Enhancing sparse representation of color images by cross channel transformation

Laura Rebollo-Neira
Mathematics Department
Aston University
B4 7ET, Birmingham, UK
Aurelien Inacio
ENSIIE, 1 Rue de la Résistance,
91000 Évry-Courcouronnes, France
May 17, 2021

Abstract
Transformations for enhancing sparsity in the approximation of color images by 2D atomic decomposition are discussed. The sparsity is firstly considered with respect to the most significant coefficients in the wavelet decomposition of the color image. The discrete cosine transform is singled out as an effective transformation for this purpose. The enhanced feature is further exploited by approximating the transformed arrays using an effective greedy strategy with a separable highly redundant dictionary. The relevance of the achieved sparsity is illustrated by a simple encoding procedure. On a set of typical test images the compression at high quality recovery is shown to significantly improve upon JPEG and WebP formats. The results are competitive with those produced by the JPEG2000 standard.

2 Introduction

In the signal processing field sparse representation usually refers to the approximation of a signal as superposition of elementary components, called atoms, which are members of a large redundant set, called a dictionary \([1]\). The superposition, termed atomic decomposition, aims at approximating the signal involving as few atoms as possible \([1,4]\). Sparsity is also relevant to data collection. Within the emerging theory of sampling known as *compressive sensing* (CS) this property is key for reconstructing signals from a reduced number of measures \([5,7]\). In particular, distributed compressive sensing (DCS) algorithms exploit inter signal correlation structure for multiple signal recovery \([8,9]\).

Sparse image representation using redundant dictionaries has been considered in numerous works e.g. \([11,13]\) and in the context of applications such as image restoration \([14,16]\), feature extraction \([17,20]\) and super resolution \([21,24]\).

The sparsity property of some types of 3D images benefits from 3D processing \([25,28]\). In particular most RGB (Red-Green-Blue) images admit a sparser atomic decomposition if
approximated by 3D atoms [29]. Within the 3D framework the gain in sparsity comes at expenses of computational cost though.

The purpose of this work is to show that the application of a transformation across the direction of the colors improve sparsity in the wavelet domain representation of the 2D color channels. The relevance of this feature is demonstrated by a simple encoding procedure rendering good compression results in comparison to the state of the art.

The contributions of the paper are listed below.

- Highlighting of a distinctive feature concerning sparse representation of RGB images. Namely, that transformations in the direction of the color channels significantly improve the sparsity of the 2D atomic decomposition of the image.

- Demonstration that the discrete cosine transform (dct) can improve sparsity with respect to the optimal decorrelation transform, i.e. that given by the eigenvectors of the correlation matrix, which is thereby image dependent.

- Showcasing of the relevance of sparsity to image compression with high quality recovery. On a set of 15 typical test images the achieved compression is shown to significantly improve upon JPEG and WebP formats. The results are competitive with those produced by JPEG2000.

The work is organised as follows: Sec. 3 introduces the mathematical notation. Sec. 4 compares several cross color transformations enhancing sparsity. A numerical example on a large data set is used to illustrate the suitability of the dct for that purpose. Sec. 5 demonstrates the gain in sparsity obtained by the atomic decomposition of color images when the dct is applied across the RGB channels. Sec. 6 illustrates the relevance of the approach to image compression with high quality recovery. The conclusions are summarised in Sec. 7.

## 3 Mathematical Notation

Throughout the paper we use the following notational convention. \( \mathbb{R} \) represents the set of real numbers. Boldface letters indicate Euclidean vectors, 2D and 3D arrays. Standard mathematical fonts indicate components, e.g., \( \mathbf{d} \in \mathbb{R}^N \) is a vector of components \( d(i) \in \mathbb{R}, i = 1, \ldots, N \). The elements of a 2D array \( \mathbf{I} \in \mathbb{R}^{N_x \times N_y} \) are indicated as \( I(i, j), i = 1, \ldots, N_x, j = 1, \ldots, N_y \) and the color channels of \( \mathbf{I} \in \mathbb{R}^{N_x \times N_y \times 3} \) as \( I(:, :, z), z = 1, 2, 3 \). The transpose of a matrix, \( \mathbf{G} \), say, is indicated as \( \mathbf{G}^\top \). A set of, say \( M \) color images is represented by the arrays \( \mathbf{I} \{m\} \in \mathbb{R}^{N_x \times N_y \times 3}, m = 1, \ldots, M \).

The inner product between arrays in \( \mathbb{R}^{N_x \times N_y} \) is given by the Frobenius inner product \( \langle \cdot, \cdot \rangle_F \) as

\[
\langle \mathbf{G}, \mathbf{I} \rangle_F = \sum_{i=1}^{N_x} \sum_{j=1}^{N_y} G(i, j)I(i, j).
\]

Consequently, the Frobenius norm \( \| \cdot \|_F \) is calculated as

\[
\| \mathbf{G} \|_F = \sqrt{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} G(i, j)^2}.
\]

Consequently, the Frobenius norm \( \| \cdot \|_F \) is calculated as

\[
\| \mathbf{G} \|_F = \sqrt{\sum_{i=1}^{N_x} \sum_{j=1}^{N_y} G(i, j)^2}.
\]
The inner product between arrays in $\mathbb{R}^N$ is given by the Euclidean inner product $\langle \cdot, \cdot \rangle$ as

$$\langle d, g \rangle = \sum_{i=1}^{N} d(i) g(i).$$

4 Cross color transformations

Given a color image $I(i,j,k), i = 1, \ldots, N_x, j = 1, \ldots, N_y, k = 1, 2, 3$ the processing of the 3 channels can be realised either in the pixel/intensity or in the wavelet domain. Since the representation of most images is sparser in the wavelet domain $[29][33]$ we approximate in that domain and reconstruct the approximated image by the inverse wavelet transform. Thus, using a $3 \times 3$ matrix $T$, we construct the transformed arrays $U$ and $W$ as follows

$$U(:, :, z) = \sum_{l=1}^{3} I(:, :, l) T(l, z) \quad z = 1, 2, 3,$$

$$W(:, :, z) = \text{dwt}(U(:, :, z)) \quad z = 1, 2, 3,$$

where $\text{dwt}$ indicates the 2D wavelet transform. For the transformation $T$ we consider the following cases

(i) The dct.

(ii) The reversible YCbCr color space transform.

(iii) The principal components (PC) transform.

(iv) A transformation learned from an independent set of images.

The dct is given by the matrix

$$\begin{pmatrix}
\frac{1}{\sqrt{3}} & \frac{\sqrt{2}}{\sqrt{3}} \cos\left(\frac{\pi}{6}\right) & \frac{\sqrt{2}}{\sqrt{3}} \cos\left(\frac{\pi}{3}\right) \\
\frac{1}{\sqrt{3}} & 0 & \frac{\sqrt{2}}{\sqrt{3}} \cos\left(\frac{5\pi}{6}\right) \\
\frac{1}{\sqrt{3}} & \frac{\sqrt{2}}{\sqrt{3}} \cos\left(\frac{7\pi}{6}\right) & \frac{\sqrt{2}}{\sqrt{3}} \cos\left(\frac{5\pi}{6}\right)
\end{pmatrix}$$

The YCbCr color space transform is given by the matrix below $[35]$

$$\begin{pmatrix}
0.299 & -0.169 & 0.5 \\
0.587 & -0.331 & -0.419 \\
0.114 & 0.5 & -0.0813
\end{pmatrix}$$

The columns of the principal components transform are the normalised eigenvectors of covariance matrix of the RGB pixels. Thus, the transformation is image dependent and optimal with respect to decorrelating the color channels.

As a first test, the approximation of the transformed channels is realised by keeping a fixed number of the largest absolute value entries, and setting the others equal to zero. In relation to this, for an image of size $L_x \times L_y \times 3$ we define the Sparsity Ratio (SR) as follows

$$\text{SR} = \frac{L_x \cdot L_y \cdot 3}{\text{Number of nonzero entries in the three channels.}}$$
The quality of a reconstructed image $\tilde{I}$, with respect to the original 8-bit image $I$, is compared using the Peak Signal-to-Noise Ratio (PSNR),

$$\text{PSNR} = 10 \log_{10} \left( \frac{255^2}{\text{MSE}} \right),$$

where

$$\text{MSE} = \frac{1}{L_x \cdot L_y \cdot 3} \sum_{i,j,z=1}^{L_x \cdot L_y \cdot 3} (I(i, j, z) - \tilde{I}(i, j, z))^2.$$ 

For the numerical examples below the transformation corresponding to case (iv) is learned from a set of images $I\{m\}$, $m = 1, \ldots, M$ all of the same size. Starting from an invertible $3 \times 3$ matrix $T^k$, with $k = 1$, the learning algorithm proceeds through the following instructions.

1) Use $T^k$ to transform the arrays $I\{m\} \rightarrow U^k\{m\} \rightarrow W^k\{m\}$ as in (1) and (2).

2) Approximate each transformed array $W^k\{m\}$ to obtain $\tilde{W}^k\{m\}$ by keeping the largest $K$ absolute value entries.

3) Apply the inverse 2D wavelet transform $\text{idwt}$ to reconstruct the approximated arrays $\tilde{U}^k\{m\}, m = 1, \ldots, M$ as

$$\tilde{U}^k\{m\}(i, j) = \text{idwt}(\tilde{W}^k\{m\}(i, j)), z = 1, 2, 3.$$

4) Use the original images $I\{m\}, m = 1, \ldots, M$ to find $G = T^{-1}$ by least squares fitting, i.e

$$G^* = \arg \max_{G} \mathcal{E}(G),$$

where

$$\mathcal{E}(G) = \sum_{m=1}^{M} \sum_{i,j,l=1}^{L_x \cdot L_y \cdot 3} (I\{m\}(i, j, l) - \tilde{I}\{m\}(i, j, l))^2$$

with

$$\tilde{I}\{m\}(i, j, l) = \sum_{z=1}^{3} \tilde{U}^k\{m\}(i, j, z)G(z, l).$$

5) While $\mathcal{E}$ decreases, or the maximum number of allowed iterations has not been reached, set $k \rightarrow k + 1$, $T^k = (G^*)^{-1}$ and repeat steps 1) - 5).

Given the arrays $\tilde{U}^k\{m\}, m = 1, \ldots, M$ the least squares problem for determining the transformation $T^k$ has a unique solution. However, the joint optimisation with respect to the arrays $\tilde{U}^k\{m\}, m = 1, \ldots, M$ and the transformation $T^k$ is not convex. Hence, the algorithm’s outcome depends on the initial value.

The transformation (iv) used in the numerical examples of Secs. 4.1 and 5.1 has been learned from $M = 100$ images, all of size $384 \times 512 \times 3$, from the UCID data set [34], which contains images of buildings, places and cars. The learning curves for two random orthonormal initialisations is shown in Fig. [1].

It is worth mentioning that, as shown in Fig. [1], learning is richer when starting comparatively distant from a local minimum (left graph in Fig. [1]). However, since the convergence is to a local minimum other random initialisations, even if generating less learning, may give better results (right graph in Fig. [1]).

The aim of the next numerical test is to demonstrate the effect on the SR (c.f. (3)) produced by the above (i) - (iv) transformations across the color channels.
Figure 1: Learning curves for the transformation (iv) corresponding to two different random orthogonal transforms initialising the process. The mean value PSNR with respect to the 100 images in the training set corresponds to SR=15 for all the images.

4.1 Numerical Test I

Using the whole Berkeley data set [36], consisting of 300 images all of size $321 \times 481 \times 3$ we proceed with each image as in [1] and [2]. The dwt corresponds to the Cohen-Daubechies-Feauveau 9/7 (CDF97) wavelet family. Fig. 3 shows the transformed channels of the image in Fig 2 including the dct transformation across channels (right graph) and without $T$ transformation (left graph).

Figure 2: One of the RGB images in the Berkeley’s data set

For the numerical test the approximations are realised fixing SR=20 and SR=10. The
reconstructed images are obtained for the approximated arrays \( \tilde{W} \) as
\[
\tilde{U}(;:;z) = \text{idwt}(\tilde{W}(;:;z)), \quad z = 1, 2, 3 
\] (5)
\[
\tilde{I}(;:;z) = \sum_{k=1}^{3} \tilde{U}(;:;k)(T^{-1}(z,k)), \quad z = 1, 2, 3 
\] (6)
where \( T^{-1} \) is the inverse of \( T \). When no transformation \( T \) is considered the image is reconstructed directly from (5).

The 2nd and 4th columns of Table 1 show the mean value PSNR (PSNR), with respect to the whole data set, for \( SR=20 \) and \( SR=10 \) respectively, corresponding to the transformations \( T \) listed in the 1st column of the table. The 3rd and 5th columns give the standard deviations (std).

While Table 1 shows that all (i) - (iv) transformations render equivalent superior results, with respect to not applying a transformation (case (v)), the dct slightly exceeds the others. Case (iv) refers to the best results achieved when initialising the learning algorithm with 500 different random orthonormal transformations. When initialising the algorithm with transformations (i) - (iii) it appears that each of these transformations is close to a local minimiser of the algorithm. This stems from the fact that such initialisations do not generate significant learning.

The common feature of most of the 300 images in the data set used in this numerical example is the correlation property of the three color channels. This property is assessed by
Table 1: Mean value PSNR, with respect to the 300 images in the Berkeley data set. The approximation of each image is realised by setting the least significant entries in the arrays $W_m = 1, \ldots, 300$ equal zero, in order to obtain $SR=20$ for all the images (2nd column) and $SR=10$ for all the images (4th column).

| Transf. | PSNR | std | PSNR | std |
|---------|------|-----|------|-----|
| (i) dct | 33.9 | 5.0 | 38.9 | 5.1 |
| (ii) YCbCr | 33.7 | 5.0 | 38.7 | 5.1 |
| (iii) PC | 33.7 | 4.9 | 38.4 | 5.0 |
| (iv) Learned | 33.7 | 4.9 | 38.7 | 5.0 |
| (v) No transf. | 29.7 | 4.8 | 32.9 | 5.2 |

Table 5 Approximations by atomic decomposition

We have seen (c.f. Table 1) that by transformation of channels it is possible to gain quality when reducing nonzero entries in the channels. Now we discuss how to improve quality further by approximating the 2D arrays by an atomic decomposition, other than just by neglecting their less significant entries. For the approximation to be successful it is important to use an appropriate dictionary. To this end, one possibility could be to learn the dictionary from training data [38–45]. However as demonstrated in previous works [29, 32, 33] a separable dictionary, which is easy to construct, is well suited for the purposes of achieving sparsity and delivers a fast implementation of the approach. Since we use that dictionary in the numerical examples, below we describe the method for constructing the atomic decomposition of the array $W$ considering specifically a separable dictionary.
Figure 4: Histograms of the correlation coefficients between the RGB channels.

Firstly we concatenate the 3 planes \( W(:, :, z), z = 1, 2, 3 \) into an extended 2D array \( W' \in \mathbb{R}^{3L_x \times L_y} \) and divide this array in small blocks \( W'_q, q = 1, \ldots, Q \). Without loss of generality the blocks are assumed to be square of size \( N_b \times N_b \) say, and are approximated using separable dictionaries \( \mathcal{D}^x = \{d^x_n \in \mathbb{R}^{N_b}, \|d^x_n\|_2 = 1\}_{n=1}^{M_x} \) and \( \mathcal{D}^y = \{d^y_m \in \mathbb{R}^{N_b}, \|d^y_m\|_2 = 1\}_{m=1}^{M_y} \), with \( M_xM_y = M \).

For \( q = 1, \ldots, Q \) every element \( W'_q \) is approximated by an atomic decomposition as below:

\[
W'_{kq} = \sum_{n=1}^{k_q} c^{k_q}_{n}(n) d^x_{\ell^x_{k_q,n}}(d^y_{\ell^y_{k_q,n}})^T, \tag{9}
\]

where \( \ell^y_{n,q} \) is the index in the set \( \{1, 2, \ldots, M_y\} \) corresponding to the label of the atom in the dictionary \( \mathcal{D}^y \) contributing to the \( n \)-th term in the approximation of the \( q \)-th block. The index \( \ell^x_{n,q} \) has the equivalent description. The assembling of the approximated blocks gives rise to the approximated array \( W'^K \); i.e., \( W'^K = \hat{J}_{q=1}^{Q} W'_{kq} \), where \( K = \sum_{q=1}^{Q} k_q \) and \( \hat{J} \) represents the assembling operation.

The approximation of the partition \( W'_q = 1, \ldots, Q \) is carried out iteratively as a two step process which selects i) the atoms in the atomic decomposition \( \{9\} \) and ii) the sequence in which the blocks in the partition are approximated. The procedure is called Hierarchized Block Wise (HBW) implementation of greedy pursuit strategies \([31, 46]\). For the selection of the atoms we apply the Orthogonal Matching Pursuit (OMP) approach \([47]\) dedicated to 2D with separable dictionaries (OMP2D) \([18]\). Thus, the whole algorithm is termed HBW-OMP2D \([31]\). The method iterates as described below.

On setting \( k_q = 0 \) and \( R^0_q = W'_q \) at iteration \( k_q + 1 \) the algorithm selects the indices \( \ell^x_{k_q+1} \).
and \( \ell_{k_q+1}^{x,q} \), as follows:

\[
\ell_{k_q+1}^{x,q} = \arg\max_{n=1,\ldots,M_x, m=1,\ldots,M_y} \left| \left\langle d_{x,q}^{x,n} R_q^{k_q} d_{y}^{y,m} \right\rangle_F \right|, \quad q = 1, \ldots, Q,
\]

(10)

where \( R_q^{k_q} = W_q^r - W_q^{rk_q} \). The calculation of \( W_q^{rk_q} \) is realised in order to minimise \( \| R_q^{k_q} \|_F \), which is equivalent to finding the orthogonal projection onto the subspace spanned by the selected atoms \( \{ A_n = d_{x,q}^{x,n} (d_{y,q}^{y,n})^T \}_{n=1}^{k_q} \). In our implementation, the calculation of the coefficients \( c^q(n), n = 1, \ldots, k_q \) in (9) is realised as

\[
c_{k_q,q}(n) = \left\langle B_{n}, W_q^{rk_q} \right\rangle_F, \quad n = 1, \ldots, k_q,
\]

(11)

where the set \( \{ B_{n}^{k_q} \}_{n=1}^{k_q} \) is biorthogonal to the set \( \{ A_n \}_{n=1}^{k_q} \) and needs to be upgraded and updated to account for each newly selected atom. Starting from \( B_1^{k_q} = A_1 = d_{x,q}^{x,1} (d_{y,q}^{y,1})^T \), the updating and upgrading is realised through the recursive equations [48,49]:

\[
\begin{align*}
B_{n+1}^{k_q} &= B_{n}^{k_q} - B_{n+1}^{k_q} \left\langle A_{k_q+1}, B_{n}^{k_q} \right\rangle_F, \quad n = 1, \ldots, k_q \\
Q_{k_q+1} &= A_{k_q+1} - \sum_{n=1}^{k_q} \frac{Q_n}{\| Q_n \|_F} \left\langle Q_n, A_{k_q+1} \right\rangle_F,
\end{align*}
\]

(12)

with the additional re-orthogonalisation step

\[
Q_{k_q+1} \leftarrow Q_{k_q+1} - \sum_{n=1}^{k_q} \frac{Q_n}{\| Q_n \|_F} \left\langle Q_n, Q_{k_q+1} \right\rangle_F.
\]

(13)

As discussed in [31,46], for \( \ell_{k_q+1}^{x,q} \) and \( \ell_{k_q+1}^{y,q} \), \( q = 1, \ldots, Q \) the indices resulting from (10), the block to be approximated in the next iteration corresponds to the value \( q^* \) such that

\[
q^* = \arg\max_{q=1,\ldots,Q} \left| \left\langle d_{x,q}^{x,q+1}, R_q^{k_q} d_{y,q}^{y,q+1} \right\rangle \right|.
\]

The algorithm stops when the required total number of \( K \) atoms has been selected. This number can be fixed using the SR, which is now calculated as

\[
SR = \frac{\text{Number of elements in the arrays}}{K}.
\]

(14)

**Remark 1.** The above described implementation of HBW-OMP2D is very effective in terms of speed, but demanding in terms of memory (the partial outputs corresponding to all the blocks in the partition need to be stored at every iteration). An alternative implementation, termed HBW Self Projected Matching Pursuit (HBW-SPMP) [32,50], would enable the application of the identical strategy to much larger images than the ones considered in this work.
5.1 Numerical example II

For this and the next numerical example, we use a mixed dictionary consisting of two classes of sub-dictionaries of different nature:

I) The trigonometric dictionaries $D^x_C$ and $D^x_S$, defined below,

$D^x_C = \{ w_c(n) \cos \frac{\pi (2i-1)(n-1)}{2M}, i = 1, \ldots, N_b \}_{n=1}^M$

$D^x_S = \{ w_s(n) \sin \frac{\pi (2i-1)(n)}{2M}, i = 1, \ldots, N_b \}_{n=1}^M$

where $w_c(n)$ and $w_s(n)$, $n = 1, \ldots, M$ are normalisation factors.

II) The dictionary $D^x_L$, which is constructed by translation of the prototype atoms in Fig. 5. The mixed dictionary $D^x$ is built as $D^x = D^x_C \cup D^x_S \cup D^x_L$ and $D^y = D^x$.

![Figure 5: Prototypes (each in different color) that generate the dictionaries $D^x_L$ by sequential translations of one point.](image)

Table 2 shows the improvement in PSNR achieved by atomic decompositions using the mixed dictionary for SR=20 and SR=10.

| Transf.   | PSNR | std | PSNR | std |
|-----------|------|-----|------|-----|
| (i) dct   | 40.5 | 5.0 | 48.1 | 4.4 |
| (ii) YCbCr| 40.3 | 5.0 | 47.8 | 4.4 |
| (iii) PC  | 39.6 | 4.8 | 46.3 | 4.5 |
| (iv) Learned | 40.3 | 4.9 | 47.4 | 4.3 |
| (v) No transf. | 34.0 | 5.2 | 39.1 | 5.6 |

Table 2: Mean value PSNR, with respect to 300 images in the Berkeley data set, produced by 2D atomic decomposition of the arrays $W\{m\} = 1, \ldots, 300$ in order to obtain SR=20 (2nd column) and SR=10 (4th column).

Notice that while case (v), which does not include any T transformation, gives superior results than by disregarding entries (c.f. Table 1) when applying any of the transformations (i) –
the results improve further. The PC transform, however, appears significantly less effective than the others. In addition to rendering the best results, the dct brings along the additional advantage of being orthonormal. Consequently, it does not magnify errors at the inversion step. Because of this, we single out the dct as the most convenient cross color transformation out of the four considered here.

6 Application to image compression

In order to achieve compression by filing an atomic decomposition we need to address two issues. Namely, the quantisation of the coefficients $c_q(n)$, $n = 1, \ldots, k_q$, $q = 1, \ldots, Q$ in (9) and the storage of the indices $(\ell_x^n, \ell_y^n)$, $n = 1, \ldots, k_q$, $q = 1, \ldots, Q$. We tackle the matters by simple but effective procedures [33].

For $q = 1, \ldots, Q$ the absolute value coefficients $|c_q(n)|$, $n = 1, \ldots, k_q$ are converted to integers through uniform quantisation as follows

$$c_q^\Delta(n) = \begin{cases} \lceil \frac{|c_q(n)| - \theta}{\Delta} \rceil, & \text{if } |c_q(n)| \geq \theta \\ 0, & \text{otherwise.} \end{cases}$$

The compression rate is given in bits-per-pixel (bpp) which is defined as

$$\text{bpp} = \frac{\text{Size of the file in bits}}{\text{Number of intensity pixels in a single channel}}.$$

At the reconstruction stage the indices $(\ell_x^n, \ell_y^n)$, $n = 1 \ldots k_q$ are recovered from the string of differences $\delta_q(n)$, $n = 2, \ldots, k_q$. The signs of the coefficients are read from the binary string. The quantised unsigned coefficients are read and transformed into real numbers as:

$$|\tilde{c}_q^\Delta(n)| = \Delta \cdot \tilde{c}_q^\Delta(n) + (\theta - \Delta/2) \quad n = 1 \ldots k_q.$$

The codec for reproducing the examples in the next sections has been made available on [52].

6.1 Numerical Example III

The relevance to image compression of the achieved sparsity by dct cross color transformation is illustrated in this section by comparison with results yielded by the compression standards JPEG, WebP, and JPEG2000, on the 15 images in Table [5]. These are typical images, used for
compression tests, available in ppm or png format. The first 9 images are classic test images taken from [53]. The last six images are portions of 1024 × 1024 × 3 pixels from very large high resolution images available on [54].

All the results have been obtained in the MATLAB environment (version R2019a), using a machine with CPU Intel(R) Core(TM) i7-3520M RAM 8GB CPU @ 2.90GHz. For the image approximation the HBW-OMP2D method was implemented with a C++ MEX file. All the channels and all the images were partitioned into blocks of size 16 × 16. The approximation times to produce the results in Table 4 are displayed in the last column of Table 3.

For realising the comparison we proceed as follows: we set the required value of PSNR as that produced by JPEG at quality=95 and tune compression with the other methods to produce the same PSNR. In our codec the tuning is realised by approximating the image up to PSNR\_o = 1.025 \cdot PSNR (where PSNR is the targeted quality) and setting the quantisation parameter \Delta so as to reproduce the targeted quality. For compression with JPEG and JPEG2000 we use
the MATLAB `imwrite` command. The comparison with WebP was realised using the software for Ubuntu distributed on [55].

| No | Image   | Size        | time (secs) |
|----|---------|-------------|-------------|
| 1  | Lenna   | 512 × 512 × 3 | 1.8         |
| 2  | Goldhill| 576 × 720 × 3 | 3.8         |
| 3  | Barbara | 576 × 720 × 3 | 3.2         |
| 4  | Baboon  | 512 × 512 × 3 | 2.7         |
| 5  | Zelda   | 576 × 784 × 3 | 2.5         |
| 6  | Sailboat| 512 × 512 × 3 | 1.8         |
| 7  | Boy     | 512 × 768 × 3 | 3.8         |
| 8  | Jupiter | 1072 × 1376 × 3 | 2.9      |
| 9  | Saturn  | 1200 × 1488 × 3 | 2.6     |
| 10 | Building| 1024 × 1024 × 3 | 9.5     |
| 11 | Cathedral| 1024 × 1024 × 3 | 7.6     |
| 12 | Flower  | 1024 × 1024 × 3 | 3.6     |
| 13 | Spider-web | 1024 × 1024 × 3 | 3.8     |
| 14 | Bridge  | 1024 × 1024 × 3 | 8.5     |
| 15 | Deer    | 1024 × 1024 × 3 | 5.8     |

Table 3: Test Images. The last column gives the approximation times to produce the results in Table 4.

7 Conclusions

The application of a cross color transformation for enhancing sparsity in the atomic decomposition of RGB images has been proposed. It was demonstrated that the effect of the transformation is to re-distribute the most significant values in the dwt of the 2D channels. As a result, when approximating the arrays by disregarding the less significant entries, the quality of the reconstructed image improves with respect to disabling the cross color transformation. Four transformations have been considered: (i) a 3 point dct, (ii) the reversible YCbCr color space transform, (iii) the PC transform, (iv) a transformation learned from an independent set of images.

The quality of the image approximation was improved further by approximating the transformed arrays by an atomic decomposition using a separable dictionary and the greedy pursuit strategy HBW-OMP2D. The dct was singled out as the most convenient cross color transformation for approximating RGB color images in the wavelet domain.

The approximation approach has been shown to be relevant for image compression. By means of a simple coding strategy the achieved compression considerably improves upon the compression standards JPEG and WebP. On a set of 15 typical test images the results are at least as good as those produced by JPEG2000. We feel confident that the findings communicated here will motivate the consideration of dct as a cross color transformations for other image processing applications which also benefit from the sparsity of a representation.
Table 4: Compression rate (bpp) corresponding to JPEG (bjp), WebP (bwb), JPEG2000 (bjp2) and the proposed sparse representation (bsr), for the values of PSNR given in the 2nd column.

| 1 | 35.9 | 3.28 | 2.21 | 1.40 | 1.37 |
|---|------|------|------|------|------|
| 2 | 36.6 | 3.63 | 2.47 | 2.11 | 2.01 |
| 3 | 37.2 | 3.67 | 2.80 | 1.91 | 1.77 |
| 4 | 28.8 | 5.80 | 4.66 | 2.74 | 2.48 |
| 5 | 39.3 | 2.61 | 1.81 | **1.08** | **1.07** |
| 6 | 30.9 | 4.49 | 3.21 | 1.62 | **1.51** |
| 7 | 32.6 | 4.34 | 3.07 | 2.36 | **2.22** |
| 8 | 48.2 | 0.60 | 0.51 | 0.24 | 0.20 |
| 9 | 49.0 | 0.34 | 0.36 | 0.15 | **0.12** |
| 10 | 37.4 | 3.41 | 2.35 | 1.84 | **1.75** |
| 11 | 38.5 | 2.84 | 1.83 | 1.26 | **1.20** |
| 12 | 41.5 | 1.78 | 1.08 | 0.56 | **0.53** |
| 13 | 34.5 | 3.57 | 2.48 | 1.61 | **1.48** |
| 14 | 45.0 | 1.55 | 1.10 | **0.58** | **0.57** |
| 15 | 30.9 | 3.76 | 2.59 | 1.04 | **0.90** |

References

[1] S. Mallat and Z. Zhang, “Matching pursuit with time-frequency dictionaries,” *IEEE Trans. Signal Process.*, Vol (41,12) 3397–3415 (1993).

[2] S. Chen, D. Donoho and M. Saunders “Atomic Decomposition by Basis Pursuit”, *SIAM Review*, 129–159 (2001).

[3] S. Mallat, A Wavelet Tour of Signal Processing: The Sparse Way, Academic Press (2009).

[4] Y. Eldar, P. Kuppinger and H. Bölscke, “Block-Sparse Signals: Uncertainty Relations and Efficient Recovery”, *IEEE Trans. Signal Process.*, 58, 3042–3054 (2010).

[5] E. Candès and M. Wakin, “An introduction to compressive sampling”, IEEE Signal Processing Magazine 25, 21 – 30 (2008).

[6] J. Romberg, “Imaging via compressive sampling”, IEEE Signal Processing Magazine 25, 14 – 20 (2008).

[7] R. Baraniuk, “More Is less: Signal processing and the data deluge”, *Science* 331, 717 – 719 (2011).

[8] M.F. Duarte, S. Sarvotham, D. Baron, M. B. Wakin, and R.G. Baraniuk, “Distributed compressed sensing of jointly sparse signals”, Proc. Asilomar Conf. Signals, Syst. Comput. 1537–1541 (2005).

[9] R. Torkamani, H. Zayyani, and R. Sadeghizadeh, ‘Model-based decentralized Bayesian algorithm for distributed compressed sensing’, 95, (2021), 116212.
[10] Jua Xiao, Rui Zhao, and Kim-Man Lam, “Bayesian sparse hierarchical model for image denoising”, Signal Processing: Image Communication (2021), 116299

[11] J. Wright, Yi Ma, J. Mairal, G. Sapiro, T.S. Huang, and S. Yan, “Sparse Representation for Computer Vision and Pattern Recognition”, Proc. of the IEEE 98, 1031 – 1044 (2010).

[12] M. Elad, Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing, Springer (2010).

[13] Z. Zhang, Y. Xu, J. Yang, X. Li, and D. Zhang, “A survey of sparse representation: algorithms and applications”, IEEE access (2015).

[14] J. Mairal, M. Eldar, and G. Sapiro, “Sparse Representation for Color Image Restoration”, IEEE Trans. Image Proces., 17, 53 – 69 (2008).

[15] W. Dong, L. Zhang, G. Shi, and X. Li, “Nonlocally Centralized Sparse Representation for Image Restoration”, IEEE Trans. Image Proces., 22, 1620 – 1630 (2013).

[16] Zhenming Su, Simiao Zhu, Xin Lv, and Yi Wan, “Image restoration using structured sparse representation with a novel parametric data-adaptive transformation matrix”, Signal Processing: Image Communication, 52, 151–172 (2017).

[17] J. Wright, A. Y. Yang, and A. Ganesh, “Robust Face Recognition via Sparse Representation”, IEEE Trans. Pattern Analysis and Machine Intelligence, 31, 210 – 227 (2009).

[18] XT. Yuan, X. Liu, and S. Yan, “Visual Classification With Multitask Joint Sparse Representation”, IEEE Trans. Image Proces., 21, 4349 – 4360 (2012).

[19] D. Heinsohn, E. Villalobos, L. Prieto, and D. Mery “Face recognition in low-quality images using adaptive sparse representations”, Image and Vision Computing, 85, 46–58 (2019).

[20] K. Raja, R. Ramachandra, and C. Busch “Collaborative representation of blur invariant deep sparse features for periocular recognition from smartphones”. Image and Vision Computing, 101, 46–58 (2020).

[21] J. Yang, J. Wright, and T. Huang, “Image Super-Resolution via Sparse Representation,” IEEE Trans. Image Proces., 19, 2861 – 2873 (2010).

[22] Y. Zhang, J. Liu, W. Yang, and Z. Guo, “Image Super-Resolution Based on Structure-Modulated Sparse Representation”, IEEE Trans. Image Proces., 9, 2797 – 2810 (2015).

[23] Chaopeng Zhang, Weirong Liu, Jie Lui, Chaorong Liu, and Changhong Shi, “Sparse representation and adaptive mixed samples regression for single image super-resolution”, Signal Processing: Image Communication, 67, 79–89 (2018).

[24] Xuesong Li, Guo Cao, Youqiang Zhang, Ayesha Shafique, and Peng Fu, “Combining synthesis sparse with analysis sparse for single image super-resolution”, Signal Processing: Image Communication, 83 115805 (2020).

[25] C. F. Caiafa and A. Cichocki, “Computing sparse representations of multidimensional signals Using Kronecker Bases”, Neural computation, 25, 186 – 220 (2013).
A. Cichocki, D. Mandic, L. De Lathauweri, G. Zhou, Q. Zhao, C. Caiafai, and H. A. Phan, “Tensor decompositions for signal processing applications: From two-way to multiway component analysis”, IEEE Signal Processing Magazine, 32, 145–163 (2015).

Q. Dai, S. Yoo, and A. Kappeler “Sparse Representation-Based Multiple Frame Video Super-Resolution”, IEEE Trans. Image Proces., 26, 2080 – 2095 (2017).

H. S. Mousavi and V. Monga, “Sparsity-based Color Image Super Resolution via Exploiting Cross Channel Constraints”, IEEE Trans. Image Proces., 26, 5094 – 5106 (2017).

L. Rebollo-Neira and D. Whitehouse, “Sparse representation of 3D images for piece-wise dimensionality reduction with high quality reconstruction”, Array, 1, doi = 10.1016/j.array.2019.100001 (2019).

U. Grenander and A. Srivastava, “Probability models for clutter in natural images”, IEEE Trans. Pattern Anal. Mach. Intell., 23, 424–429 (2001) 10.1109/34.917579

L. Rebollo-Neira, R. Matiol, and S. Bibi, “Hierarchized block wise image approximation by greedy pursuit strategies,” IEEE Signal Process. Letters, 20, 1175–1178 (2013).

L. Rebollo-Neira, “Effective sparse representation of X-Ray medical images”, International Journal for Numerical Methods in Biomedical Engineering, in press (2017).

L. Rebollo-Neira, “A competitive scheme for storing sparse representation of X-Ray medical images”, PLoS ONE, 13(8): e0201455. https://doi.org/10.1371/journal.pone.0201455 (2018)

G. Schaefer and M. Stich, “UCID: an uncompressed color image database”, Proceedings Volume 5307, Storage and Retrieval Methods and Applications for Multimedia 2004; (2003) https://doi.org/10.1117/12.525375.

D. S. Taubman and M. W. Marcellin, JPEG2000 Image Compression, Fundamentals, Standards and Practice, Kluwer Academic Publishers (2002).

https://www2.eecs.berkeley.edu/Research/Projects/CS/vision/bbsd/ (Last access May 2021).

E. Gershikov and M. Porat, “On color transforms and bit allocation for optimal subband image compression”, Signal Processing: Image Communication, 22, 1–18 (2007).

K. Kreutz-Delgado, J. F. Murray, B. D. Rao, K. Engan, Te-Won Lee, and T. J. Sejnowski, “Dictionary Learning Algorithms for Sparse Representation”, Neurocomputing, 15, 349–396 (2003).

M. Aharon, M. Elad, and A. Bruckstein, “K-SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation”, IEEE Trans. Signal Process. 54, 4311–4322 (2006).

R. Rubinstein, M. Zibulevsky, and M. Elad, “Double Sparsity: Learning Sparse Dictionaries for Sparse Signal Approximation”, IEEE Trans. Signal Process. 58, 1553–1564 (2010).
[41] I. Tošić and P. Frossard, “Dictionary Learning: What is the right representation for my signal?”, *IEEE Signal Processing Magazine*, **28**, 27–38 (2011).

[42] J. Zepeda, C. Guillemot, and E. Kijak, “Image Compression Using Sparse Representations and the Iteration-Tuned and Aligned Dictionary”, *IEEE Journal of Selected Topics in Signal Processing*, **5**, 1061–1073 (2011).

[43] S. Hawe, M. Seibert, and M. Kleinsteuber, “Separable dictionary learning”, Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition, 438–445 (2013).

[44] M. Srinivas, R. R. Naidu, C.S. Sastry, and C. Krishna Mohana, “Content based medical image retrieval using dictionary learning”, *Neurocomputing*, **168**, 880–895 (2015).

[45] B. Wen, S. Ravishankar, and Y. Bresler, “Structured Overcomplete Sparsifying Transform Learning with Convergence Guarantees and Applications”, *International Journal of Computer Vision*, **114**, 137–167 (2015).

[46] L. Rebollo-Neira, “Cooperative greedy pursuit strategies for sparse signal representation by partitioning”, *Signal Processing*, **125**, 365–375 (2016).

[47] Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad, “Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition,” *Proc. of the 27th ACSSC*, 1, 40–44 (1993).

[48] L. Rebollo-Neira, J. Bowley, A. Constantinides and A. Plastino, “Self contained encrypted image folding”, *Physica A*, **391**, 5858–5870 (2012).

[49] L. Rebollo-Neira and D. Lowe, “Optimized orthogonal matching pursuit approach”, *IEEE Signal Process. Letters*, **9**, 137–140 (2002).

[50] L. Rebollo-Neira, M. Rozložník, and P. Sasmal, “Analysis of the Self Projected Matching Pursuit Algorithm”, *Journal of The Franklin Institute* 8980–8994 (2020).

[51] K. Skretting, https://www.mathworks.com/matlabcentral/fileexchange/2818-huffman-coding-and-arithmetic-codin (Last access May 2021).

[52] http://www.nonlinear-approx.info/examples/node015.html (Last access May 2020).

[53] https://www.hlevkin.com/06testimages.htm (Last access May 2021).

[54] https://imagecompression.info/test_images (Last access May 2021).

[55] https://developers.google.com/speed/webp (Last access May 2021).