Color Sparse Representations for Image Processing: Review, Models, and Prospects — Source link ↗

Quentin Barthélemy, Anthony Larue, Jerome Mars

Institutions: University of Grenoble

Topics: Color histogram, Color image, Color balance, Color filter array and Color quantization

Related papers:

- Vector Sparse Representation of Color Image Using Quaternion Matrix Analysis

- Hypercomplex Fourier Transforms of Color Images

- Quaternion Structural Similarity: A New Quality Index for Color Images

- Quaternion Collaborative and Sparse Representation With Application to Color Face Recognition

- Color and Scale: The Spatial Structure of Color Images

Color Sparse Representations for Image Processing: Review, Models, and Prospects

Quentin Barthélemy, Anthony Larue, Jerome Mars

Color Sparse Representations for Image Processing: Review, Models and Prospects

Quentin Barthélemy, Anthony Larue and Jérôme I. Mars

Abstract—**Sparse representations have been extended to deal with color images composed of three channels. A review of dictionary-learning-based sparse representations for color images is made here, detailing the differences between the models, and comparing their results on real data and simulated data. These models are considered in a unifying framework that is based on the degrees of freedom of the linear filtering/transformation of the color channels. Moreover, this allows it to be shown that the scalar quaternionic linear model is equivalent to constrained matrix-based color filtering, which highlights the filtering implicitly applied through this model. Based on this reformulation, the new color filtering model is introduced, using unconstrained filters. In this model, spatial morphologies of color images are encoded by atoms, and colors are encoded by color filters. Color variability is no longer captured in increasing the dictionary size, but with color filters, this gives an efficient color representation.**

Index Terms—**Sparse representation; dictionary learning; color image; model; quaternion; color filter.**

I. INTRODUCTION

In image processing, sparse representations based on dictionary learning have become a widespread tool for grayscale images. In this framework, the redundant dictionary that is most adapted to a given dataset is estimated such that each datum is decomposed as a sparse linear sum of learned atoms [1]. Redundancy coupled with sparsity provides efficient representations for image processing [2]–[4], such as compression [5], denoising [6], [7], deblurring [2], inpainting [8], [9], demosaicking [7], super-resolution [10], [11], clustering [12], classification [13], [14], and others.

The question here is: how can we extend these representations to color images? A color image is composed of three color channels that are expressed in different spaces, such as RGB, YCbCr, HSL, Lab, XYZ, and other transformed color bases [15]. The classical RGB (*i.e.*, red, green, blue) color space is chosen as it respects the hypothesis of the spherical structure of the noise, which is required by sparse approximations [7]. It is also the simplest way to express natural images. An efficient color representation has to work on the whole color space to capture correlations between channels, encode the color diversity of the database without increasing the redundancy (*i.e.*, the dictionary size), and avoid color artifacts.

Q. Barthélemy is with Mensia Technologies, Institut du Cerveau et de la Moelle Epinière, iPEPS Hôpital de La Pitié-Salpêtrière, 75013 Paris, France (e-mail: q.barthelemy@gmail.com).

A. Larue is with CEA, LIST, 91191 Gif-sur-Yvette Cedex, France (e-mail: anthony.larue@cea.fr).

J.I. Mars is with Univ. Grenoble Alpes, GIPSA-Lab, 38000 Grenoble, France (e-mail: jerome.mars@gipsa-lab.grenoble-inp.fr).

For color representations, a naïve and classical approach consists of processing each color channel independently [6], although this approach cannot consider correlations between the color channels[1]. A first approach [7] that considers the whole color space consists of vectorizing the color channels. In this case, the dictionary contains many gray atoms and it introduces artifacts, such as lack of color saturation, washing effects, and hue bias. To improve the solution, adapting the dictionary on a specific image provides more colored atoms, but does not suppress color artifacts. Following in this way [17], [7], the scalar product is modified to enforce average colors, which improves the efficiency of the representation for denoising, inpainting and demoisacking. However, some color distortions and blurring effects persist. When adapted to three-dimensional (3D) structures, quaternions extend complex numbers using a real part and three imaginary parts. Quaternion-based methods have been shown to efficiently deal with color images [18]–[22], encoding the color channels into the three imaginary parts. Recently, sparse representations have been extended to quaternions [23]–[25], which have shown better performances for reconstruction [23], denoising [23], [25], inpainting [23] and super-resolution [24].

In this article, dictionary-learning-based sparse representations for color images are first reviewed, detailing the differences between the models, and comparing their results on real data and simulated data. These models are considered in a unifying framework that is based on the degrees of freedom of the linear filtering/transformation of the color channels. This allows it to be shown that the scalar quaternionic linear model is equivalent to constrained matrix-based color filtering, which highlights the filtering implicitly applied through this model. Based on this reformulation, a new model is introduced: the color filtering model using unconstrained filters. These reformulated models are compared, to show the differences with respect to their degrees of freedom. However, the complete study of this new model will be reported in a future article.

In Section II, quaternion algebra is presented, so as to understand the color representation based on the quaternionic model. In Section III, sparse approximation and dictionary learning problems are explained. The different models for color sparse representations are reviewed in Section IV, and in particular, the vectorized and quaternionic models. These are then compared with real and simulated data in Section V. In Section VI, these models are reformulated in a unifying framework in terms of the color filtering, and a new model is

---

[1]For the LabelMe database studied here [16], the median correlation coefficient is 0.97 between the red and green channels, 0.93 between the red and blue, and 0.97 between the blue and green.

proposed. These models are then compared in Section VII. In Section VIII, several directions for future research are given as prospects to improve the results obtained by color sparse representations.

II. QUATERNIONS

In this section, quaternion algebra and its vector representation are presented. For the notations, lower-case letters denote scalars, bold lower-case letters denote vectors, and bold upper-case letters denote matrices. Also, sans serif font is used for quaternions, to distinguish them from real numbers.

A. Quaternion algebra

Quaternion algebra is denoted as $\mathbb{H}$, and it is an extension of the complex space $\mathbb{C}$ using three imaginary parts [26], [27]. A quaternion $\mathsf{q} \in \mathbb{H}$ is defined as:

$$\mathsf{q} = q_0 + q_1 \mathsf{i} + q_2 \mathsf{j} + q_3 \mathsf{k} , \quad (1)$$

where $q_0, q_1, q_2, q_3 \in \mathbb{R}$, and the imaginary units are defined as:

$$\mathsf{ij} = \mathsf{k}, \ \mathsf{jk} = \mathsf{i}, \ \mathsf{ki} = \mathsf{j} \quad \text{and} \quad \mathsf{ijk} = \mathsf{i}^2 = \mathsf{j}^2 = \mathsf{k}^2 = -1 . \quad (2)$$

Quaternionic algebra is characterized by its noncommutativity: in general $\mathsf{qr} \neq \mathsf{rq}$, for $\mathsf{q}, \mathsf{r} \in \mathbb{H}$. The real part is $\Re(\mathsf{q}) = q_0$, and the imaginary part is $\Im(\mathsf{q}) = q_1 \mathsf{i} + q_2 \mathsf{j} + q_3 \mathsf{k}$. If its real part is null, a quaternion is said to be pure, and otherwise it is said to be full. The conjugate q^* is defined as: $\mathsf{q}^* = \Re(\mathsf{q}) - \Im(\mathsf{q})$ and we have $(\mathsf{qr})^* = \mathsf{r}^* \mathsf{q}^*$. The modulus is defined as $|\mathsf{q}| = \sqrt{\mathsf{q}^* \mathsf{q}}$, and the inverse as $\mathsf{q}^{-1} = \mathsf{q}^* / |\mathsf{q}|^2$. A unit quaternion is a quaternion with a modulus of one.

Moreover, an N-dimensional quaternionic vector is defined as $\mathbf{q} \in \mathbb{H}^N$:

$$\mathbf{q} = \boldsymbol{q}_0 + \boldsymbol{q}_1 \mathsf{i} + \boldsymbol{q}_2 \mathsf{j} + \boldsymbol{q}_3 \mathsf{k} , \quad (3)$$

where $\boldsymbol{q}_0, \boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3 \in \mathbb{R}^N$. In the following, $[\ \cdot \ , \ \cdot \ ]$ will denote the horizontal concatenation and $[\ \cdot \ ; \ \cdot \ ]$ the vertical concatenation. For vectors, $(.)^T$ denotes the transpose operator, and $(.)^H$ the conjugate transpose operator. For matrices, $\boldsymbol{Q}(:, n)$ and $\boldsymbol{Q}(m, :)$ denote the nth column and the mth row, respectively, of the matrix $\boldsymbol{Q}$.

B. Vector representation of quaternions

For any nonnull quaternion $\mathsf{x} \in \mathbb{H}$, the orthogonal matrix $\boldsymbol{U}(\mathsf{x}) \in \mathbb{R}^{4 \times 4}$ is defined as [26]:

$$\boldsymbol{U}(\mathsf{x}) = \frac{1}{|\mathsf{x}|} \begin{bmatrix} x_0 & x_1 & x_2 & x_3 \\ -x_1 & x_0 & -x_3 & x_2 \\ -x_2 & x_3 & x_0 & -x_1 \\ -x_3 & -x_2 & x_1 & x_0 \end{bmatrix} . \quad (4)$$

Using the vector representation of quaternion $\mathsf{q} \in \mathbb{H} \equiv [q_0, q_1, q_2, q_3] \in \mathbb{R}^4$, the product of two quaternions q and $\mathsf{x} \in \mathbb{H}$, with x being a unit quaternion, can be expressed in terms of the product of a vector and an orthogonal matrix:

$$\mathsf{qx} \equiv [q_0, \ q_1, \ q_2, \ q_3] \, \boldsymbol{U}(\mathsf{x}) . \quad (5)$$

This comes from expansion of the quaternion product:

$$\mathsf{qx} = (q_0 + q_1 \mathsf{i} + q_2 \mathsf{j} + q_3 \mathsf{k}) (x_0 + x_1 \mathsf{i} + x_2 \mathsf{j} + x_3 \mathsf{k}) \quad (6)$$

$$= q_0 x_0 - q_1 x_1 - q_2 x_2 - q_3 x_3$$
$$+ (q_0 x_1 + q_1 x_0 + q_2 x_3 - q_3 x_2) \, \mathsf{i}$$
$$+ (q_0 x_2 - q_1 x_3 + q_2 x_0 + q_3 x_1) \, \mathsf{j}$$
$$+ (q_0 x_3 + q_1 x_2 - q_2 x_1 + q_3 x_0) \, \mathsf{k} . \quad (7)$$

Now, considering a quaternion vector $\mathbf{q} \in \mathbb{H}^N \equiv [\boldsymbol{q}_0, \boldsymbol{q}_1, \boldsymbol{q}_2, \boldsymbol{q}_3] \in \mathbb{R}^{N \times 4}$ and a non-unit quaternion $\mathsf{x} \in \mathbb{H}$, the product becomes:

$$\mathbf{q}\mathsf{x} \equiv |\mathsf{x}| \, [\boldsymbol{q}_0, \ \boldsymbol{q}_1, \ \boldsymbol{q}_2, \ \boldsymbol{q}_3] \, \boldsymbol{U}(\mathsf{x}) . \quad (8)$$

This vector representation of this quaternionic product will be useful in the following.

III. DICTIONARY LEARNING PRINCIPLE

In this section, linear decomposition, sparse approximation, and dictionary-learning problems are presented.

A. Linear decomposition

Considering a signal $\boldsymbol{y} \in \mathbb{R}^N$ of N samples and a matrix $\boldsymbol{D} \in \mathbb{R}^{N \times M}$ of M vectors $\{\boldsymbol{d}_m\}_{m=1}^M$, the linear decomposition of the signal $\boldsymbol{y}$ is carried out on the matrix $\boldsymbol{D}$, such that:

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x} + \boldsymbol{e} = \sum_{m=1}^M \boldsymbol{d}_m x_m + \boldsymbol{e} , \quad (9)$$

where $\boldsymbol{x} \in \mathbb{R}^M$ are the coding coefficients, and $\boldsymbol{e} \in \mathbb{R}^N$ is the residual error. Moreover, the matrix $\boldsymbol{D}$ is normalized, which means that its columns are normalized, so that coefficients $\boldsymbol{x}$ reflect the energy of each vector in the signal.

Due to noncommutativity, two quaternionic linear models are possible [28]: coefficients can be multiplied on the left side of the vectors, or on their right side. With the definition of a quaternionic signal $\mathbf{y} \in \mathbb{H}^N$ and a dictionary $\mathbf{D} \in \mathbb{H}^{N \times M}$, the right-multiplication linear decomposition is:

$$\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{e} = \sum_{m=1}^M \mathbf{d}_m \mathsf{x}_m + \mathbf{e} , \quad (10)$$

where $\mathbf{x} \in \mathbb{H}^M$ is the coefficients vector, and $\mathbf{e} \in \mathbb{H}^N$ is the error. For this model, considering quaternionic vectors $\mathbf{q}, \mathbf{r} \in \mathbb{H}^N$, the right Hermitian scalar product is defined as $\langle \mathbf{q}, \mathbf{r} \rangle = \mathbf{r}^H \mathbf{q} \in \mathbb{H}$, and its associated ℓ_2 norm is denoted by $\|.\|_2$.

B. Sparse approximation

1) Generalities: When the matrix $\boldsymbol{D}$ of the linear model contains more vectors than the number of samples, *i.e.*, $M > N$, it is called a dictionary, and its normalized vectors are called atoms. The dictionary is thus a redundant/ overcomplete basis, and it provides a more efficient representation than a basis [1], [29], [30]: it is more robust to noise, it has more flexibility for matching patterns in the data, and it allows a more compact representation. As the dictionary is redundant, the linear system of Eq. (9) is thus underdetermined and has multiple possible solutions for $\boldsymbol{x}$. The introduction

of constraints, such as smoothness, positivity, sparsity, or others, allows the solution to be regularized. In particular, the decomposition of a signal $\boldsymbol{y}$ under a sparsity constraint can be formalized by [31]:

$$\min_{\boldsymbol{x}} \ \|\boldsymbol{y} - \boldsymbol{D}\boldsymbol{x}\|_2^2 \ \text{ s.t. } \|\boldsymbol{x}\|_s \leq K \ , \tag{11}$$

where $\|\boldsymbol{x}\|_s$ is the ℓ_0 pseudo-norm or the ℓ_1 norm, and K is a constant. Many algorithms deal with this problem for real signals: greedy optimizations such as matching pursuit [32] or orthogonal matching pursuit (OMP) [33], and also ℓ_1-norm based optimizations that ensure convex properties, such as basis pursuit [34], least-angle regression [35], iterative shrinkage-thresholding [36], and others (see [31] for a review).

2) OMP and Q-OMP: Orthogonal matching pursuit is a greedy sparse approximation that iteratively selects the atoms in an active dictionary that best explain the current residue [33]. As described in Algorithm 1, at the current iteration k, OMP selects the atom $\boldsymbol{d}_m$ that produces the strongest decrease in the residue. This is equivalent to selecting the atom that is the most correlated with the residue: the scalar products between the residue $\boldsymbol{e}^{k-1}$ and atoms $\boldsymbol{d}_m$ are computed (step 4). The selection (step 6) searches the maximum of their absolute values to determine the optimal atom $\boldsymbol{d}_{m^k}$. An active dictionary $\boldsymbol{D}^k \in \mathbb{R}^{N \times k}$ is formed, which contains all of the selected atoms (step 7). Active, *i.e.*, nonzero, coding coefficients $\boldsymbol{x}^k \in \mathbb{R}^k$ are computed via the orthogonal projection of $\boldsymbol{y}$ on $\boldsymbol{D}^k$ (step 8).

Algorithm 1 : $\boldsymbol{x} = \text{OMP}\,(\boldsymbol{y}, \boldsymbol{D})$

1: **initialization**: $k = 1$, $\boldsymbol{e}^0 = \boldsymbol{y}$, active dictionary $\boldsymbol{D}^0 = \varnothing$
2: **repeat**
3:     **for** $m \leftarrow 1, M$ **do**
4:         Scalar Product: $c_m^k \leftarrow \boldsymbol{d}_m^H \, \boldsymbol{e}^{k-1}$
5:     **end for**
6:     Selection : $m^k \leftarrow \arg\max_m \left| c_m^k \right|$
7:     Active Dictionary: $\boldsymbol{D}^k \leftarrow [\boldsymbol{D}^{k-1}, \boldsymbol{d}_{m^k}]$
8:     Active Coefficients: $\boldsymbol{x}^k \leftarrow \arg\min_{\boldsymbol{x}} \|\boldsymbol{y} - \boldsymbol{D}^k \boldsymbol{x}\|_2^2$
9:     Residue: $\boldsymbol{e}^k \leftarrow \boldsymbol{y} - \boldsymbol{D}^k \boldsymbol{y}^k$
10:    $k \leftarrow k + 1$
11: **until** stopping criterion

The first extension of the sparse approximation to quaternions is the quaternionic OMP (Q-OMP) [37], [28]. Due to the noncommutativity of quaternions, two Q-OMPs are detailed in [28] that depend on the multiplication side in the linear decomposition. The quaternionic coefficients are calculated by orthogonal projection of the signal on selected atoms, and they are computed recursively using block matrix inversion [37], [28]. Another QOMP version is also given in [23], [24], although with naïve implementation of the orthogonal projection, using the expensive quaternionic pseudo-inverse. Note that these algorithms are given for full quaternions, without constraint on the signal, atoms and residue during the coefficient estimations.

The Q-OMP algorithm is not detailed here as it is similar to Algorithm 1. There are two main differences: quaternions are noncommutative, so the order of the variables in multiplications is crucial (*e.g.*, the scalar product in step 4); and calculations are more expensive, as quaternions are composed of four real numbers. As shown in Eq. (7), the product of two quaternions contains 16 terms, whereas a complex one contains 4 terms, and a real one, only 1 term.

C. Dictionary learning

1) Generalities: The goal of dictionary learning is to empirically learn the dictionary that is the most adapted to a signals set, *i.e.*, where each signal of this set is sparsely approximated over the learned dictionary [1], [5], [38].

A training set of signals $\boldsymbol{\mathcal{Y}} = \left[\boldsymbol{y}_p\right]_{p=1}^P \in \mathbb{R}^{N \times P}$ is now considered (the index p is added to the other variables), and the associated coefficients are denoted by $\boldsymbol{\mathcal{X}} = \left[\boldsymbol{x}_p\right]_{p=1}^P \in \mathbb{R}^{M \times P}$. The admissible set of dictionaries is convex, and it is defined as $\mathcal{S}_D = \{\boldsymbol{D} \in \mathbb{R}^{N \times M} : \|\boldsymbol{d}_m\|_2 \leqslant 1, \ m = 1 \ldots M\}$. The dictionary learning problem can be formalized as:

$$\min_{\boldsymbol{\mathcal{X}}, \boldsymbol{D} \in \mathcal{S}_D} \ \|\boldsymbol{\mathcal{Y}} - \boldsymbol{D}\boldsymbol{\mathcal{X}}\|_F^2 \ \text{ s.t. } \|\boldsymbol{x}_p\|_s \leq K, p = 1 \ldots P \ . \tag{12}$$

This non-convex problem is usually solved by alternating between two steps: extraction of the main atoms (*i.e.*, the sparse approximation step) that are then learned (*i.e.*, the dictionary update step). This optimization scheme reduces the error criterion iteratively. There are several dictionary learning algorithms (DLAs) [1], such as maximum likelihood (ML) [29], method of optimal directions (MOD) [38], [39] and K-SVD [5] for batch methods, and online dictionary learning [40] and recursive least squares (RLS)-DLA [41] for online methods, which are less expensive in computational time and memory than batch methods.

2) K-SVD and K-QSVD: In the K-SVD method [5] described in Algorithm 2, all of the signals are decomposed with OMP (step 4) and the atoms are then updated one by one, fixing others when the considered atom is updated simultaneously with its associated coefficients:

$$\min_{\boldsymbol{\mathcal{X}}(m,:), \boldsymbol{d}_m} \ \|\boldsymbol{\mathcal{Y}} - \boldsymbol{D}^i \boldsymbol{\mathcal{X}}\|_F^2 = \|\boldsymbol{E}_m - \boldsymbol{d}_m \boldsymbol{\mathcal{X}}(m,:)\|_F^2 \ . \tag{13}$$

This is solved through computation of a truncated SVD of $\boldsymbol{E}_m = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V}^H$ (steps 8, 9), which gives a rank-1 approximation: the atom $\boldsymbol{d}_m^{i+1}$ is defined as the first column of $\boldsymbol{U}$ (step 10), and the coefficients $\boldsymbol{\mathcal{X}}(m,:)$ as the product between the highest singular value $\lambda_1 = \boldsymbol{\Lambda}(1,1)$ and the first row of $\boldsymbol{V}^H$ (step 11). To keep the sparsity of coefficients, only nonzero coefficients are used in these steps, which are selected by the active set $\Gamma_m = \{p = 1 \ldots P : \boldsymbol{\mathcal{X}}(m,p) \neq 0\}$ (step 7).

Using quaternionic sparse approximation, some DLAs have been extended to quaternions: K-SVD has been extended to K-QSVD [23], [25] and online learning [24].

The K-QSVD algorithm is not detailed here as it is very close to Algorithm 2. The main differences are the use of Q-OMP instead of OMP for sparse approximation (step 4), and the use of Q-SVD [42] instead of SVD for the dictionary update (step 9). As noted previously in Section III-B2, the variables order is crucial (as for the coefficients update, in step 11) and calculations are more expensive.

Algorithm 2 : $\boldsymbol{D} = \text{K-SVD} \left(\boldsymbol{\mathcal{Y}} = [\boldsymbol{y}_p]_{p=1}^P \right)$

1: **initialization**: $i = 0$; $\boldsymbol{D}^0 = [M \text{ white noise atoms}]$
2: **repeat**
3:     **for** $p \leftarrow 1, P$ **do**
4:         Sparse approximation: $\boldsymbol{x}_p = \text{OMP} \left(\boldsymbol{y}_p, \boldsymbol{D}^i \right)$
5:     **end for**
6:     **for** $m \leftarrow 1, M$ **do**   Dictionary update
7:         Active set: $\Gamma_m = \{p = 1 \ldots P : \boldsymbol{\mathcal{X}}(m, p) \neq 0\}$
8:         Residue: $\boldsymbol{E}_m = \boldsymbol{\mathcal{Y}} - \sum_{\nu \neq m} \boldsymbol{d}_\nu^i \boldsymbol{\mathcal{X}}(\nu, :)$
9:         SVD: $(\boldsymbol{U}, \boldsymbol{\Lambda}, \boldsymbol{V}) = \text{SVD} \left(\boldsymbol{E}_m(:, p)|_{p \in \Gamma_m} \right)$
10:       Atom update: $\boldsymbol{d}_m^{i+1} = \boldsymbol{U}(:, 1)$
11:       Coefficients update: $\boldsymbol{\mathcal{X}}(m, p)|_{p \in \Gamma_m} = \lambda_1 \boldsymbol{V}^H(1, :)$
12:     **end for**
13:     $i = i + 1$
14: **until** stopping criterion

IV. MODELS FOR COLOR IMAGES

In this section, linear models for color images are reviewed. A color image patch of $\sqrt{N}$ by $\sqrt{N}$ pixels is filled in three N-dimensional vectors. Its RGB channels are called $\boldsymbol{y}_c \in \mathbb{R}^N$, where $c = \{r, g, b\}$ is defined for red, green and blue, respectively. This image is decomposed on a dictionary composed of three channels that are denoted $\boldsymbol{D}_c \in \mathbb{R}^{N \times M}$, where $c = \{r, g, b\}$.

As previously said, a naïve model consists of treating the three RGB channels separately as [10], [43], [44]:

$$\boldsymbol{y}_c = \boldsymbol{D}_c \boldsymbol{x}_c + \boldsymbol{e}_c , \quad c = \{r, g, b\} , \tag{14}$$

with the coefficients $\boldsymbol{x}_c \in \mathbb{R}^M$ and the residue $\boldsymbol{e}_c \in \mathbb{R}^N$. This model, which treats channels independently as gray images, does not work on the whole color space, and it cannot capture correlations between channels.

A. Vectorized model

In the vectorized model [7], the RGB channels are concatenated as $\boldsymbol{y} = [\boldsymbol{y}_r; \boldsymbol{y}_g; \boldsymbol{y}_b] \in \mathbb{R}^{3N}$ and $\boldsymbol{D} = [\boldsymbol{D}_r; \boldsymbol{D}_g; \boldsymbol{D}_b] \in \mathbb{R}^{3N \times M}$. The coding coefficients $\boldsymbol{x} \in \mathbb{R}^M$ are now common to the three vectorized channels:

$$\boldsymbol{y} = \boldsymbol{D}\boldsymbol{x} + \boldsymbol{e} = \sum_{m=1}^M \boldsymbol{d}_m x_m + \boldsymbol{e} , \tag{15}$$

with the residue $\boldsymbol{e} = [\boldsymbol{e}_r; \boldsymbol{e}_g; \boldsymbol{e}_b] \in \mathbb{R}^{3N}$. Using K-SVD on the vectorized data, this approach gives better results than the previous approach, although processed patterns tend to be mono-chromatic following the gray axis ($r = g = b$) [45], [7]. Gray atoms introduce artifacts, such as lack of color saturation, washing effects, and hue bias. To improve the solution, adapting the dictionary to a specific image provides more colored atoms, but does not suppress color artifacts.

To overcome these color artifacts, the scalar product is modified to enforce average colors, improving the K-SVD [17], [7]. For this purpose, the sparse approximation step is modified, transforming the color patches, such as:

$$\boldsymbol{y}_c \leftarrow \boldsymbol{y}_c + a\, \bar{\boldsymbol{y}}_c , \quad c = \{r, g, b\} , \tag{16}$$

where $a = 1.5$ is an empirically fixed parameter, and $\bar{\boldsymbol{y}}_c$ is the spatial average of channel c. Atoms are transformed too, and a normalization factor is taken into account during atom selection (Algorithm 1, step 6). This transformation is equivalent to spatial filtering, and it provides a more colored learned dictionary, which gives good results for denoising, inpainting and demosaicking of color images [7]. This color enforcement can increase low frequencies, which generates blurring effects, and some color distortions persist.

B. Quaternionic model

In the literature, many studies have shown that quaternions are adapted to color images [18]–[22], [46], [47]. Quaternions produce good results, as they take into account the structure between the color channels. The RGB channels are filled in pure quaternions. The color patch is now defined as $\mathbf{y} = 0 + \boldsymbol{y}_r \mathsf{i} + \boldsymbol{y}_g \mathsf{j} + \boldsymbol{y}_b \mathsf{k} \in \mathbb{H}^N$, and the dictionary as $\mathbf{D} = 0 + \boldsymbol{D}_r \mathsf{i} + \boldsymbol{D}_g \mathsf{j} + \boldsymbol{D}_b \mathsf{k} \in \mathbb{H}^{N \times M}$. The linear model is:

$$\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{e} = \sum_{m=1}^M \mathbf{d}_m \mathsf{x}_m + \mathbf{e} , \tag{17}$$

where $\mathbf{x} \in \mathbb{H}^M$ is the full quaternionic coefficients vector, and $\mathbf{e} = \boldsymbol{e}_0 + \boldsymbol{e}_r \mathsf{i} + \boldsymbol{e}_g \mathsf{j} + \boldsymbol{e}_b \mathsf{k} \in \mathbb{H}^N$ is the full quaternionic residue. This model gives excellent results for color image processing [20], [23], and it reduces blurring effects and color distortions using a quaternionic extension of the K-SVD, called the K-QSVD [23]. It also has no empirically fixed parameter.

However, the main drawback is in the processing of the real part of the quaternionic data. Although the image $\mathbf{y}$ and the atoms $\mathbf{D}$ are pure, the obtained sparse approximation $\hat{\mathbf{y}} = \mathbf{D}\mathbf{x}$ is not necessarily pure, and even this would be convenient. Its real part has to be reset, $\Re(\hat{\mathbf{y}}) \leftarrow \mathbf{0}$, for some tasks such as reconstruction and denoising, which increases the residue. In the same way, during the dictionary learning iterations, the atoms are not necessarily pure after their updates (Algorithm 2, step 10). For each updated atom, its real part has to be reset to ensure a pure atom [23], [24]: $\Re(\mathbf{d}_m) \leftarrow \mathbf{0}$, and it is then re-normalized. Despite this drawback, the quaternionic model gives better performances for tasks such as reconstruction [23], denoising [23], [25], inpainting [23], and super-resolution [24].

The difference between these two models is illustrated in Fig. 1: the vectorized model of Eq. (15) in Fig. 1(a), and the quaternionic model of Eq. (17) in Fig. 1(b).
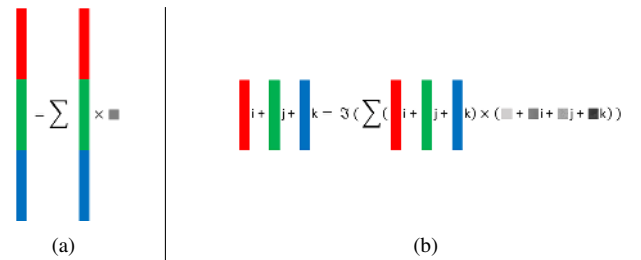

Fig. 1. Illustration of the color models: (a) is the vectorized model of Eq. (15), and (b) is the quaternionic model of Eq. (17).

V. MODELS COMPARISON

After this review, the existing color models are applied to real data, and then to simulated data. The corresponding DLAs derived from the same learning process K-SVD are compared.

A. Comparison of learned dictionaries

In this experiment, the dictionaries learned by K-SVD [5], *i.e.*, the improved K-SVD [7] (iK-SVD), and K-QSVD [23] are compared[2]. DLAs are applied to the LabelMe database [16], and $P = 2500$ patches are randomly selected to form the training database. DLAs learn dictionaries composed of $M = 256$ atoms, and the size patch is fixed to 7, which gives atoms of $N = 7 \times 7 = 49$ pixels. Each dictionary is trained with $I = 100$ learning iterations, a sparsity $K = 6$ for sparse approximation, and an initial dictionary composed of randomly selected training patches.

Learned atoms are normalized, although they can contain negative values due to the learning process. To visualize these, a shift and a rescale are required. Considering an atom d, we have:

$$d \leftarrow (d - d_{min})/(d_{max} - d_{min}) , \qquad (18)$$

and this is finally multiplied by 255. This shift and rescale can be applied channel-wise (on each channel d_c, independently) as in [7], [23], or globally (on the three channels simultaneously). The channel-wise transformation does not preserve the color distributions of the original atom as it equalizes the contrast in the three channels. This tends to center and thicken the distributions, thus providing atoms following the gray axis. The two types of visualization can be useful: the channel-wise allows focus on the spatial structures; and the global one, on colors.

The dictionary learned with K-SVD, iK-SVD and K-QSVD are visualized in Fig. 2, with the two types of rescales: the channel-wise shift and rescale (top), and the global rescale (bottom). All dictionaries contain similar spatial morphologies, *i.e.*, atoms with edges and others with low-frequencies and smooth textures. However, their color contents are different. The dictionary learned by the vectorized K-SVD is shown in Fig. 2 (left column). In the channel-wise rescale (Fig. 2 left top), there are many color-less atoms, dedicated to the reconstruction of spatial structures. Colored atoms contain opposite colors, such as yellow-blue or green-magenta, which is a phenomenon that was already observed in [45]: atoms tend to follow the diagonals of the RGB cube. Achromatic atoms disappear with the global rescale (Fig. 2 left bottom). The dictionary learned by improved vectorized K-SVD and iK-SVD is shown in Fig. 2 (middle column). In the channel-wise rescale (Fig. 2 middle top), iK-SVD atoms are more colored than K-SVD atoms. This is obtained by the color average enforcement of Eq. (16). However, this enforcement can increase low-frequencies [7], and thus generate blurring effects in atoms. Average enforcement also tends to produce similar distributions in the channels of the iK-SVD atoms, which explains that both visualizations (Fig. 2 top, bottom) are

[2]Quaternionic algorithms are coded with the Quaternion toolbox for Matlab (QTFM) [48].
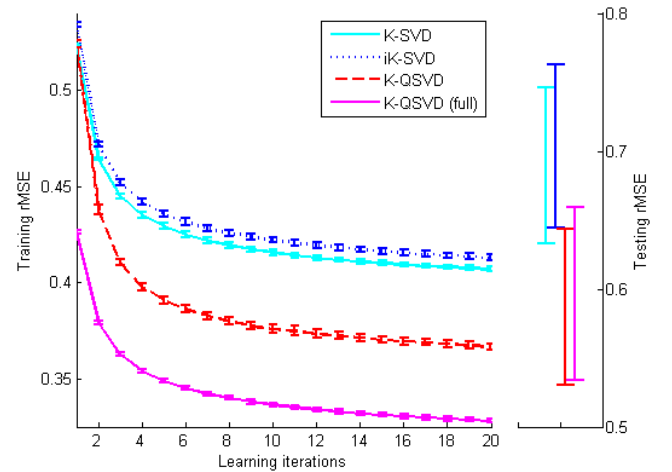

Fig. 3. Left: averaged training fitting errors as a function of learning iterations, for K-SVD (cyan, solid line), iK-SVD (blue, dotted line), K-QSVD (red, dashed line) and K-QSVD (full) with full quaternionic atoms (magenta, solid line). Right: averaged testing the fitting errors to evaluate over-fitting.

similar, as both rescales have similar effects. The dictionary learned by K-QSVD is shown in Fig. 2 (right column). For both visualizations, this is the most colored dictionary with respect to the previous ones. It encodes spatial morphologies and diversified colors. So, we can assume that the quaternionic model captures more colors than the vectorized ones.

B. Comparison of fitting errors

Another useful evaluation is to compare the evolution of the fitting errors during the learning process. The optimization criterion of Eq. (12) is slightly changed in:

$$\|\mathcal{Y} - \boldsymbol{D}\mathcal{X}\|_F^2 / \|\mathcal{Y}\|_F^2 \qquad (19)$$

This thus gives the relative mean square error (rMSE). This allows measurement of the efficiency of the color representation. Due to the random initialization of the atoms on image patches, and the random order in the atoms update, the experiment is averaged on 15 learnings. The average and standard deviation of the rMSE are plotted in Fig. 3, as a function of the learning iterations (the iterations are stopped at 20, as there is no supplementary information in the following iterations). The settings of the previous experiment are retained, except that the training set is now composed of $P = 1000$ color patches, randomly generated. To evaluate over-fitting, averaged rMSE is also computed on a distinct testing set composed of 1000 random patches.

The K-SVD has two specificities: it is batch optimization, and the atoms updated by SVD are already normalized, which avoids a re-normalization step that generally generates fluctuations in the criterion minimization. If the strict decrease in the MSE at each iteration of K-SVD is observed empirically, the theoretical proof is not available, due to the nonconvexity of the sparse approximation step carried out using ℓ_0-pursuit algorithm [5], [49].

In Fig. 3, the smooth curve of K-SVD shows this property, and low errors underline the efficiency of this color representation. For iK-SVD, this is the same model as K-SVD, but with
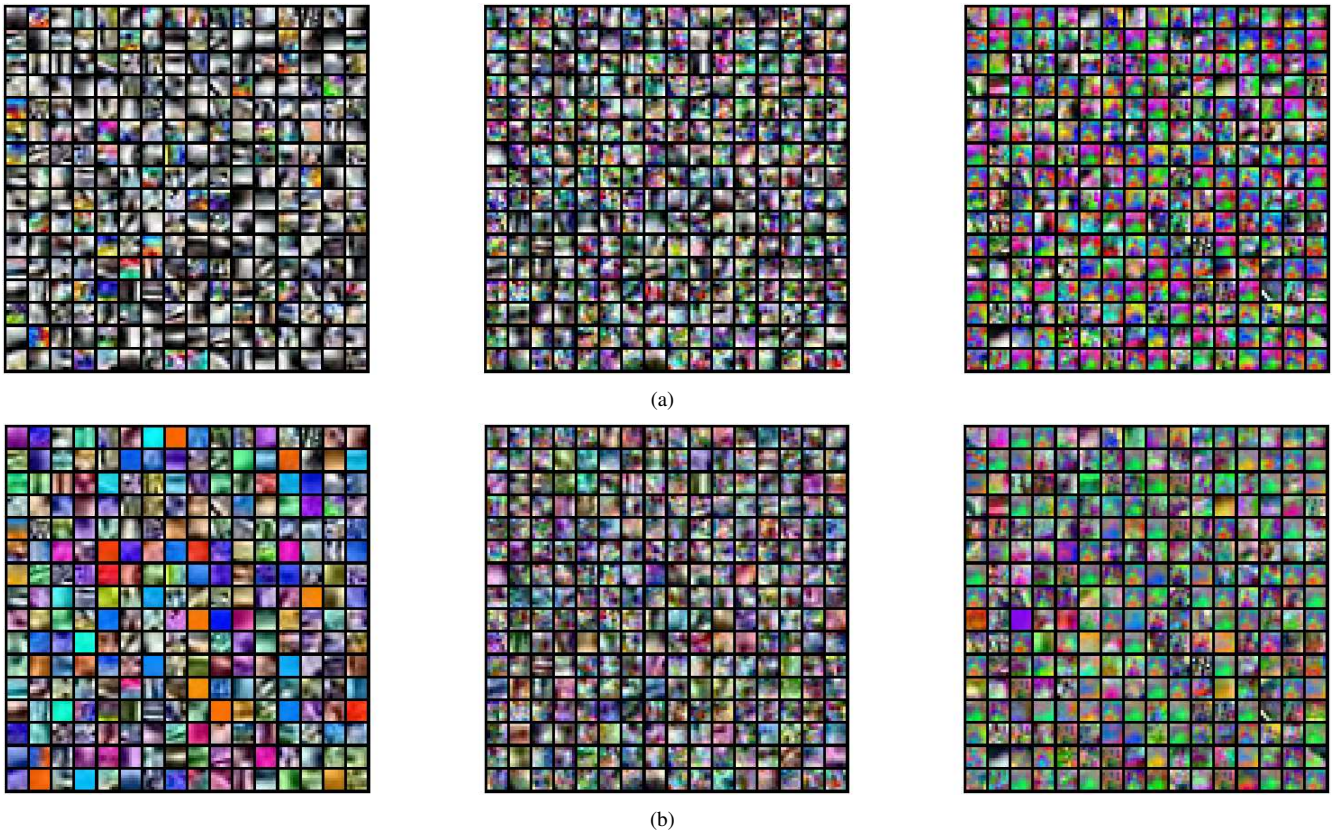
(a)

(b)

Fig. 2. Visualization of the color image dictionaries learned by K-SVD (left column), improved K-SVD (middle), and K-QSVD (right), with the channel-wise shift and rescaling (a), and the global shift (b). Better colors can be seen on a computer screen.

a modified criterion during the sparse approximation step. This additional constraint provides a more colored dictionary, but explains why its global learning criterion is slightly worse than the original K-SVD. K-QSVD is used with full quaternionic atoms (Algorithm 2), denoted as K-QSVD (full), it has the best criterion evolution. However, its atoms have a nonnull real part that is difficult to interpret with respect to color images. For K-QSVD, the reset of the real part of the atoms and their re-normalization (Algorithm 2 with modifications given in Section IV-B) are the unique difference from K-QSVD (full), thus explaining the important difference between their respective criteria. Moreover, this additional step generates fluctuations in the reduction of rMSE, indicated by higher deviations.

As usual in machine learning, the testing fitting errors are higher than the training ones. On the right part of Fig. 3, the testing rMSE increase is similar for the different methods. Consequently, if the quaternionic models are more complex than the vectorized ones, they do not over-fit the training data. This experiment has shown the representative power of the different models, with a clear advantage for the quaternionic model. However, a slight over-fitting could be observed for K-SVD (full): since patches are pure quaternions, the real parts of atoms learned on the training set, and which help to decrease the training rMSE, then increase the rMSE on the testing set.

To conclude, these experiments demonstrate the efficiency of the quaternionic model over the vectorized models. This is correlated to the results obtained in color image processing,

such as reconstruction [23], denoising [23], [25], inpainting [23] and super-resolution [24].

VI. UNIFYING FRAMEWORK FOR COLOR MODELS

One main objective of this article is to reformulate existing models in a unifying framework, in order to shed new light on color models, and a new model is proposed. For this reformulation, the color channels are not vertically vectorized any more, but horizontally concatenated in a matrix.

A. Reformulated vectorized model

The color patch is now defined as a matrix $\boldsymbol{Y} = [\boldsymbol{y}_r, \boldsymbol{y}_g, \boldsymbol{y}_b] \in \mathbb{R}^{N \times 3}$. Similarly, each atom is defined as $\boldsymbol{D}_m = [\boldsymbol{d}_{m,r}, \boldsymbol{d}_{m,g}, \boldsymbol{d}_{m,b}] \in \mathbb{R}^{N \times 3}$. The vectorized model in Eq. (15) becomes:

$$\boldsymbol{Y} = \sum_{m=1}^{M} x_m \, \boldsymbol{D}_m + \boldsymbol{E} \;, \tag{20}$$

with the residue $\boldsymbol{E} = [\boldsymbol{e}_r, \boldsymbol{e}_g, \boldsymbol{e}_b] \in \mathbb{R}^{N \times 3}$. This basic reshape of the decomposition model allows its unique degree of freedom to be shown: the scaling factor x_m.

B. Reformulated quaternionic model

The quaternionic model of Eq. (17) ignores the real part of the product $\boldsymbol{D}\mathbf{x}$. So, only the imaginary part of the product
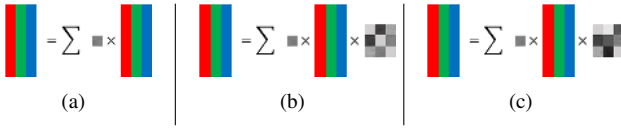
Fig. 4. Illustration of the reformulated color models: (a) is the vectorized model of Eq. (20), (b) is the quaternionic model of Eq. (23) and (c) is the the color filter model of Eq. (24).

$\mathbf{d}_m \mathsf{x}_m$ is studied. The matrix $\boldsymbol{V}(\mathsf{x}) \in \mathbb{R}^{3\times3}$ is defined as the bottom right part of matrix $\boldsymbol{U}(\mathsf{x})$:

$$\boldsymbol{V}(\mathsf{x}) = \frac{1}{|\mathsf{x}|} \begin{bmatrix} x_0 & -x_3 & x_2 \\ x_3 & x_0 & -x_1 \\ -x_2 & x_1 & x_0 \end{bmatrix} , \qquad (21)$$

and the imaginary part of the product $\mathbf{d}\mathsf{x}$ is considered, with the pure atom $\mathbf{d} \equiv [\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3] \in \mathbb{R}^{N\times3}$. Using Eq. (8), we thus have:

$$\Im(\mathbf{d}\mathsf{x}) \equiv |\mathsf{x}| \, [\boldsymbol{d}_1, \ \boldsymbol{d}_2, \ \boldsymbol{d}_3] \, \boldsymbol{V}(\mathsf{x}) . \qquad (22)$$

For each atom, defining $\alpha_m = |\mathsf{x}_m|$ and $\boldsymbol{V}_m = \boldsymbol{V}(\mathsf{x}_m)$ as the quaternionic model in Eq. (17), this becomes:

$$\boldsymbol{Y} = \sum_{m=1}^{M} \alpha_m \, \boldsymbol{D}_m \, \boldsymbol{V}_m + \boldsymbol{E} . \qquad (23)$$

Matrix $\boldsymbol{V}_m$ applies a linear transformation/combination between the color channels. This reformulation shows that this model adds color transformation due to $\boldsymbol{V}_m$, which we will call the color filter. Consequently, this model has four degrees of freedom: one for the scaling factor, and three for the color filter (four variables in matrix $\boldsymbol{V}_m$, less the normalization constraint). This reformulation, which highlights the implicit color filtering applied through the scalar quaternionic linear model, explains why quaternions are so efficient for color image processing in general. This linear transformation of the channels enables relations and interactions between the channels, which allows the color channel correlations to be captured. We note that the vectorized model of Eq. (20) is recovered, with $\boldsymbol{V}_m = \boldsymbol{I}_3$ if $\mathsf{x}_m > 0$, and with $-\boldsymbol{I}_3$ otherwise.

Note that quaternion-based color filters have already been studied in [46]. However, they were derived from equations that were linear with respect to the image vector, but not with respect to the coefficients. Consequently, these color filters are not adapted for the sparse linear model of Eq. (17).

C. Color filtering model

Pursuing this idea of filtering matrix, the generalized multivariate model [50] is considered here. Each multivariate atom $\boldsymbol{D}_m$ is linearly transformed by an unconstrained matrix $\boldsymbol{W}_m \in \mathbb{R}^{3\times3}$:

$$\boldsymbol{Y} = \sum_{m=1}^{M} \alpha_m \, \boldsymbol{D}_m \, \boldsymbol{W}_m + \boldsymbol{E} . \qquad (24)$$

This model has nine degrees of freedom: one for the scaling factor, and eight for the color filter $\boldsymbol{W}_m$ (nine variables, less a normalization constraint).

In the reformulated quaternionic model of Eq. (23), the channels of $\boldsymbol{Y}$ are constrained linear transformations of atoms: Eq. (21) shows that the matrices $\boldsymbol{V}_m$ have a constrained structure that is inherited from the quaternion algebra. In the current model, matrices $\boldsymbol{W}_m$ have no more constraints: they are data-driven estimated, and they can generate colors that cannot be generated by the quaternionic model. Indeed, this model is invariant to linear transformations/combinations. This affine-invariance allows to span all the column space of the atom $\boldsymbol{D}_m$, contrary to the quaternionic model. Adding this color filter, each atom $\boldsymbol{D}_m$ can be replicated in a huge family of linearly transformed color patterns $\boldsymbol{D}_m \boldsymbol{W}_m$. Consequently, this model allows the color image analysis to be separated into two parts: the spatial morphology of images is encoded by atoms $\boldsymbol{D}_m$, and the color diversity is encoded by the color filters $\boldsymbol{W}_m$.

The difference between these three reformulated models is illustrated in Fig. 4: the vectorized model of Eq. (20) in Fig. 4(a), the quaternionic model of Eq. (23) in Fig. 4(b), and the color filter model of Eq. (24) in Fig. 4(c).

Concerning the implementation, a supplementary advantage of this model is that the data are always defined in $\mathbb{R}^{N\times3}$: there are no more problems to ensure atoms are pure quaternions, resetting their real parts. However, a new constraint has to be added. In Eq. (12), the atoms are normalized to avoid the uncertainty of the product $\boldsymbol{D}\mathcal{X}$. Without constraint on the atom norm, if $\{\mathcal{X}, \boldsymbol{D}\}$ is a solution, $\{\lambda\mathcal{X}, \frac{1}{\lambda}\boldsymbol{D}\}$ is also a solution for all nonnull scalar $\lambda \in \mathbb{R}$. As a new matrix has been added in the color filtering model of Eq. (24), this introduces a new uncertainty. Consequently, a normalization is required for the matrices $\boldsymbol{W}_m$. Among the possibilities, the normalization such that $\|\boldsymbol{D}_m \boldsymbol{W}_m\|_F = \|\boldsymbol{D}_m\|_F = 1$ keeps the atom norm equal after linear transformation. This normalization appears to be an appropriate choice, as it has been used efficiently for matrices $\boldsymbol{W}_m$ restricted to rotations [51].

VII. COMPARISON OF THE REFORMULATED MODELS

In this section, the three reformulated models are compared, showing the differences concerning their degrees of freedom. To fully understand, this is first illustrated in the spatial space, and then in the color space.

A. Illustration in spatial space

To more easily understand the differences between the linear transformations, the model differences are illustrated with a 3D pattern in a $\mathsf{x}, \mathsf{y}, \mathsf{z}$ spatial space. The considered pattern is a cube composed of eight points, living in $\mathbb{R}^{8\times3}$.

First, the linear transformation $\boldsymbol{V}(\mathsf{x})$ given in Eq. (21) is characterized. General linear transformations include isotropic and anisotropic scalings, rotations, reflections and shearings [52]. The linear transformation $\boldsymbol{V}(\mathsf{x})$ is composed of an isotropic scaling of $x_0/|\mathsf{x}|$. Once this scaling is set apart, $\boldsymbol{V}$ can be seen as a shearing transformation [52]:

$$\boldsymbol{S} = \begin{bmatrix} 1 & sh_{\mathsf{xy}} & sh_{\mathsf{xz}} \\ sh_{\mathsf{yx}} & 1 & sh_{\mathsf{yz}} \\ sh_{\mathsf{zx}} & sh_{\mathsf{zy}} & 1 \end{bmatrix} , \qquad (25)$$
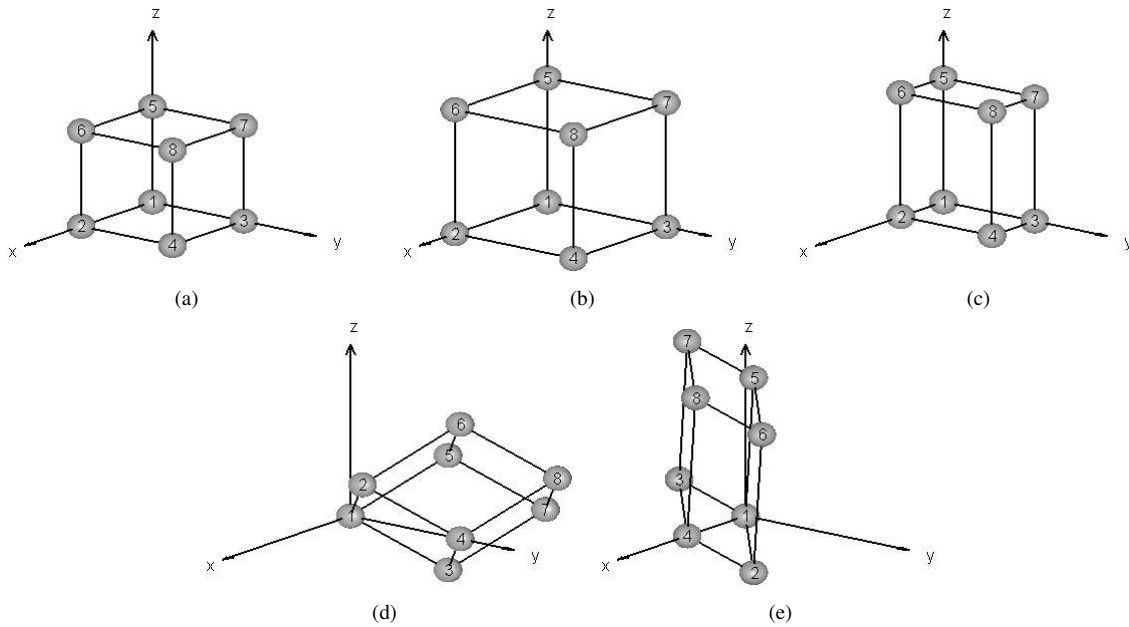
Fig. 5. Illustration of 3D linear transformations in x, y, z spatial space: original atom (a), isotropic scaling (b), anisotropic scaling (c), anti-symmetric shearing (d), and general transformation (e).

where sh_{xy} is the coefficient that shears y into x. In the spatial domain, the shearing transforms a cube into a parallelepipoid. Consequently, matrix V given in Eq. (21) is a scaled shearing transformation with anti-symmetric off-diagonal coefficients, thus generating particular proportions in the parallelepipoid.

Fig. 5 illustrates the effects of the different linear transformations on the original cube of Fig. 5(a), with numbered vertices. The isotropic scaling of Fig. 5(b) corresponds to a transformation provided by the vectorized model of Eq. (20). An anisotropic scaling is represented in Fig. 5(c). The anti-symmetric shearing of Fig. 5(d) corresponds to a transformation that is provided by the quaternionic model of Eq. (23). The general transform of Fig. 5(e) corresponds to a transformation provided by the color filtering model of Eq. (24). In the example of Fig. 5(e), the transformation is an anisotropically scaled and sheared reflection (the order of the vertices is inversed, see numbers), an example of a transformation that cannot be reached by the quaternionic model of Eq. (23).

This illustration in the spatial space has allowed the differences between the models to be more easily seen.

B. From spatial space to color space

For the interpretation of such linear transformations, it is possible to carry out a comparison between the x, y, z axis in spatial space and the RGB axis in color space. The considered pattern consists of 10 equally spaced values along each of the 12 edges of the cube of the previous experiment. These 3D values define the color pixels in the RGB color cube [46], thus giving a color image.

Keeping the same linear transformations as the previous experiment, these are applied on this color image. Fig. 6 shows the effects of the transformations on the 3D RGB cube (top) and on the associated color image (bottom).

C. Illustration in color space

The differences are now illustrated in the color space with a real image. In the previous experiment of Section V-A, the visualized atoms were learned from a database: different models generate different dictionaries. Here, considering a same atom, the patterns generated by the degrees of freedom of each model are illustrated. A subpart of the image Lena is considered, and is coded over 8 bits, from 0 to 255. This sub image is then normalized to create an atom.

For the vectorized model with one degree of freedom, scalar multiplication is applied to the atom. Three scaling factors are chosen: $x = 255$, 180 and 105. The results here are illustrated in Fig. 7(a)-(c). For $x = 255$, the original image is recovered, and for $x = 1$, the original atom is visually black. Intermediary factors are proportional to the luminous intensity. In Fig. 7(d), each atom pixel is represented as a color point in a 3D RGB color cube, to show the distribution of the pixels in the color space. For the quaternionic model with four degrees of freedom, a scalar multiplication is applied to the atom as well as the color filter V given in Eq. (21). The three previous scalars are kept: $\alpha = 255$, 180 and 105, and two filters V are randomly generated. After a global shift and rescale, the resulting patterns are plotted in Fig. 7(e)-(g) for the first color filter, and in Fig. 7(i)-(k) for the second color filter. For the filtering model with nine degrees of freedom, scalar multiplication is applied to the atom as well as an unconstrained color filter W. The three previous scalars are kept: $\alpha = 255$, 180 and 105, and two filters W are randomly generated. After a global shift and rescale, the resulting patterns are plotted in Fig. 7(m)-(o) for the first color filter, and in Fig. 7(q)-(s) for the second color filter.

An infinity of color filters could be generated to illustrate the differences between the degrees of freedom of the models. Contrary to the spatial space, it is less easy to visualize the
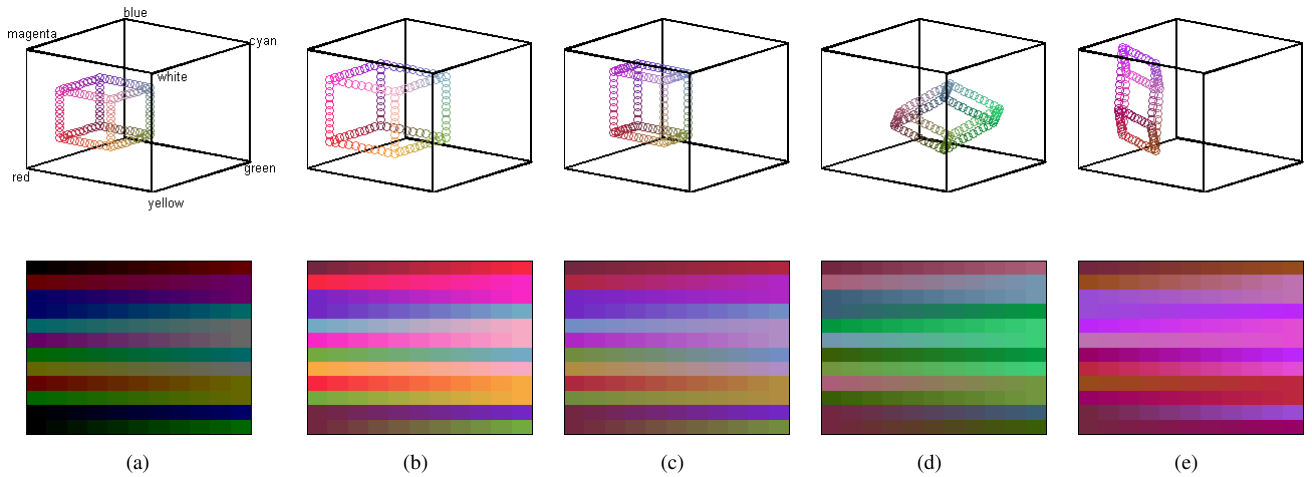
Fig. 6. Illustration of the linear transformations in the RGB color space, with the same transformations as used in Fig. 5: original image (a), isotropic scaling (b), anisotropic scaling (c), anti-symmetric shearing (d), and general transformation (e). Pixel values in the 3D RGB cube (top) and the associated color image (bottom).

improvement in the color space that is due to the nine degrees of freedom of the color filtering model with respect to the four degrees of freedom of the model quaternionic. However, the RGB color cubes of Fig. 7 show that the color filter model can apply transformations less usual than the quaternionic model, and obviously, than the vectorized one. With color filters, a unique atom $\boldsymbol{D}_m$ can be replicated in a highly redundant collection of linearly filtered color patterns $\boldsymbol{D}_m\boldsymbol{W}_m$.

To conclude, this section has highlighted the advantage of the color filter model over the existing models.

VIII. PROSPECTS

For color image processing based on sparse representation, many open questions remain. Several research directions are listed in this section.

A. Prospects relating to quaternions

Recent studies have dealt with the quaternionic model [23]–[25], although many problems have not been studied yet, such as:

1) For sparse approximations, to our knowledge, Q-OMP is currently the unique existing algorithm. Methods based on the ℓ_1-norm with convex properties [31] can be extended to quaternions. For dictionary learning, the different quaternionic DLAs can be compared, with the adaptation of classical comparison criteria to quaternions [53]. Theoretical aspects for sparse approximation [54], [55] and dictionary learning [49], [56], [57] can be extended to quaternions too.

2) Contrary to the actual reset of the real part and renormalization during the dictionary learning [23], [24], dedicated optimizations that ensure atoms are pure quaternionic vectors can be useful, to avoid loss of performance and fluctuations in the learning optimization. However, where the coefficients are currently estimated without constraint, sparse approximations can be improved for this purpose too. Note that in [25], the atoms were tolerated to be full. What is the impact of such an approach? Is there a risk of over-fitting?

3) Optimization of the algorithm implementations is a major issue. Having rapid implementations is crucial for the processing of large databases, as in sparse approximation and dictionary learning, and more generally in machine learning. This is especially because the calculus is already heavy for quaternions, as noted in Section III-B2. This has a critical impact during the quaternionic matrix calculus.

To illustrate this, the two available implementations of the Q-OMP are compared here in a short experiment: the recursive Q-OMP [28], [37] and the naïve Q-OMP [23], [24]. The vectorized OMP [33] is also used as the baseline. The computational times of the decompositions are evaluated for different database sizes[3]. The results are averaged five times and are plotted in Fig. 8 for different values of signal length N and signal number P. Performance differences between the Q-OMP implementations are clearly highlighted in Fig. 8. The recursive implementation outperforms the naïve implementation for small parameters, and the difference is greater for larger parameters.

B. Prospects relating to color filters

As this model is new, all aspects have to be studied.

1) First of all, as mentioned in Section VI-C, a new constraint has to be added for matrices $\boldsymbol{W}_m$. If the normalization of each product $\boldsymbol{D}_m\boldsymbol{W}_m$ appears appropriate, other possibilities exist, such as the normalization of each matrix $\boldsymbol{W}_m$, or any normalization. Which is the most suitable, theoretically and empirically?

2) The sparse approximations dedicated to this model have to jointly estimate coefficients and color filters. Multiple optimizations are conceivable: should these be simultaneous or alternating estimations? Similarly, associated DLAs have to

---

[3] A dictionary of $\mathbf{D} \in \mathbb{H}^{N \times M}$ of $M = 2N$ atoms with N samples is generated from white uniform noise and is then normalized. Then, P signals $\{\mathbf{y}_p \in \mathbb{H}^N\}_{p=1}^{P}$ are randomly created. The computational time (in seconds) is noted to decompose the P signals over the dictionary $\mathbf{D}$, when the Q-OMP is stopped after $K = 3$ iterations for each signal. The algorithms were run on Matlab 2013 (8.1), as 32-bit, on an Intel Core i7 CPU of 2.4 GHz.
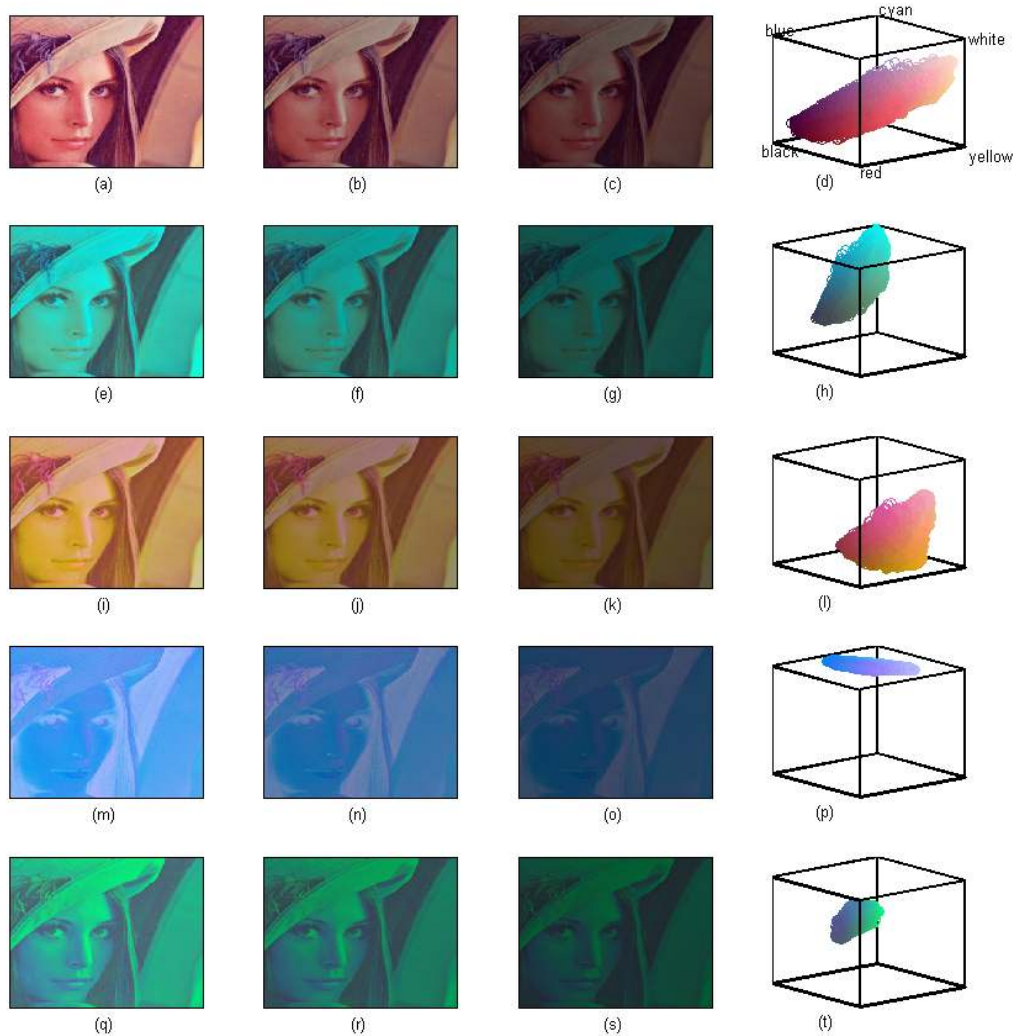
Fig. 7. Comparison of the transformations applied by the color models on an atom. Vertical: scalar multiplication: $x = 255$ (left), 180 (middle), and 105 (right). Horizontal: color filtering: no filtering for the vectorized model (1st row), two random filterings for the quaternionic model (2nd and 3rd rows), and two random filterings for the color filter model (4th and 5th rows).

take into account color filters during atom updates. Theoretically, the uniqueness and stability of these learned dictionaries have to be established.

3) The color dimension that is composed of three channels here, can be generalized in a spectral dimension composed of C channels. This generalization allows multi- or hyperspectral images to be considered [58], denoted as $Y \in \mathbb{R}^{N \times C}$ and decomposed on spectral atoms $D_m \in \mathbb{R}^{N \times C}$ with their associated spectral filters $W_m \in \mathbb{R}^{C \times C}$. This thus opens a new field of applications (*e.g.*, remote sensing, spectroscopy, and others).

C. Other general prospects

Other general research directions can be studied for color images, such as:

1) Generally, a global dictionary is learned on a database, and it is then adapted on the processed image to give a local/ adaptive dictionary [3], [6]. This step is important, to ensure better efficiency [6] and more colored atoms [7]. This point seems to be less crucial for the quaternion-based approach where the global dictionary is already efficient [23], [25]. Thus, the influence of global *versus* local dictionary is not clear yet.

2) Dictionary learning algorithms have several hyperparameters. It will be useful to test the influence of these parameters on the processing [6], [7], [25], such as the patch size, redundancy level M/N of the dictionary, number of iterations in the DLAs, and others.

3) Current color methods can be improved, including in particular the structure in the patch analysis, such as the multi-
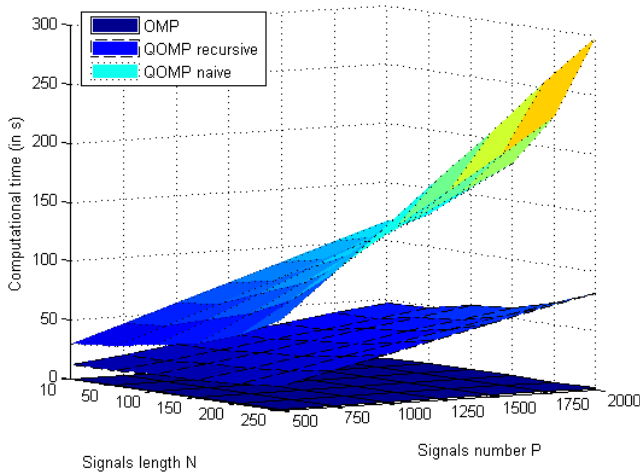
Fig. 8.   Averaged computational time (in seconds) of the vectorized OMP, recursive Q-OMP, and naïve Q-OMP to decompose signals, as a function of the signal length N and the signal number P.

scale architecture [7], [59], spatial shift-invariance (also called convolutional) [60], [61], rotation invariance [62], multi-layer architecture from deep learning [63], and others.

Even though this section is not an exhaustive list, all of these points can contribute to improvements in the results of color image processing.

IX. CONCLUSIONS

In this article, a review of dictionary-learning-based sparse representations for color images is provided, which details the differences between the models and compares their results for real and simulated data. These models are considered in a unifying framework that is based on the degrees of freedom of the linear filtering/transformation of the color channels. Moreover, this allows the scalar quaternionic linear model to be shown as equivalent to constrained matrix-based color filtering, which highlights the filtering implicitly applied through this model. With four degrees of freedom, this linear filtering is a scaled shearing transformation with anti-symmetric off-diagonal coefficients. Then, the new color filtering model is introduced, using unconstrained filters and with nine degrees of freedom. In this model, spatial morphologies of color images are encoded by atoms, and the colors are encoded by color filters. Color variability is no longer captured with increasing of the dictionary size, but with color filters, which gives an efficient color representation.

When the atoms can be replicated in a collection of shifted patterns (*resp.* rotated patterns), the model is called shift-invariant [60], [61] (*resp.* rotation invariant [62]). Similarly, as atoms can be replicated in a family of colored patterns, the model introduced can be called color invariant. This opens a lot of prospects for the improvement of color image processing.

ACKNOWLEDGEMENTS

REFERENCES

[1] I. Tošić and P. Frossard, "Dictionary learning," *IEEE Signal Processing Magazine*, vol. 28, pp. 27–38, 2011.

[2] M. Elad, M. Figueiredo, and Y. Ma, "On the role of sparse and redundant representations in image processing," *Proceedings of the IEEE - Special Issue on Applications of Sparse Representations and Compressive Sensing*, vol. 98, pp. 972–982, 2010.

[3] A. Bruckstein, D. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, vol. 51, pp. 34–81, 2009.

[4] J. Wright and Y. Ma, "Sparse representation for computer vision and pattern recognition," *Proceedings of the IEEE*, vol. 98, pp. 1031–1044, 2010.

[5] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. on Signal Processing*, vol. 54, pp. 4311–4322, 2006.

[6] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Trans. on Image Processing*, vol. 15, pp. 3736–3745, 2006.

[7] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. on Image Processing*, vol. 17, pp. 53–69, 2008.

[8] O. Guleryuz, "Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising - Part I: theory," *IEEE Trans. on Image Processing*, vol. 15, pp. 539–554, 2006.

[9] ——, "Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising - Part II: adaptive algorithms," *IEEE Trans. on Image Processing*, vol. 15, pp. 555–571, 2006.

[10] J. Yang, J. Wright, T. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Trans. on Image Processing*, vol. 19, pp. 2861–2873, 2010.

[11] R. Zeyde, M. Elad, and M. Protter, "On single image scale-up using sparse-representations," in *Curves and Surfaces*, ser. Lecture Notes in Computer Science, vol. 6920, 2012, pp. 711–730.

[12] Y. Chen, S. Sastry, V. Patel, P. Phillips, and R. Chellappa, "In-plane rotation and scale invariant clustering using dictionaries," *IEEE Trans. on Image Processing*, vol. 22, pp. 2166–2180, 2013.

[13] J. Wright, A. Yang, A. Ganesh, S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 31, pp. 210–227, 2009.

[14] R. Ptucha and A. Savakis, "LGE-KSVD: Robust sparse representation classification," *IEEE Trans. on Image Processing*, vol. 23, pp. 1737–1750, 2014.

[15] G. Sharma and H. Trussel, "Digital color imaging," *IEEE Trans. on Image Processing*, vol. 6, pp. 901–932, 1997.

[16] B. Russell, A. Torralba, K. Murphy, and W. Freeman, "LabelMe: a database and web-based tool for image annotation," *International Journal of Computer Vision*, vol. 77, pp. 157–173, 2008.

[17] J. Mairal, M. Elad, and G. Sapiro, "Sparse learned representations for image restoration," in *IASC2008*, 2008.

[18] S. Sangwine, "Fourier transforms of colour images using quaternion or hypercomplex, numbers," *Electronics Letters*, vol. 32, pp. 1979–1980, 1996.

[19] C. Moxey, S. Sangwine, and T. Ell, "Hypercomplex correlation techniques for vector images," *IEEE Trans. on Signal Processing*, vol. 51, pp. 1941–1953, 2003.

[20] N. Le Bihan and S. Sangwine, "Quaternion principal component analysis of color images," in *Proc. IEEE Int. Conf. Image Processing ICIP 2003*, 2003, pp. 809–812.

[21] T. Ell and S. Sangwine, "Hypercomplex Fourier transforms of color images," *IEEE Trans. on Image Processing*, vol. 16, pp. 22–35, 2007.

[22] B. Augereau and P. Carré, "Hypercomplex polynomial wavelet packet application for color image," in *Conf. Applied Geometric Algebras in Computer Science and Engineering AGACSE*, 2012.

[23] L. Yu, Y. Xu, H. Xu, and H. Zhang, "Quaternion-based sparse representation of color image," in *Proc. IEEE Int. Conf. Multimedia and Expo ICME 2013*, 2013, pp. 1–7.

[24] M. Yu, Y. Xu, and P. Sun, "Single color image super-resolution using quaternion-based sparse representation," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing ICASSP '14*, 2014, pp. 5804–5808.

[25] A. Carmeli and J. Turek, "Quaternion K-SVD for color image denoising," Technion - Israel Institute of Technology, Tech. Rep., 2013.

[26] C. Deavours, "The quaternion calculus," *American Mathematical Monthly*, vol. 80, pp. 995–1008, 1973.

[27] F. Zhang, "Quaternions and matrices of quaternions," *Linear Algebra and its Applications*, vol. 251, pp. 21–57, 1997.

[28] Q. Barthélemy, A. Larue, and J. Mars, "Sparse approximations for quaternionic signals," *Advances in Applied Clifford Algebras*, vol. 24, pp. 383–402, 2014.

[29] M. Lewicki and T. Sejnowski, "Learning overcomplete representations," *Neural Comput.*, vol. 12, pp. 337–365, 1998.

[30] S. Mallat, *A Wavelet Tour of signal processing*. 3rd edition, New-York : Academic, 2009.

[31] J. Tropp and S. Wright, "Computational methods for sparse solution of linear inverse problems," *Proceedings of the IEEE*, vol. 98, pp. 948–958, 2010.

[32] S. Mallat and Z. Zhang, "Matching pursuits with time-frequency dictionaries," *IEEE Trans. on Signal Processing*, vol. 41, pp. 3397–3415, 1993.

[33] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Conf. Record of the Asilomar Conf. on Signals, Systems and Comput.*, 1993, pp. 40–44.

[34] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. Sci. Comput.*, vol. 20, pp. 33–61, 1998.

[35] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani, "Least Angle Regression," *Ann. Stat.*, vol. 32, pp. 407–499, 2004.

[36] I. Daubechies, M. Defrise, and C. De Mol, "An iterative algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. LVII, pp. 1413–1457, 2004.

[37] Q. Barthélemy, A. Larue, and J. Mars, "Quaternionic sparse approximation," in *Conf. Applied Geometric Algebras in Computer Science and Engineering AGACSE*, 2012.

[38] K. Engan, S. Aase, and J. Husøy, "Multi-frame compression: theory and design," *Signal Process.*, vol. 80, pp. 2121–2140, 2000.

[39] K. Skretting, J. Husøy, and S. Aase, "General design algorithm for sparse frame expansions," *Signal Process.*, vol. 86, pp. 117–126, 2006.

[40] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, pp. 19–60, 2010.

[41] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Trans. on Signal Processing*, vol. 58, pp. 2121–2130, 2010.

[42] N. Le Bihan and J. Mars, "Singular value decomposition of quaternion matrices : A new tool for vector-sensor signal processing," *Signal Process.*, vol. 84, pp. 1177–1199, 2004.

[43] S. Roth and M. Black, "Fields of Experts: a framework for learning image priors," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition CVPR 2005*, 2005, pp. 860–867.

[44] ——, "Fields of Experts," *International Journal of Computer Vision*, vol. 82, pp. 205–229, 2009.

[45] P. Hoyer and A. Hyvärinen, "Independent component analysis applied to feature extraction from colour and stereo images," *Network: Computation in Neural Systems*, vol. 11, pp. 191–210, 2000.

[46] T. Ell, "Hypercomplex color affine filters," in *Proc. IEEE Int. Conf. Image Processing ICIP 2007*, 2007, pp. 249–252.

[47] R. Soulard, P. Carré, and C. Fernandez-Maloigne, "Vector extension of monogenic wavelets for geometric representation of color images," *IEEE Trans. on Image Processing*, vol. 22, pp. 1070–1083, 2013.

[48] S. Sangwine and N. Le Bihan, "Quaternion toolbox for matlab," http://sourceforge.net/projects/qtfm, 2005.

[49] M. Aharon, M. Elad, and A. Bruckstein, "On the uniqueness of overcomplete dictionaries, and a practical way to retrieve them," *Linear Algebra and its Applications*, vol. 416, pp. 48–67, 2006.

[50] Q. Barthélemy, "Représentations parcimonieuses pour les signaux multivariés," Ph.D. dissertation, Université de Grenoble, 2013.

[51] Q. Barthélemy, A. Larue, and J. Mars, "Decomposition and dictionary learning for 3D trajectories," *Signal Process.*, vol. 98, pp. 423–437, 2014.

[52] J. Vince, *Matrix Transforms for Computer Games and Animation*. Springer London, 2012.

[53] S. Chevallier, Q. Barthélemy, and J. Atif, "On the need for metrics in dictionary learning assessment," in *Proc. Eur. Signal Process. Conf. EUSIPCO '14*, 2014, pp. 1427–1431.

[54] D. Donoho, M. Elad, and V. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. on Information Theory*, vol. 52, pp. 6–18, 2006.

[55] J. Tropp, "Just relax : Convex programming methods for subset selection and sparse approximation," *IEEE Trans. on Information Theory*, vol. 52, pp. 1030–1051, 2006.

[56] R. Gribonval and K. Schnass, "Dictionary identification — sparse matrix-factorization via ℓ_1-minimization," *IEEE Trans. on Information Theory*, vol. 56, pp. 3523–3539, 2010.

[57] R. Gribonval, R. Jenatton, and F. Bach, "Sparse and spurious: dictionary learning with noise and outliers," *Preprint arXiv:1407.5155*, 2014.

[58] Y. Chen, N. Nasrabadi, and T. Tran, "Hyperspectral image classification using dictionary-based sparse representation," *IEEE Trans. on Geoscience and Remote Sensing*, vol. 49, pp. 3973–3985, 2011.

[59] J. Mairal, G. Sapiro, and M. Elad, "Learning multiscale sparse representations for image and video restoration," *SIAM Multiscale Modelling and Simulation*, vol. 7, pp. 214–241, 2008.

[60] H. Wersing, J. Eggert, and E. Körner, "Sparse coding with invariance constraints," in *Proc. Int. Conf. Artificial Neural Networks ICANN*, 2003, pp. 385–392.

[61] M. Mørup, M. Schmidt, and L. Hansen, "Shift invariant sparse coding of image and music data," Technical University of Denmark, Tech. Rep., 2008.

[62] M. Mørup and M. Schmidt, "Transformation invariant sparse coding," in *Proc. Machine Learning for Signal Processing MLSP '11*, 2011, pp. 1–6.

[63] A. Krizhevsky, I. Sutskever, and G. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems NIPS*, 2012, pp. 1097–1105.

Quentin Barthélemy received the Engineering degree and the M.Res. in signal and images analysis and processing (with distinction) from Grenoble Institut National Polytechnique (Grenoble INP), France, both in 2009; and the Ph.D. degree in signal processing from Grenoble University and CEA-LIST (Alternative Energies and Atomic Energy Commission), France, in 2013. His Ph.D. dissertation deals with sparse representations for multivariate signals, including invariances as shift, rotation and affine transformation.

He joined Mensia Technologies at the Institut du Cerveau et de la Moelle Epinière, Paris, France, in 2013, to develop signal processing and machine learning methods for real-time EEG analysis. His research interests are sparse representation, time-frequency analysis, source separation and Riemannian geometry.

Anthony Larue received the Aggregation in electrical engineering at the Ecole Normale Supérieure de Cachan, France, in 2002, the M.S. degree in automatic control and signal processing from the University of Paris XI, France, in 2003, and the Ph.D. degree in signal processing from Institut National Polytechnique of Grenoble, France, in 2006. His Ph.D. dissertation deals with blind deconvolution of noisy data for seismic applications.

He joined the CEA-LIST, Gif-sur-Yvette, France, in 2006. His research interests are signal processing, machine learning, and especially sparse decomposition of multicomponent signals. Since 2010, he has been the head of the Data Analysis and System intelligence Laboratory, which developed data analysis algorithms for health, security, or energy applications.

Jérôme I. Mars received the M.S. degree in geophysics from Joseph Fourier University of Grenoble, France, in 1986 and the Ph.D. degree in signal processing from the Institut National Polytechnique of Grenoble, France, in 1988.

From 1989 to 1992, he was a Postdoctoral Researcher at the Centre des Phénomènes Aléatoires et Geophysiques, Grenoble, France. From 1992 to 1995, he was a Visiting Lecturer and Scientist at the Materials Sciences and Mineral Engineering Department at the University of California, Berkeley.

He is Professor with Grenoble-INP (Grenoble Institute of Technology), Grenoble, France, within the Grenoble Image Parole Signal Automatic Laboratory (GIPSA-Lab) in the field of Signal Processing. He is currently a Director of GIPSA-Lab after several years of leading the Image and Signal Dept. His research interests include signal processing, source separation, time frequency time-scale representation in the field of underwater acoustics and antennas. Dr. Mars is member of EAGE.