have observed that randomly replacing the cells of a SG CFA with a pure (Bayer like) CFA, approximately increases the PSNRs by 0.3 db on average. For the block division strategy, non-local mean and strategies similar to [38] might be employed to increase the PSNRs of reconstructed images further. Also in this paper, we only utilized DCT as the sparsifying spatial dictionary. Finding similar frequency bands for other types of dictionaries (e.g. different types of wavelets, trained dictionaries

and so on) and designing corresponding color dictionaries for them might lead to further improvements to the visual quality of CD.

Table 1 Comparing PSNRS of demosaiced images by methods of DL [35], LPA [13], LSSC [38] and six configurations of CD. Here SG, BA, -Fi and -O respectively stand for Second Generation CFA, Bayer CFA, Oracle and Filtering strategies in selecting the proper block size for each macro-block of size 16×16.

Image number	LSSC	DL	LPA	CD-BA-8 × 8	CD-BA-16 × 16	CD-SG-8 × 8	CD-SG-16 × 16	CD-SG-FI	CD-SG-O
1	41.36	38.46	39.45	37.68	37.71	38.28	41.30	**41.51**	**41.51**
2	42.24	40.89	**41.36**	38.54	38.52	40.92	40.52	41.27	41.27
3	44.24	42.66	**43.47**	36.96	37.30	41.54	39.59	42.56	42.56
4	42.45	40.49	40.84	40.94	41.07	39.85	39.43	**40.86**	**40.86**
5	39.45	38.07	37.51	37.62	37.81	37.64	36.59	**38.57**	**38.57**
6	41.71	40.19	40.92	39.10	39.27	39.32	41.18	**41.71**	**41.71**
7	44.06	42.35	**43.06**	41.44	41.60	42.09	40.93	42.57	42.58
8	37.57	35.58	37.13	35.20	35.41	36.76	38.64	**38.75**	**38.75**
9	43.83	43.05	**43.50**	42.19	42.45	42.48	42.37	43.43	43.44
10	43.33	42.54	42.77	42.19	42.25	41.87	41.75	**42.91**	**42.91**
11	41.51	40.01	40.51	38.99	39.25	39.33	40.44	**40.80**	**40.81**
12	44.90	43.45	44.01	43.36	43.50	43.63	43.07	44.52	**44.53**
13	36.35	34.75	36.08	33.91	34.11	32.97	36.41	**36.43**	**36.43**
14	38.77	**36.91**	36.86	35.86	36.17	36.09	34.29	36.52	36.52
15	41.74	39.82	**40.09**	39.13	39.02	38.86	33.88	39.77	39.77
16	44.91	43.75	44.02	42.54	42.75	42.44	44.65	**44.73**	**44.73**
17	41.98	41.68	41.75	40.78	41.07	41.06	42.28	**42.33**	**42.33**
18	38.38	37.64	37.59	36.93	36.78	36.40	37.94	**38.46**	**38.46**
19	42.31	41.01	41.55	39.97	40.13	41.30	42.21	**42.65**	**42.65**
20	42.27	41.24	41.48	40.42	40.57	41.04	41.52	42.37	**42.38**
21	40.65	39.10	39.61	38.25	38.48	38.74	40.71	**40.98**	**40.98**
22	39.24	38.37	38.44	37.90	38.16	38.57	38.32	39.03	39.03
23	44.34	43.22	**43.92**	38.70	38.69	42.55	39.18	42.95	42.95
24	35.89	35.55	35.44	34.74	34.76	35.13	36.68	36.80	**36.81**
Average	41.40	40.03	40.47	38.89	39.03	39.54	39.75	40.94	40.94

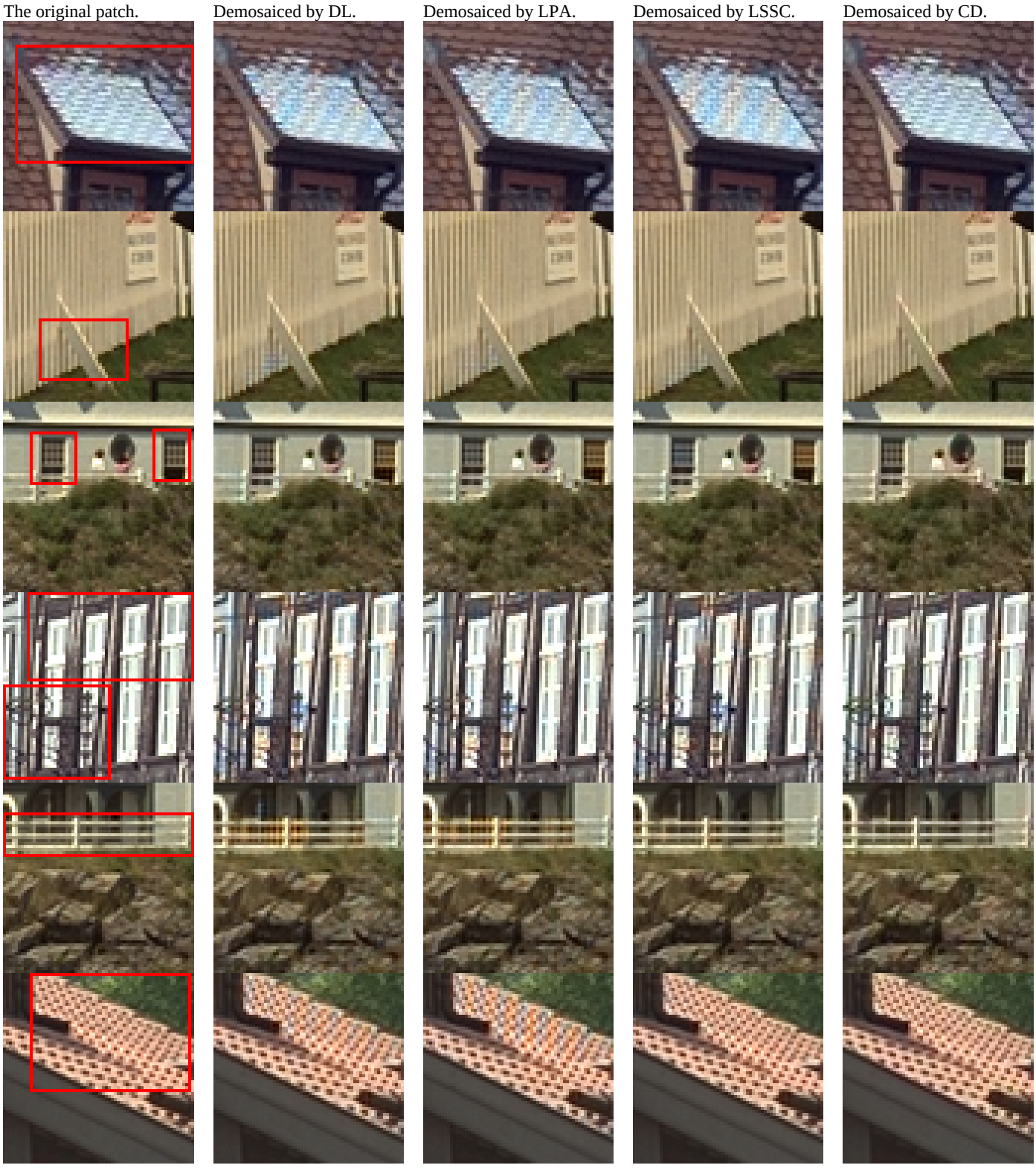


Figure 7 Comparing the visual qualities of demosaiced images on some patches from Kodak test images. For the CD framework in these figures, we have used the Second Generation CFA in the sampling stage and utilized the trained filter shown in Figure 5 to select the proper block size to demosaic each macro-block of size 16×16. Columns from left to right: (1) The original patch, (2) demosaiced by DL, (3) demosaiced by LPA, (4) demosaiced by LSSC and (5) demosaiced by CD.

IV. CONCLUSION

In this paper, we presented Compressive Demosaicing (CD), a framework based on Compressed Sensing for demosaicing of natural images. Broadly speaking, given the CFA samples, CD finds the sparse representation of the underlying image in a fixed sparsifying dictionary. In the design of such dictionary, inter-pixel and inter-channel correlations are treated separately. However, the sparsifying transforms for each of those domains would be mixed in a non-separable form to construct the final sparsifying dictionary. In the design of the sparsifying transforms for the color domain, we also considered some well-known facts about the human visual system. Our simulation results have shown that CD can outperform state of the art demosaicing approaches visually and under the quantitative measures in most of the test images.

REFERENCES

[1] J. Nakamura Image Sensors and Signal Processing for Digital Still Cameras. CRC Press 2005.
[2] US patent 3971065, Bryce E. Bayer, “Color imaging array”, issued 1976-07-20.
[3] B.K.Gunturk, J.Glotzbach and et. al, “Demosaicking: color filter array interpolation,” IEEE SP Magazine, 22(1):44-54, Jan 2005.
[4] K. Hirakawa and T. W. Parks, “Adaptive homogeneity-directed demosaicing algorithm,” IEEE Transactions on IP, 14(3):360-369, Mar 2005.
[5] K. Hirakawa and P. J. Wolfe, “Spatio-spectral color filter array design for optimal image recovery,” IEEE Trans. IP, 17(10):18761890, Oct. 2008.
[6] B.K.Gunturk, Y. Altunbasak, R. M. Mersereau, “Color plane interpolation using alternating projections,” IEEE Trans. IP, 11(9):997-1013, 2002.
[7] P. W. H. Lemmens and J. J. Seidel. “Equiangular lines”. J. Algebra, 24:494512, 1973.
[8] W. T. Freeman, “Median filter for reconstructing missing color samples,” U.S. Patent 4 724 395, 1988.
[9] X. Li, “Demosaicing by successive approximation,” IEEE Transactions on IP, 14(2):370-379, 2005.
[10] W. Lu and Y. Tan. “Color filter array demosaicking: New method and performance measures”. IEEE Trans. on IP, 12(10):1194-1210, Oct. 2003.
[11] R. Kimmel, “Demosaicing: image reconstruction from color CCD samples,” IEEE Trans. on IP, 8(9):1221-1228, 1999.
[12] X. Wu and N. Zhung, “Primary-consistent soft-decision color demosaicking for digital cameras”, IEEE Trans. on IP, 13(9):1263 - 1274, Sep 2004.
[13] D. Paliy, V. Katkovnik and et. al., “Spatially adaptive color filter array interpolation for noiseless and noisy data”. International Journal of Imaging Systems and Technology, 17(3): 105 - 122, Oct. 2007.
[14] D. Menon, S. Andriani, G. Calvagno, “Demosaicing with directional filtering and a posteriori decision”. IEEE Trans. IP, Jan. 2007.
[15] J. Mairal, M. Elad, and G. Sapiro, “Sparse Representation for Color Image Restoration”, IEEE Trans. on Image Processing, Vol. 17, No. 1, pages 53-69, January 2008.
[16] Julien Mairal, Francis Bach, Jean Ponce and et. al., “Non-local Sparse Models for Image Restoration”, ICCV 2009.
[17] David Donoho, “Compressed sensing”. IEEE Trans. on IT, 52(4):1289-1306, Apr 2006.
[18] E. Cands, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information”. IEEE Trans. on IT, 52(2) pp. 489 - 509, Feb. 2006.
[19] David Donoho and Yaakov Tsaig, “Extensions of compressed sensing”. Signal Processing, 86(3), pp. 533-548, Mar. 2006.
[20] David L. Donoho, Yaakov Tsaig, Iddo Drori, and Jean-Luc Starck, “Sparse solution of underdetermined linear equations by stagewise orthogonal matching pursuit”. Preprint, 2007.
[21] D. Needell and J. A. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples”. Preprint, 2008.
[22] Y. C. Pati R. Rezaiifar, and P. S. Krishnaprasad, “Orthogonal Matching Pursuit: Recursive Function Approximation with Applications to Wavelet Decomposition”, Proceedings of the 27 th Annual Asilomar Conference on Signals, Systems, and Computers, Nov. 1993.
[23] S. Chen, , D. Donoho, and M. Saunders. “Atomic decomposition by basis pursuit”. Tech. Report 479, Dep. of Stat., Stanford Univ. May 1995.
[24] Tibshirani, R. (1996). “Regression shrinkage and selection via the lasso.”. J. Royal. Statist. Soc B. 58 (1): 267288.
[25] E. Candès and D. Donoho, “Curvelets – a surprisingly effective nonadaptive representation for objects with edges.” In: A. Cohen, C. Rabut and L. Schumaker, Editors, *Curves and Surface Fitting*: Saint-Malo 1999, Vanderbilt University Press, Nashville (2000), pp. 105–120.
[26] I. W. Selesnick, R. G. Baraniuk and N. G. Kingsbury, “The Dual-Tree Complex Wavelet Transform,” IEEE SP Magazine, 22(6):123-151, 2005.
[27] R. Eslami and H. Radha, “New Image Transforms using Hybrid Wavelets and Directional Filter Banks: Analysis and Design,” ICIP, Sep. 2005.
[28] M. Duarte, M. Davenport and et. al “Single-pixel imaging via compressive sampling.” IEEE SP Magazine, 25(2) pp. 83-91, March 08.
[29] http://sparselab.stanford.edu
[30] M. N. Do and M. Vetterli, “The contourlet transform: an efficient directional multiresolution image representation,” IEEE Trans. on IP, 14(12):2091-2106.
[31] Mtys A. Sustika, Joel A. Troppb, Inderjit S. Dhillona and Robert W. Heath Jr. “On the existence of equiangular tight frames’, preprint.
[32] A. A. Moghadam, Mohammad Aghagolzadeh, Hayder Radha, Mrityunjay Kumar, “Compressive Demosaicing”, IEEE Multimedia Signal Processing 2010.
[33] M.Elad, “Optimized projections for compressed sensing,” IEEE Trans. SP., 5(12):56955702, 2007.

[34] P. Nagesh and B. Li, “Compressive imaging of color images,” Proc. IEEE ICASSP, 2009.
[35] L. Zhang and X. Wu, “Color demosaicking via directional linear minimum mean square-error estimation”, IEEE Trans. Image Process., vol. 14, p.2167 , 2005.
[36] M. Aghagolzadeh, A. A. Moghadam, H. Radha, M. Kumar “Bayer and panchromatic color filter array demosaicing by sparse recovery”, SPIE, Digital Photography, 2011.
[37] M. Aghagolzadeh, A. A. Moghadam, H. Radha, M. Kumar “Incoherent Color Frames for Compressive Demosaicing”, IEEE ICASSP 2011.
[38] J. Mairal, F. Bach, J. Ponce, G. Sapiro and A. Zisserman. “Non-Local Sparse Models for Image Restoration”. International Conference on Computer Vision, Tokyo, Japan, 2009.
[39] A. Buades, B. Coll, and J. Morel. “A non-local algorithm for image denoising”. In Proc. IEEE CVPR, 2005.
[40] D.Donoho, A.Maleki, A.Montanari, “Message-passing algorithms for compressed sensing”, PNAS, 2009.
[41] W. Zhu, K. Parker, M.A. Kriss, “Color filter arrays based on mutually exclusive blue noise patterns”, J. Vis. Commun. Image Represent. 10:3(1999) 245267.
[42] J. Mairal, F. Bach, J. Ponce and G. Sapiro. “Online Dictionary Learning for Sparse Coding”. International Conference on Machine Learning, Montreal, Canada, 2009.
[43] Chang, Chih-Chung and Lin, Chih-Jen, “A library for support vector machines”, ACM Transactions on Intelligent Systems and Technology, Vol. 2, issue 3, 2011.
[44] http://egr.msu.edu/waves/people/nima/CD/