[Array 1-2 (2019) 100001](https://doi.org/10.1016/j.array.2019.100001)

Contents lists available at [ScienceDirect](http://www.sciencedirect.com/science/journal/25900056)

Array

journal homepage: [www.elsevier.com/journals/array/2590-0056/open-access-journal](http://www.elsevier.com/journals/array/2590-0056/open-access-journal)

Sparse representation of 3D images for piecewise dimensionality reduction with high quality reconstruction

Laura Rebollo-Neira [\*](#_bookmark0), Daniel Whitehouse

*Mathematics Department, Aston University, B47ET, Birmingham, UK*

A R T I C L E I N F O

*Keywords:*

Image representation with dictionaries Greedy pursuit algorithms

A B S T R A C T

Sparse representation of 3D images is considered within the context of data reduction. The goal is to produce high quality approximations of 3D images using fewer elementary components than the number of intensity points in the 3D array. This is achieved by means of a highly redundant dictionary and a dedicated pursuit strategy especially designed for low memory requirements. The benefit of the proposed framework is illustrated in the first instance by demonstrating the gain in dimensionality reduction obtained when approximating true color images as very thin 3D arrays, instead of performing an independent channel by channel approximation. The full power of the approach is further exemplified by producing high quality approximations of hyper-spectral images with a reduction of up to 371 times the number of data points in the representation.

1. Introduction

Sparse representation of 2D images has been a subject of extensive research in the last fifteen years [[1](#_bookmark13)–[3]](#_bookmark13). Applications which benefit from sparsity range from image restoration [[4,5]](#_bookmark14) and classification [[6](#_bookmark15)–[8]](#_bookmark15) to feature extraction [[9,10]](#_bookmark16) and super resolution reconstructions [[11,12]](#_bookmark17). While sparse representation of 3D arrays has received less attention, the advantage of modeling these arrays as a superposition of 3D elementary components is recognized in previous publications [[13](#_bookmark18)–[16]](#_bookmark18).

At present, the most widely used multichannel images in every day life are true color images. The simplest way of sparsely representing these images is channel by channel, or adding constraints of correlation across colors [[4,17]](#_bookmark14). However, as demonstrated in this work, sparsity in the representation of true color images can increase substantially if the approximation is realized by means of 3D elements taken from a highly redundant dictionary. The effect is of course more pronounced for arrays involving more channels, such as hyper-spectral images.

From a practical view point, the current drawbacks of 3D sparse modeling using a large dictionary are (i) storage requirements and (ii) the complexity of the concomitant calculations. In this paper we propose a method which, by addressing (i) leaves room for possible high perfor- mance implementations using Graphics Processing Unit (GPU) pro- gramming. While the approach is illustrated using Central Processing Unit (CPU) programming, the storage requirements are shown to fit within 48 Kb's of fast access shared memory of a GPU when the

approximation of a 3D image is realized with a partition block size of 8 ×8 × 8 and with a separable dictionary of redundancy 125.

The main contributions of the paper are listed below.

* The low memory implementation of the Orthogonal Matching Pursuit (OMP) strategy, called Self Projected Matching Pursuit (SPMP) [[18]](#_bookmark20) is dedicated to operating in 3D (SPMP3D) with separable dictionaries. This technique delivers an iterative solution to the 3D least squares problem which requires much less storage than direct linear algebra methods. It could therefore be also applied with any other of the pursuit strategies that include a least squares step [[14,19](#_bookmark19)–[22]](#_bookmark19).
* The C++ MEX file for the SPMP3D method has been made available on a dedicated website [[23]](#_bookmark21). All the scripts for reproducing the results of the paper in the MATLAB environment have also been placed on that website.
* Remarkable reduction in the dimensionality of the representation of true color images and hyper-spectral images, with high quality reconstruction, is demonstrated using highly redundant and highly coherent separable dictionaries.

The results suggest that the method may be of assistance to image processing applications which rely on a transformation for data reduction as a first step of further processing. For examples of relevant applications we refer to Refs. [[24](#_bookmark22)–[28]](#_bookmark22).

\* Corresponding author.

*E-mail address:* [rebollol@aston.ac.uk](mailto:rebollol@aston.ac.uk) (L. Rebollo-Neira).

<https://doi.org/10.1016/j.array.2019.100001>

Received 10 February 2019; Received in revised form 17 April 2019; Accepted 29 May 2019

Available online 20 June 2019

2590-0056/© 2019 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Notational convention

*D* = *D x* *D y* *D z*of three 1D dictionaries *D x* = {d*x* ∈ R*Nx*

*Mx*

}*m*=1

}

,

*D y* = {d*y* ∈ R*Ny My*

*m*

}

, and *D z* = {d*z* ∈ R*Nz Mz*

, with *MxMy Mz* = *M*.

R represents the set of real numbers. Boldface letters are used to

*m m*=1

*m m*=1

indicate Euclidean vectors, 2D and 3D arrays. Standard mathematical fonts indicate components, e.g., d ∈ R*N* is a vector of components *d*(*i*)∈ R; *i* = 1;…;*N*. The elements of a 3D array I ∈ R*Nx* ×*Ny* ×*Nz* are indicated as *I*(*i*;*j*;

*m*); *i* = 1;…; *Nx*; *j* = 1;…; *Ny* ; *m* = 1;…; *Nz*. Moreover, for each *m*-value

I*m* ∈ R*Nx* ×*Ny* stands for the 2D array of elements *Im*(*i*;*j*) = *I*(*i*;*j*;*m*); *i* = 1;…;

*Nx*; *j* = 1;…;*Ny* , which, when not leaving room for ambiguity will also be

represented as *I*(:; :; *m*). The transpose of a matrix, Gsay, is indicated as

For computational purposes the 1D dictionaries are stored as three matrices D*x* ∈ R*Nx* ×*Mx* , D*y* ∈ R*Ny* ×*My* and D*z* ∈ R*Nz* ×*Mz* . Suppose now that a 3D array I ∈ R*Nx* ×*Ny* ×*Nz* is to be approximated by an atomic decomposition

of the form

I*k* = X *c*(*n*)d*xx* d*yy* d*z z* ; (3)

*k*

ℓ*n*

ℓ*n*

ℓ*n*

*n*=1

GT.

The inner product between 3D arrays, say I ∈ R*N* ×*N* ×*N* and G

*x*

*y*

*z*

where for *n* = 1;…;*k*the atoms d*xx* , d*yy* and d*zz* are selected from the given

R*Nx* ×*Ny* ×*Nz* , is given as:

∈

*n*

*n*

*n*

ℓ ℓ ℓ

*N Ny Nz*

1D dictionaries. The common step of the techniques we consider for constructing approximations of the form (3) is the stepwise selection of

the atoms in the atomic decomposition. On setting *k* = 1and I0 = 0at

DG I

; ⟩3D =

*G*(*i*; *j*; *m*)*I*(*i*; *j*; *m*).

X*x* X X

*i*=1

*j*=1 *m*=1

*x y z*

iteration *k* the algorithm selects the indices ℓ*k* , ℓ*k* and ℓ*k* as follows

For G ∈ R*Nx* ×*Ny* ×*Nz* with tensor product structure, i.e. for G = g*x* g*y*

g*z*, with g*x* ∈ R*Nx* ; g*y* ∈ R*Ny* and g*z* ∈ R*Nz* , we further have

ℓ*x*; ℓ*y*; ℓ*z* = arg max d*x* d*y* d*z*; R*k*—1⟩

; (4)

G; I⟩3D

*Nz*

= ⟨g*x*; I*m*g*y gz*(*m*)= ⟨p; g*z* ; (1)

X

*m*=1

*k k k*

*n i s* 3D

*n*=1;…;*M*

*x*

*i*=1;…;*My s*=1;…;*Mz*

where for each value of *m* the vector I*m*g*y* ∈ R*Nx* arises by the standard matrix-vector multiplication rule and p ∈ R*Nz* is given by its components

*p*(*m*) = ⟨g*x*; I*m*g*y* ⟩; *m* = 1; …; *Nz*. Note that ⟨p; g*z*⟩indicates the Euclidean

inner product in 1D, i.e.

*Nz*

X

p; g*z*⟩= *p*(*m*)*gz*(*m*).

*m*=1

The definition (1)induces the norm I3D = p⟨ﬃﬃIﬃﬃ;ﬃﬃIﬃﬃ⟩ﬃﬃ3ﬃﬃDﬃﬃ.

1. Sparse representation of multi-channel images

Suppose that a 3D image, given as an array I ∈ R*Nx* ×*Ny* ×*Nz* of intensity pixels, is to be approximated by the linear decomposition

*k*

I = X *c*(*n*)Dℓ ; (2)

*k*

*n*

*n*=1

where each *c*(*n*)is a scalar and each Dℓ is an element of R*Nx* ×*Ny* ×*Nz* to be selected from a set, *D* = {D*n*}*M* , called a ‘dictionary’.

*n*

*n*=1

A sparse approximation of I ∈ R*Nx* ×*Ny* ×*Nz* is an approximation of the form (2) such that the number *k* of elements in the decomposition is significantly smaller than *N* = *NxNy Nz*. The terms in the decomposition

(2) are taken from a large redundant dictionary, from where the elements Dℓ*n* in (2), called ‘atoms’, are chosen according to an optimality criterion. Within the redundant dictionary framework for approximation, the problem of finding the sparsest decomposition of a given multi-channel image can be formulated as follows: *Given an image and a dictionary,*

with R*k*—1 = I — I*k*—1. It is the determination of the coefficients *c*(*n*); *n* =

1; …; *k*in (3) that gives rise to pursuit strategies which go with different

names.

* 1. *Matching Pursuit in 3D (MP3D)*

The MP approach in 3D would simply calculate the coefficients in (3)

as

*c*(*n*)= Dd*xx* d*yy* d*z z* ; R*n*—1⟩3D; *n* = 1; …; *k*. (5)

ℓ*n*

ℓ*n*

ℓ*n*

The main drawback of the MP method is that it may select linearly dependent atoms. Moreover, that approximation is not stepwise optimal because at iteration *k* the coefficients (5) do not minimize the norm of the residual error. The pursuit strategy that overcomes these limitations is the so called OMP [[30]](#_bookmark24).

* 1. *Orthogonal Matching Pursuit in 3D*

The implementation of OMP in 3D (OMP3D) we describe here is the 3D extension of the implementation of OMP in 2D given in Ref. [[32]](#_bookmark26). An alternative algorithm called Kronecker-OMP, which is based on the Tucker representation of a tensor, is discussed in Ref. [[14]](#_bookmark19). Our algorithm is based on adaptive biorthogonalization and Gram-Schmidt orthogo- nalization procedures, as proposed in Ref. [[31]](#_bookmark25) for the one dimensional case.

In order to ensure the coefficients *c*(*n*); *n* = 1;…;*k*involved in (3) are

*approximate the image by the* ‘*atomic decomposition*’ *(*2*) such that the number k of atoms is minimum.* Unfortunately, the numerical minimization of the number of terms to produce an approximation up to a desired

error, involves a combinatorial problem for exhaustive search. Hence, the

*k* 2 3D

fulfill that

such that ¨R ¨

*k*

= ⟨R*k*;R*k*⟩3D

is minimum, the decomposition (3) should

solution is intractable. Consequently, instead of looking for the sparsest

I*k* = X *c*(*n*)d*xx* d*yy* d*z z* = bPV I; (6)

solution, one looks for a ‘satisfactory solution’, i.e., a solution such that the number of *k*-terms in (2) is considerably smaller than the image dimension. For 2D images this can be effectively achieved by greedy

*n*=1

where

span{d*xx* d*yy* d*zz k*

PV*k*

ℓ*n* ℓ*n* ℓ*n k*

is the orthogonal projection operator onto V*k* =

b

}

. This is ensured by requiring that R*k* = I —

pursuit strategies in the line of the Matching Pursuit (MP) [[29]](#_bookmark23) and OMP

[[30]](#_bookmark24) methods, if dedicated to 2D separable dictionaries [[14,18,32,33]](#_bookmark19).

Within a tensor product framework the consideration of OMP in 3D is

ℓ*n*

ℓ*n*

ℓ*n n*=1

PbV I. The required representation of PbV is of the form PbV I = P*k* A*n*⟨

*k*

*k*

*k*

*n*=1

B*k*; I⟩3D, where each A*n* ∈ R*Nx* ×*Ny* ×*Nz* is an array with the selected atoms

natural.

Let's assume that a 3D dictionary is obtained as the tensor product

*n*

A*n* = d*xx* d*yy* d*xz* . The concomitant biorthogonal reciprocal set B*k*;

*n*

*n*

*n*

ℓ ℓ ℓ *n*

*n* = 1; …; *k*comprises the unique elements of R*Nx* ×*Ny* ×*Nz* satisfying the

conditions:

1. ⟨A*n*; B*k* ⟩3D = *δn*;*m* = 8< 1 if *n* = *m*

*m*

0 if *n* /= *m*.

where the coefficients *a*(*n*); *n* = 1;…;*k*are arbitrary numbers. Every array

I ∈ R*Nx* ×*Ny* ×*Nz* can be expressed as

*k*

I = ~I + R~ . (8)

1. V*k* = span{B*k*}*k* . :

*n n*=1

For ~I*k*to be the optimal representation of Iin V =

*k*

span{d*xx* d*yy* d*zz k* , in the sense of minimizing the norm of the

ℓ*n*

Thus, the coefficients *c*(*n*); *n* = 1; …; *N*in (6) which guarantee mini-

ℓ*n* ℓ*n*

R~

}

*n*=1

R~ = 0. The SPMP3D method fulfills

mum norm of the residual error are calculated as

residual , it should be true that PbV*k*

*k n*

*c*(*n*)= B ; I⟩3D; *n* = 1; …; *k*.

The required arrays B*k*; *n* = 1;…;*k*should be upgraded and updated to account for each newly selected atom. Starting from *k* = 1; R0 = I, B1 =

*n*

1

this property by approximating R~ in V*k*, via the MP method, and sub-

tracting that component from R~ . The following algorithm describes the

whole procedure.

Starting from *k* = 0and R0 = I, at each iteration, implement the steps below.

W1 = A1 = d*xx* d*yy* d*zz* , where

ℓ1 ℓ1 ℓ1

indices (ℓ*x*;ℓ*y* ;ℓ*z* ). Save this triple in the array *L*(*k*;1 : 3) = (ℓ*x*;ℓ*y*;ℓ*z* );

i) Increase *k* ← *k* +

1and apply the criterion (4) for selecting the triple of

and set

*k k k*

*k k k*

ℓ*x* ; ℓ*y* ; ℓ*z* = arg max

d*x* d*y* d*z*; R*k*—1⟩

*c*(*k*)= Dd*xx* d*yy* d*z z* ; R*k*—1⟩

1 1 1

*n*=1;…;*Mx*

*n i s* 3D

ℓ*k* ℓ*k* ℓ*k* 3D

*i* = 1; …; *My*

;

*s* = 1; …; *Mz*

at iteration *k* + 1the indices ℓ*x*

; ℓ*y*

; ℓ*z*

corresponding to the new

and implement the update of the residue R*k* =

d*zz* as follows:For *s* = 1; …*Nz*calculate

ℓ

*k*

R*k*—1 —

*c*(*k*)d*xx*

*k*

ℓ

d*yy*

ℓ*k*

*k*+1

are selected as in (4). The required

*x y*

atom A*k*+1 = d *x*

*k*+1

d *y*

*k*+1

d*zz*

*k*+1

ℓ

ℓ

ℓ

*k*+1

*k*+1

ΔR*k*

*s* R*k*—1

*s c k* d*x* d*y* T*dz s*

reciprocal set B*k*+1; *n* = 1;…;*k* + 1is adapted and upgraded by extending the recursion formula given in Ref. [[31]](#_bookmark25) as follows.

ℓ

ℓ

ℓ

*n*

( :; :; )=

( :; :; )— ( ) *x*

*k*

*y k*

*z* ( );

*k*

to update R*k* as

B*k*+1 = B*k* — B*k*+1 A*k* 1 ; B*k* ⟩

*n*

*n*

*k*+1

+

where

*n* 3D

; *n* = 1; …; *k*;

R*k* = R

*k*—1

— ΔR*k*.

B*k*+1 = W*k* 1 ,  W*k* 1  2 ;

*k*+1

+

+

3D

ii) Given the indices *L*(*n*;1 : 3) = (ℓ*x*;ℓ*y* ;ℓ*z* ); *n* = 1;…;*k*of the previously

with

*n n n*

selected atoms, and a tolerance

*k*

— X  W W ; A

W = A *n* ⟩ ;

*k*+1

*k*+1

W

2

*n*

*k*+1 3D

⟨

*ε* for the projection error, realize the

orthogonal projection up to that error as follows. Set *j* = 1, R~ 0 =

*k*

R and at iteration *j* apply the steps a) - c) below.

*n*=1

*n* 3D

including, for numerical accuracy, the re-orthogonalization step:

1. For *n* = 1; …; *k*evaluate

*α*(*n*) = Dd*x* d*y* d*z* ; R~ *j*—1 ⟩ ; (9)

ℓ

ℓ

ℓ

3D

W ← W X W*n* W W

*k*

2

*k*+1

*k*+1 —

⟨

*n*=1 ||W*n* ||3D

*n*;

*k*+1 ⟩3D.

*x y z*

*n n n*

and single out the value *k*\*such that

Although the image approximation is carried out by partitioning the images into relatively small 3D blocks, memory requirements of the OMP3D method are high. Indeed, the above are 2(*k* + 1)nonseparable

arrays each of dimension *N* = *NxNy Nz*which need to be stored in double

*k*\* = arg max *α*(*n*) . (10)

*n*=1;…;*k*

The value *k*\* signalizes the indices ℓ*x* ; ℓ*y* ; ℓ*z* corresponding to the

*k k k*

\* \* \*

precision. Hence, we consider next a low memory implementation of the orthogonal projection step, which avoids having to store the arrays W*n*; *n* = 1; …; *k*and B*k*; *n* = 1; …; *k*and fully exploits the separability of the

*n*

already selected atoms with maximum correlation with the residual R~ *j*—1.

*c*(*k*\*) ← *c*(*k*\*)+ *α*(*k*\*)

dictionary.

and for *s* = 1; …; *Nz*evaluate

* 1. *Self Projected Matching Pursuit in 3D (SPMP3D)*

*k*

ΔR~ *j* ( :; :; *s*)= *α*(*k*\*) d*x*

d*y*

*y*

ℓ*k*\*

T *z*

ℓ*z*\*

*k*

*d*

(*s*)

The Self Projected Matching Pursuit (SPMP) methodology was introduced in Ref. [[18]](#_bookmark20) and conceived to be used with separable dictio- naries in 2D (SPMP2D). Because the technique is based on calculations of inner products, it can be easily extended to operate in 3D (SPMP3D).

ℓ*x*\*

Suppose that at iteration *k* the selection process has chosen the atoms

to update the residual R~ *j* as

R~ *j* = R~ *j*—1 — ΔR~ *j* .

labelled by the triple of indices {ℓ*x*; ℓ*y* ; ℓ*z k*

and let ~I*k*be the atomic

1. If |*α*(*k*\*)| < εstop. Otherwise update the coefficient

decomposition

*k*

*n n n* }*n*=1

This step subtracts from the residual a component in V*k*and add that

*k*

~I*k* = X *a*(*n*)d*x*

ℓ

ℓ

ℓ

*x n*

*y n*

*z n*

*n*=1

d*y B* d*z* ; (7)

component to the approximation. ~I

1. Increase *j* ← *j* + 1and repeat the steps a) - c) to keep subtracting components of R~ *j* in V*k*until iteration, *J* say, for which the stopping criterion b) is met. This criterion indicates that, up to tolerance *ε*, the

residual has no component in V*k*so that one can set R*k* = R~ *J*—1.

Continue with steps i) and ii) to keep enlarging V*k*until, for a required tolerance error *ρ*, the condition R*k* < *ρ*is reached.

3*D*

Remark 1. For each fixed value *k* the rate of convergence

Algorithm 3Implementation of the self projection steps a) - c). Procedure [R~ ; c] = Proj3D(R; D*x* ; D*y* ; D*z* ; L; c; ε; MaxJ).

Input: As in Algorithm 2, plus the coefficients of the atomic decomposition c, a tolerance parameter *ε* for the numerical error of the projection, and a maximum

number of permitted iterations, MaxJ.

Output: Orthogonal residual R~ . Coefficients ~c of the optimized atomic decomposition.

for *j* = 1 : MaxJ

{Selection of atoms using Algorithm 2}

[*α*\*; *k*\*] = SelTrip(R; D*x* ; D*y* ; D*z* ; L)

{Check stopping criterion}

limI —

*j*→∞

R~ *j* =

bPV*k*

if |*α*\*| < ε then

stop end if

through the steps a) - c) above is given in Ref. [[34]](#_bookmark28) for the one dimen- sional case. The proof for 3D is identical to that proof, because a 3D array can be represented as a long 1D vector. What varies is the implementa- tion. A vectorized version of the algorithm would not be applicable in this

I

{Update the coefficients}

*c*(*k*\*) ← *c*(*k*\*)+ *α*\*

{Update the residual}

for *s* = 1 : *Nz* then

R(:; :; *s*) ← R(:; :; *s*)— *α*\*(d*xx* )T*R*(:; :; *s*)d*yy dz* (*s*)

ℓ ℓ*z*\*

context.

end for end for

*k*\* ℓ*k*\* *k*

* + 1. *Implementation details*

{For clarity in the description only, we re-name here the residual and coefficients}

R = R; c = c

The bulk of the computational burden in the SPMP3D method lies in

the realization of the selection of atoms (4). Algorithm 1 outlines a

procedure implementing the process. It should be stressed once again that the algorithm is designed to use as little memory as possible, rather than to reduce complexity.

Algorithm 1 Implementation of the selection of atoms (c.f. (4)) Procedure [*α*; ℓ*x*; ℓ*y* ; ℓ*z*] = Sel3DAtom(R; D*x* ; D*y* ; D*z* )

Input: 3D array R, matrices D*x* , D*y* D*z* the columns of which are the atoms in the corresponding dictionaries.

Output: selected indices ℓ*x*; ℓ*y* ; ℓ*z*, and *α* = ⟨d*xx* d*yy* d*z z* ; R⟩3D

Due to computational complexity and memory requirements, pursuit strategies using general dictionaries can only be implemented on an image partitioned into small blocks.

We consider nonoverappling blocks. The approximation of each block is carried out independently of the others. When the approximation of all the blocks is concluded, these are assembled together to produce the approximation of the whole image. While the sparsity results yielded by

{Initiate the algorithm}

ℓ ℓ ℓ

the OMP3D and the SPMP3D methods are theoretically equivalent, we

(*Nz* ; *Mz* )= size(D*z* ) ; *Mx* = size(D*x* ; 2); *My* = size(D*y* ; 2) *q* = zeros(*Mx* ; *My* )*α* = 0

for *m* = 1 : *Mz* do

*q*(:; :) = 0

for *s* = 1 : *Nz*

*q*(:; :) = *q*(:; :)+ DT*R*(:; :; *s*)D*y dz* (*s*)

*x m*

end for

{Realize (4) by finding the partial maximum, and its argument, for each *m*-plane}

[*l*1 ; *l*2 ; *q*~] = max(|*q*(:; :)|)

if*q*~ > *α*then

*α* = *q*~; ℓ*x* = *l*1 ; ℓ*y* = *l*2 ; ℓ*z* = *m*

end if

end for

At iteration *k* the outputs of Algorithm 1 are saved as *c*(*k*)= *α* and *L*(*k*;

1 : 3) = (ℓ*x*;ℓ*y* ;ℓ*z*). The implementation details for selecting the triple of indices at the projection step are given in Algorithm 2. This is used in Algorithm 3 for the realization of the actual projection to recalculate the coefficients in the atomic decomposition.

Algorithm 2Selection of the triple of indices from the reduced dictionary (c.f. (10)) Procedure [*α*\*; *k*\*] = SelTrip(R; D*x* ; D*y* ; D*z* ; L)

Input: As in Algorithm 1 plus the array L, with the triple of indices

*L*(*n*; 1 : 3)= (ℓ*x*; ℓ*y* ; ℓ*z* ); *n* = 1…*k*

have seen that the latter implementation is much more economic in terms of storage demands. As discussed in Remark 2 below, this feature makes the SPMP3D algorithm suitable for possible GPU implementations using only the fast access shared memory. Assuming for simplicity in the no- tation that a 3D image is partitioned into cubes of size *N*3 and the dic-

tionaries *D x*, *D y* and *D z* are all of the same size *Nb* × *rNb*, where *r* > 1 is

*b*

the redundancy of the 1D dictionary, the SPMP3D algorithm storage needs are as follows.

1. Two *N*3 arrays for the intensity block in the image partition and the residual of the corresponding approximation.

*b*

1. Three matrices of size *Nb* × *rNb* for each dictionary, in case they are different.
2. A *r*2 × *N*2 array for the selection of indices in Algorithm 1.

*b*

1. A vector of *k* real numbers to store the coefficients of the atomic decomposition and *k* vectors of size 3 to store the indices of the atoms in the atomic decomposition. The value of *k* is the total number of atoms in the approximation of the block.

Since the stepwise complexity is dominated by the selection of indices

Output: \*

*n n n*

(c.f. (10)) to update the coefficients and

(c.f. (4)), within this setup it is O(*r*3 *N*5) and for true color images

*k* and the corresponding values of *α*

residual {Initiate the algorithm}

*α*\* = 0

for *n* = 1 : *k* do

*p* = 0

for *s* = 1 : *Nz* do

*b*

O(*r*3 *N*3).

*b*

Remark 2. By considering blocks of size 8 ×8 × 8 and dictionaries of redundancy *r* = 5 in each dimension, the above listed storage needs of

the SPMP3D algorithm comfortably fit the fast access shared memory of a

*x* T *y z*

*p* = *p* + (dℓ*x* ) *R*(:; :; *s*)dℓ*y d*ℓ*z* (*s*)

GPU in CUDA, which currently is 48 Kb. Indeed, in the worst-case sce-

*n n n*

end for 3

if |*p*| > |*α*\*| then

*k*\* = *n* and *α*\* = *p*

end if end for

nario (corresponding to an approximation of zero error using *k* = 8 atoms for the approximation of an 8 ×8 × 8 block) SPMP3D would require 38 Kb to store most of the arrays in double precision, except for those with the selected indices which contain integer numbers. This still leaves 10 Kb for temporary variables to be used within calculations.

* 1. *Mixed dictionaries*

A key factor for the success in the construction of sparse representa- tions is to have a good dictionary. While a number of techniques for

atoms h*i*; *i* = 1; …; 4 in the left graph of [Fig. 1](#_bookmark2) are generated by dis- cretization of linear B-spline functions of different support. For *m* = 1; 2; 3; 4 those functions are defined as follows:

learning dictionaries from training data have been proposed in the literature [[35](#_bookmark29)–[42]](#_bookmark29), they are not designed for learning large and highly coherent separable dictionaries. Nevertheless, previous works [[18,32,33,](#_bookmark20) [43]](#_bookmark20) have demonstrated that highly redundant and highly coherent

separable dictionaries, which are easy to construct, achieve remarkable

*x m*

*hm*(*x*)= 2 —

8

>>>< *x*

*m*

>

>:

if 0 ≤ *x* < *m*

if *m* ≤ *x* < 2*m*

(11)

levels of sparsity in the representation of 2D images. Such dictionaries are not specific to a particular class of images. A discrimination is only made to take into account whether the approximation is carried out in the pixel

intensity or in the wavelet domain.

0 otherwise.

The remaining prototypes, h5; h6 and h7, in the left graph of [Fig. 1](#_bookmark2) are generated taken the derivatives of the previous functions: *h*5(*x*) =

' ' '

As will be illustrated by the numerical examples in the next section, the approximation of the images we are considering are sparser when realized in the wavelet domain (wd). This entails the following steps:

(*h*2(*x*)) , *h*6(*x*)= (*h*3(*x*)) and *h*7 = (*h*4(*x*)) . The corresponding dictio- naries *D Hm* ; *m* = 1; …; 7 are built by discretization of the variable *x* in

(11) and sequential translation of one sampling point, i.e.,

*D H* = {*wh* (*n*)*hm*(*i* — *n*)|*Nx*; *i* = 1; …; *Nx*}*M*

; *m* = 1; …; 7;

* Apply a wavelet transform to each channel I*m*; *m* = 1; …; *Nz* to obtain *m m*

*n*=1

the arrays U*m*;*m* = 1;…;*Nz*. For the numerical examples we have used

the 9/7 Cohen-Daubechies-Feauveau biorthogonal wavelet transform

where the notation *hm*(*i* — *n*)|*Nx* indicates the restriction to be an array of size *Nx*. The numbers *wh* (*n*); *n* = 1; …; *M*; *m* = 1; …; 7 are normaliza-

[[44]](#_bookmark30). *m* *x*

* Approximate the array U ∈ R*Nx* ×*Ny* ×*Nz* exactly as it is done in the pixel domain (pd).
* Apply the inverse wavelet transform to the approximated planes to

tion factors. The dictionary *D* Lp arises by the union of the dictionaries

*D H* ; *m* = 1; …; 7 i.e., *D x* = ∪7 *D H* . The whole mixed dictionary

*m* Lp *m*=1 *m*

*D x* is finally formed as *D x* = *D x* ∪*D x*∪*D x* . For the other dimension

recover the approximated intensity channels.

pd

we take *D y*

pd

pd

= *D x* .

pd

C S Lp

The mixed dictionary we consider for the 2D approximation consists of two sub-dictionaries: A trigonometric dictionary, *D x* , which is the common sub-dictionary for the approximation in both domains, and a dictionary of localized atoms, which contains atoms of different shapes

*T*

For approximations in the wd we use the dictionary of localized atoms *D x* as proposed in Ref. [[33]](#_bookmark27), which is built by translation of the pro- totype atoms p*i*; *i* = 1; …; 7 in the right graph of [Fig. 1](#_bookmark2). Notice that p1 =

h1 and p3 = h5. The remaining prototypes are given by the vectors:

Lw

p2 = (1; 1; 0; 0; …; 0)⊥ ∈ R*Nx* ; p4 = (1; 1; 1; 0; …; 0)⊥ ∈ R*Nx* ; p5 =

when used in each domain.

*x* (—1; 1; 1; 0; …; 0)⊥ ∈ R*Nx* ; p

= (1; —1; 1; 0; …; 0)⊥ ∈ R*Nx* ; p =

The trigonometric dictionary is the union of the dictionaries *D* C and 6 7

*D x* defined below:

S

(—1; —1; 1; 0; …; 0)⊥ ∈ R*Nx* ; The corresponding dictionaries *D P* ; *m* = 1;

…; 7 are built as in the previous case by sequential translation of one

*m*

*x π*(2*i* — 1)(*n* — 1)

*Mx*

sampling point,

*D* C =

*wc*(*n*)cos

2*Mx*

; *i* = 1; …; *Nx*

*n*=1

*D* = *w*

(*n*)*p* (*i* — *n*) *N* ; *i* = 1; …; *N* }*M*

; *m* = 1; …; 7;

S 2*Mx*

*D x* =

*ws*(*n*)sin

*π*(2*i* — 1)(*n*)

; *i* = 1; …; *Nx*

*Pm*

*pm*

*m*

*x*

*x*

*n*=1

*n*=1

where the numbers *wpm* (*n*); *n* = 1; …; *M*; *m* = 1; …; 7 are normalization

factors. The dictionaries *D P* ; *m* = 1; …; 7 give rise to *D x* = ∪7

*Mx*

*D P* .

where *wc*(*n*) and *w* (*n*); *n* = 1; …; *M*

;

are normalization factors, and

*m* Lw

*x x* *x*

*i*=1 *m*

*s x* The latter generates the mixed dictionary *D* wd = *D* C∪*D* S∪*D x*

and

usually *Mx* = 2*Nx*. Thus, the trigonometric dictionary is constructed as *y x* Lw

*D x* = *D x* ∪*D x*.

*D* wd = *D* wd.

*T* C S

*x* The corresponding 2D dictionaries *D* pd = *D x*

*D y*

and

For approximations in the pd we add the dictionary, *D* Lp, which is *y*

pd pd

built by translation of the prototype atoms in the left graph of [Fig. 1](#_bookmark2). This type of dictionary is inspired by a general result holding for continuous spline spaces. Namely, that spline spaces on a compact interval can be spanned by dictionaries of B-splines of broader support than the corre- sponding B-spline basis functions [[45,46]](#_bookmark31). Thus, the first 4 prototype

wd

*D* wd = *D x* *D* wd are very large, but never used as such. All the cal- culations are carried out using the 1D dictionaries. In order to demon- strate the gain in sparsity attained by the approximation of 3D images by

partitioning into 3D blocks, we use dictionaries *D* wd and *D* pd only for the approximation of the single channel 2D images. For the 3D case we

Fig. 1. The left graph illustrates the prototype atoms which generate by translation the dictionaries *D Hm* ; *m* = 1; …; 7. The prototypes in the right graph generate by translation the dictionaries.*D Pm* ; *m* = 1; …; 7.

maintain the redundancy of the 3D dictionary equivalent to that of the 2D

*x*

dictionary, by considering the 1D dictionary *D*~

= *D x* ∪*D x*∪*D P* .

atoms in the atomic decomposition (12). For the channel by channel

pd

C

S

1

where for the 3D representation *K* = P*Q* 1*kq*; with *kq* the number of

Notice that *D P*1 is the standard Euclidean basis for R*Nx* , also called the Dirac's basis, i.e., the basis arising by translation of the first atom in [Fig. 1](#_bookmark2).

*q*=

decomposition of a *Nz*-channel image, each channel is partitioned into

*P* = (*Nx* · *Ny*)/*N*2 blocks I*p z*; *p* = 1;…; *P*, which are approximated by the

Notice that *D*~ *x* ⊂*D x*

and *D*~ *x* ⊂*D x*

. We also consider *D*~ *y D*~ *x*

*b* ;

and

pd pd

pd wd

pd = pd

2D atomic decompositions

~ *z* = *D*~ *x* , but taking *Nx* = *Nz*. The redundancy of the resulting dic- tionary *D*~ = *D*~ *x* *D*~ *y* *D*~ *z* is equivalent to the redundancy of the

*D*

pd

pd

pd

pd

pd

ℓ*y*;*p*;*l*

*z*

*n*

*n*

pd

2D dictionary *D*

pd

. In 3D we use the same dictionary in both domains

*kp*;*l*

I*kp*;*l* = *cp*;*l* (*n*)d*x*

*p*

*p*

ℓ*x*;*p*;*l*

X

*n*=1

d*y*

; *l* = 1; …; *N* ; (14)

*D*~ wd = *D*~ pd.

where the indices ℓ*x*;*p*;*l*; ℓ*y*;*p*;*l* are selected for each channel *l* by the

*n n*

1. Numerical results

OMP2D algorithm. Accordingly, the number *K* in (13) is given as *K* =

P*Nz* P*P*

*l*=1

*p*=1

*kp*;*l*, with *kp*;*l* the number of atoms in the atomic decomposi-

The merit of the simultaneous approximation of multiple channel images is illustrated in this section by recourse to two numerical exam- ples. Firstly we make the comparison between the sparsity produced by the joint approximation of the Red-Green-Blue (RGB) channel images partitioned into blocks of size *Nb* ×*Nb* × 3 and the sparsity obtained by the independent approximation of each channel partitioned into blocks of size *Nb* × *Nb*. Secondly, the full power of the approach is illustrated through the gain in sparsity attained by approximating hyper-spectral

images partitioned into 3D blocks, vs the plane by plane approximation.

tion (14).

Notice that the SR is a measure of the reduction of dimensionality for representing an image. The larger the value of the SR the smaller the dimensionality of the atomic decomposition representing the whole image. The required quality of the approximation is ensured with respect to the Mean Structural SIMilarity (MSSIM) index [[47,48]](#_bookmark32) and the clas- sical Peak Signal-to-Noise Ratio (PSNR), which for a 3D image is defined as

!

In both cases, once the approximation of each 3D block I*q*

in the

PSNR = 10 log10

(Imax)2

MSE

I — I*K*

; MSE = 3D;

image partition is completed, for *q* = 1; …; *Q* the *kq*-term atomic decomposition of the corresponding block is expressed in the form

*kq*

X

I*kq* = *cq*(*n*)d*xx*;*q* d*yy*;*q* d*z z*;*q* . (12)

*q*

ℓ*n*

ℓ*n*

ℓ*n*

*Nx* · *Ny* · *Nz*

where Imax is the maximum intensity range and I*K* the image approxi- mation.

*n*=1

The sparsity of the representation of an image of dimension *N* = *Nx* ·

*Ny* · *Nz* is measured by the Sparsity Ratio (SR), which is defined as:

SR = *N*; (13)

*K*

* 1. *Example I*

In this example we use the Kodak data set consisting of 24 true color images shown in [Fig. 2](#_bookmark3).

The approximations are realized in both domains by maintaining the

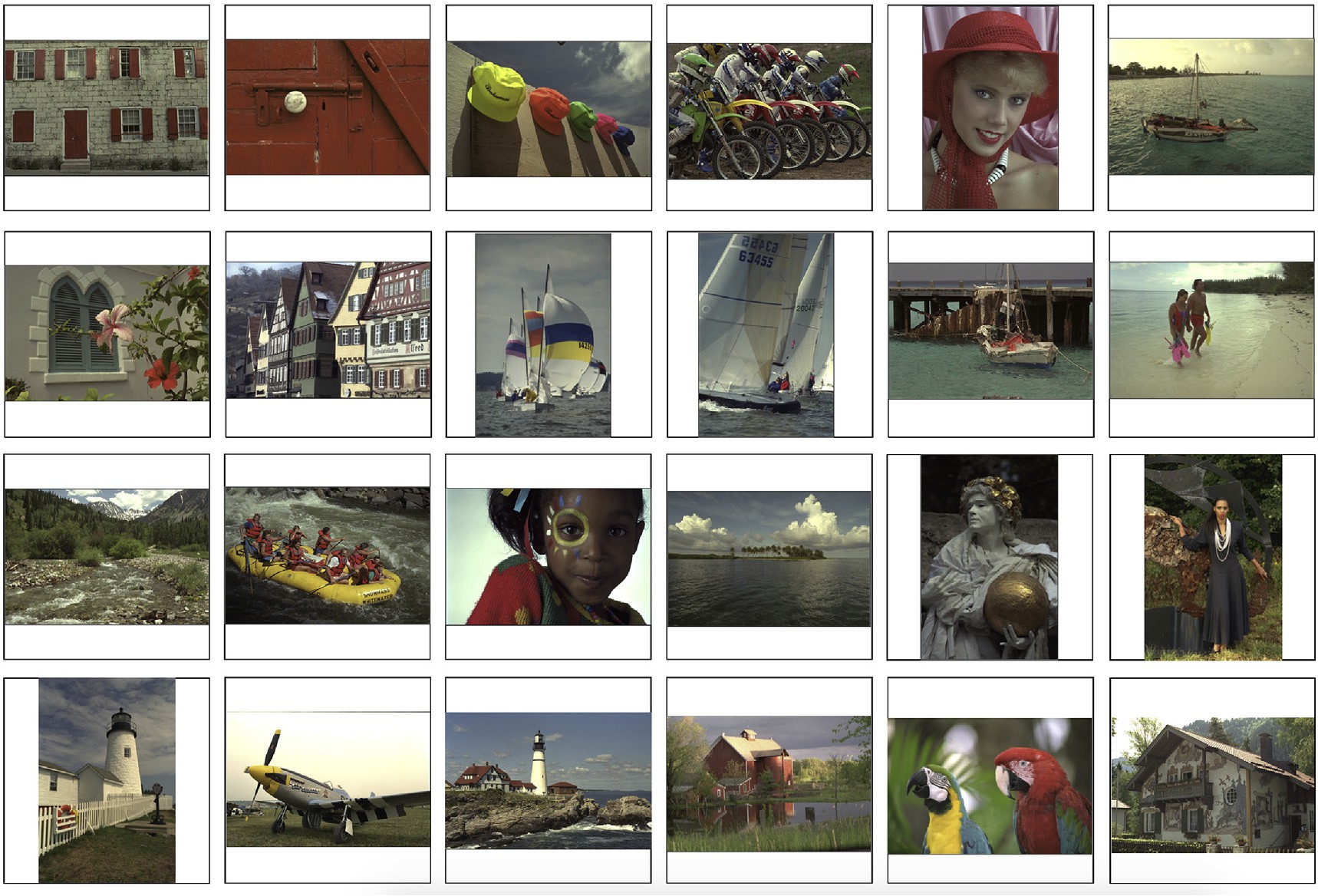


Fig. 2. Illustration of the Kodak data set consisting of 24 true color images, credit Rich Franzen [[49]](#_bookmark33). The size of these images is 768 × 512 × 3, for most of them, except for numbers 4, 9, 10, 17, 18 and 19, which are of size 512 × 768 × 3. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

same redundancy in the 2D and 3D dictionaries. For the independent approximation of the 2D channels the partitions are realized with blocks of size 8 × 8 and 16 × 16 (a partition of block size 24× 24 does not improve results for this data set). Accordingly, the simultaneous approximation of the 3 color channels involves partitions of block size 8 ×8 × 3 and 16 ×16 × 3 respectively.

As already discussed, for the independent approximation of the 2D channels we consider the dictionaries *D* pd (in the pd) and *D* wd (in the wd) as given in Sec. [3.4](#_bookmark1). For the simultaneous approximation of the 3

channels we consider the dictionaries *D*~ pd given in the same section.

Both dictionaries have redundancy of 125.

The average values of SR (SR), with respect to the 24 images in the set, are given in [Table 1](#_bookmark5) for the approaches and partitions indicated by the first column.

All the results in the left half of the table correspond to PSNR = 45 dB and all the results in the right half correspond to PSNR = 41 dB. The third and fifth columns give the standard deviations (std). For completeness we have also produced the SR rendered by nonlinear thresholding of the wavelets coefficients (last row in the table). Notice that the resulting sparsity is poor in comparison with the other 2D results.

All the results were obtained in the MATLAB environment on a notebook 2.9 GHz dual core i7 3520 M CPU and 4 GB of memory. For the channel by channel approximation a C++ MEX file implementing OMP2D was used. For the 3D approximation SPMP3D was implemented by a C++ MEX file.

As observed in [Table 1](#_bookmark5) the largest SR is achieved in the wd and partition 16 ×16 × 3 (c.f. last but one row of [Table 1](#_bookmark5)). However, the results obtained by partition 8 ×8 × 3 are very close (c.f. last row of the upper half of [Table 1](#_bookmark5)) and constitute a better tradeoff between SR and approximation time.

[Fig. 3](#_bookmark4) shows the actual values of SRs for this partition in the wd for each of the 24 images in the data set (c.f. [Fig. 2](#_bookmark3)). The average time for the 3D approximation was 53 s per image.

[Fig. 4](#_bookmark6) demonstrates the gain in visual quality obtained when the approximation of Images 3, 7 and 12 are realized simultaneously in 3D, instead of independently for each 2D channel. In both cases the SR is fixed at a high value SR = 63.5. While the 3D approximation is still of good quality (c.f. images on the left in [Fig. 4](#_bookmark6)) the distortion of the channel by channel approximation is very noticeable even at the scale of the figure (c.f. images on the right in [Fig. 4](#_bookmark6)).





Fig. 3. SR for the 45 dB approximation, in the wd, of each of the 24 images in the Kodak data set (c.f. [Fig. 2](#_bookmark3) enumerated from top left to bottom right). The results for the independent approximation of each 2D color channel are repre- sented by the filled circles and those corresponding to the simultaneous approximation of the 3 channels are represented by the filled squares. The corresponding partitions are of size 8 × 8 and 8 × 8 × 3. (For interpretation of the references to color in this figure legend, the reader is referred to the Web version of this article.)

the three channels in the block, which is roughly the *kq*-value corre- sponding to the equivalent block in the gray scale image. The right graph corresponds to *kq* for each block of size 8 ×8 × 3 in the 3D approxima- tion. Both approximations are realized in the pd. The lower graph is the image given as 3 channels of 512 × 768 pixels each. It follows from the figure that the points corresponding to the 3D approximation give mode details about the image.

* 1. *Example II*

We consider now the approximation of the hyper-spectral images illustrated in [Fig. 6](#_bookmark8). Details on the images acquisition and processing are described in Refs. [[50](#_bookmark34)–[52]](#_bookmark34).

All four images are of size 1016 × 1336 × 32, and have been

approximated in partitions of block size *N* × *N* , with *N* = 8; 16, and 24

As a final remark it is worth noting that the number *kq* of atoms in the

for the 2D approximation, and 8

*b b b*

approximation of each block *q* of an image partition produces a mean- ingful summary of local sparsity.

The upper graphs of [Fig. 5](#_bookmark7) are a representation of the piecewise sparsity corresponding to Image 22 in the Kodak data set. Both graphs are arrays of 64 × 96 points. Each point corresponds to the number *kq* of

×8 × 8 for the 3D approximation. For the 2D channel by channel approximation we use the dictionaries *D* pd and *D* wd as defined in Sec. [3.4](#_bookmark1). For the 3D approximation we maintain

the redundancy as in 2D using the dictionary *D*~ pd introduced Sec. [3.4](#_bookmark1)

and *D*~ *a* = *D*~ *a* .

wd

pd

atoms in the approximation of a block *q*. The left graph corresponds to block size 8 × 8 in the 2D approximation, by taking the average *kq* over

Because the range of intensity varies across the images, in order to compare SRs with different approaches we fix the Signal to Noise Ratio (SNR)

Table 1

Mean value of the SR, with respect to the 24 images in the set, obtained with the 2D and 3D approximations in both the pd and wd for two different sizes of the image partition. The last row in the table gives the results corresponding to standard nonlinear thresholding of wavelet coefficients, to achieve the same quality of the approximation as with the dictionaries: PSNR = 45dB (left half) and PSNR = 41dB (right half).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| PSNR | 45 dB |  |  | 41 dB |  |  |
|  | SR | std |  | SR | std |
| pd 2D 8 × 8 | 6.2 | 2.0 |  | 9.1 | 3.5 |  |
| pd 3D 8 × 8 × 3 | 10.3 | 2.9 |  | 16.1 | 5.5 |  |
| wd 2D 8 × 8 | 7.1 | 2.6 |  | 11.8 | 5.8 |  |
| wd 3D 8 × 8 × 3 | 11.6 | 3.8 |  | 20.9 | 9.2 |  |
| pd 2D 16 × 16 | 7.1 | 2.5 |  | 11.1 | 5.0 |  |
| pd 3D 16 × 16 × 3 | 11.6 | 3.6 |  | 18.8 | 7.5 |  |
| wd 2D 16 × 16 | 7.5 | 2.7 |  | 12.0 | 6.2 |  |
| wd 3D 16 × 16 × 3 | 12.4 | 3.9 |  | 20.4 | 8.9 |  |
| Thresholding in the wd | 3.2 | 1.1 |  | 4.9 | 2.6 |  |

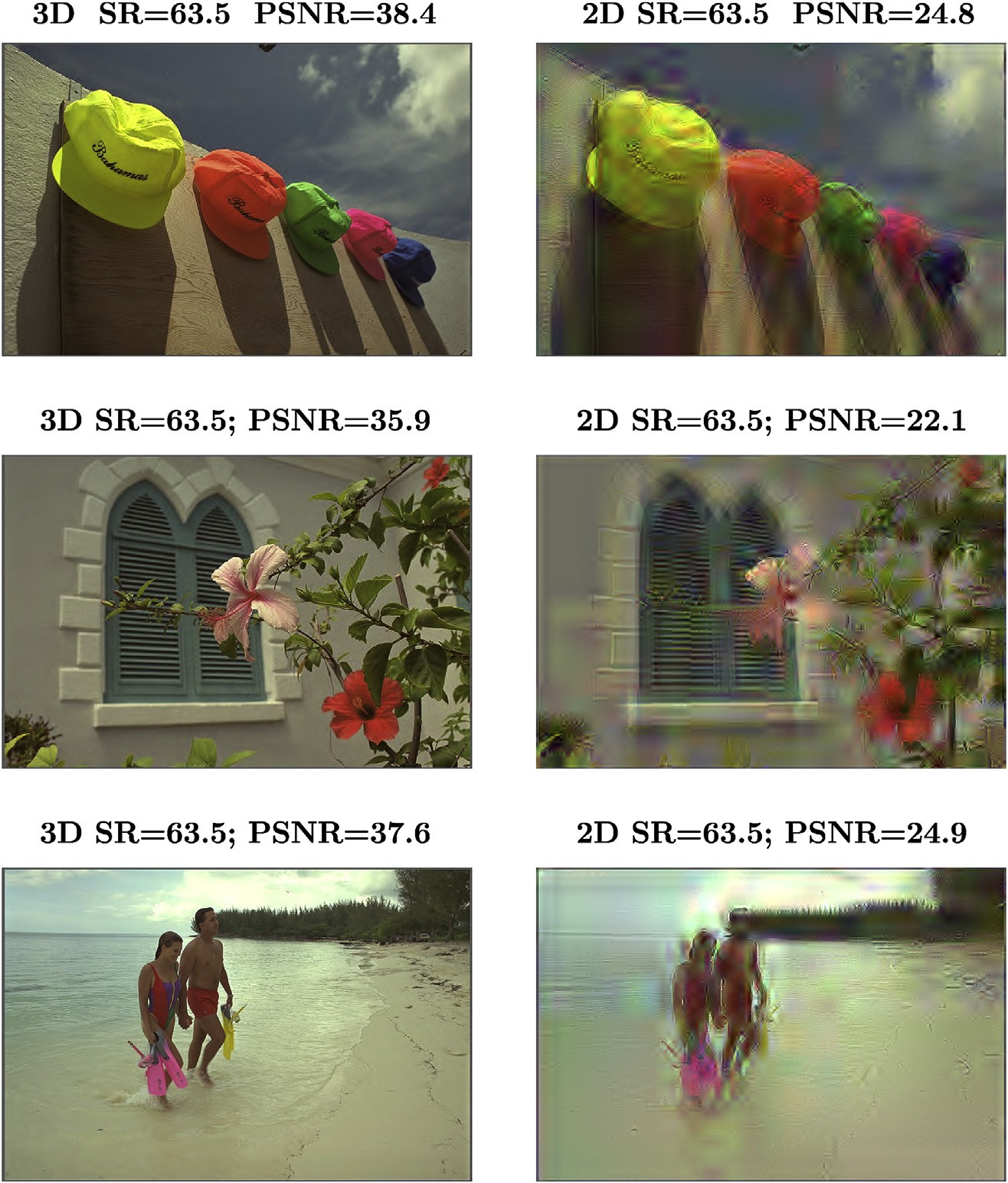


Fig. 4. Approximations of Images 3, 7 and 12 in the Kodak data set, for SR = 63.5. The images on the left are the 3D approximations. The images on the right are the 2D channel by channel approximations.



Fig. 5. The upper graphs are a representation of the piecewise sparsity corresponding to Image 22 in the Kodak data set. Both graphs are arrays of 64× 96 points. Each point corresponds to the number *kq* of atoms in the approximation of a block *q*. The left graph corresponds to the 2D approximation and the right graph to the 3D approximation. The lower graph is the image given as 3 channels of 512 × 768 pixels each.



Fig. 6. Illustration of the hyper-spectral images available on [[53,54]](#_bookmark35). From top left to bottom right in [Table 2](#_bookmark9) are labelled as Ribei., Graff., Rose, and Col. The of size of all four images is 1016 ×1336 × 32 pixels.

SNR = 10 log10

!. (15)

excellent visual quality.

The SRs produced by the 3D approximation are indicated by SR3D

and

Every block in the partition is approximated up to the same error. With all the approaches, two global values of SNR (31 dB and 33 dB) were considered. These values of SNR correspond to the values of PSNR shown in [Tables 2 and 3](#_bookmark9). In all of the cases the approximations are of



I 2 3D

I — I*K* 2

3D

Table 2

Values of SR for the approximation in the pixel-intensity domain of the images listed in the first row. SR2D indicates the SR for the plane by plane approximation in partition of block side *Nb* = 8, 16, and 24. SR3D corresponds to a partition in 3D blocks of size 8 × 8 × 8. The times for completing the approximations are

given immediately below the sparsity results in minutes.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Image | Ribei. | Graff. | Rose | Col. |
| SNR = 31 dB |  |  |  |  |
| PSNR | 46.8 | 48.2 | 47.8 | 46.7 |
| SR2D *Nb* = 8 | 19.2 | 19.2 | 24.1 | 47.7 |
| Time (min) | 1.6 | 1.6 | 1.3 | 0.9 |
| SR2D *Nb* = | 27.3 | 25.5 | 38.7 | 110.6 |

those produced by the 2D plane by plane approximation by SR2D. The times for completing the approximations are given in the row right after the corresponding sparsity result.

Remark 3. In both [Tables 2 and 3](#_bookmark9) the values of SR3D are significantly larger than the values of SR2D, except for the Col. image and 24 × 24 blocks. For this image we were able to increase the 3D block size up to 16 ×16 × 16 and the results for SNR = 31dB are SR3D = 357 in the pd and SR3D = 892 in the wd (35 min and 10 min respectively). For SNR = 33 dB SR3D = 247 in the pd and SR3D = 590 in the wd (55 min and 20 min respectively).

Table 3

Same description as in [Table 2](#_bookmark9), but the approximations are realized by applying

first a wavelet transform to each of the 32 channels.

Image Ribei. Graff. Rose Col. SNR = 31 dB

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 16 |  |  |  |  | PSNR | 46.8 | 48.2 | 47.8 | 46.7 |
| Time (min) | 3.4 | 3.8 | 2.1 | 1.1 | SR2D *Nb* = 8 | 28.6 | 26.8 | 38.6 | 56.5 |
| SR2D *Nb* = | 29.6 | 26.8 | 44.2 | 147.5 | Time (min) | 1.4 | 1.5 | 1.2 | 0.8 |
| 24 |  |  |  |  | SR2D *Nb* = 16 | 36.5 | 34.1 | 63.4 | 144.8 |
| Time (min) | 7.6 | 9.2 | 4.5 | 1.5 | Time (min) | 2.7 | 3.5 | 2.3 | 0.9 |
| SR3D *Nb* = 8 | 49.1 | 59.7 | 74.6 | 137.2 | SR2D *Nb* = 24 | 37.2 | 35.7 | 71.1 | 193 |
| Time (min) | 18 | 15 | 10 | 6 | Time (min) | 9.2 | 12 | 4.8 | 1.8 |
| SNR = 33 dB |  |  |  |  | SR3D *Nb* = 8 | 86.5 | 108.0 | 182.2 | 371.7 |
| PSNR | 48.8 | 50.2 | 49.8 | 48.7 | Time (min) | 13 | 10 | 6 | 3 |
| SR2D *Nb* = 8 | 15.2 | 15.4 | 19.3 | 41.5 | SNR = 33 dB |  |  |  |  |
| Time (min) | 2.3 | 2.1 | 1.7 | 1.1 | PSNR | 48.8 | 50.2 | 49.9 | 48.7 |
| SR2D *Nb* = | 20.4 | 19.5 | 29.1 | 86.4 | SR2D *Nb* = 8 | 22.6 | 21.8 | 33.0 | 56.1 |
| 16 |  |  |  |  | Time (min) | 1.7 | 1.8 | 1.5 | 1.0 |
| Time (min) | 5.4 | 5.6 | 2.9 | 1.2 | SR2D *Nb* = 16 | 26.6 | 25.8 | 48.2 | 118.3 |
| SR2D *Nb* = | 21.9 | 20.5 | 32.7 | 106.3 | Time (min) | 3.5 | 5.0 | 2.3 | 1.1 |
| 24 |  |  |  |  | SR2D *Nb* = 24 | 21.9 | 26.8 | 52.0 | 144.0 |
| Time (min) | 12 | 14 | 6.8 | 1.9 | Time (min) | 12 | 15 | 8.5 | 1.9 |
| SR3D *Nb* = 8 | 33.5 | 41.6 | 53.2 | 106.5 | SR3D *Nb* = 8 | 55.1 | 70.5 | 129.5 | 313.3 |
| Time (min) | 25 | 21 | 16 | 8 | Time (min) | 23 | 18 | 10 | 1.8 |



Fig. 7. Illustration of two remote sensing hyper-spectral images taken from Ref. [[55]](#_bookmark36). The graph on the left is the Urban image (size 320 ×320 × 128 pixels). The graph on the right is a portion of the University of Pavia image (256 ×256 × 96 pixels).





Fig. 8. SR vs SNR values for the 3D approximation in both the pd and wd for the Urban and University of Pavia remote sensing images.

On comparing the two tables a drastic improvement in the values of SR3D is observed when the approximation is realized in the wavelet domain. This feature is a consequence of the fact that the planes of the natural images are very sparse in the wavelet domain. In order to high- light differences we produce next the SR3D corresponding to the two remote sensing images in [Fig. 7](#_bookmark10). The graph on the left represents the Urban remote sensing hyper-spectral image taken from Ref. [[55]](#_bookmark36). The graph on the right is a portion of the University of Pavia image also taken

from Ref. [[55]](#_bookmark36).

[Fig. 8](#_bookmark11) plots the SR vs four values of SNR, corresponding to the 3D approximations of the Urban and University of Pavia images in both the pd and wd.

Notice that the results in the pd are much closer to the results in the wd than they are in the case of the natural images in [Fig. 6](#_bookmark8). This is because, as illustrated in [Fig. 9](#_bookmark12), the planes of the remote sensing images are not as sparse in the wd as the planes of the natural images are.

1. Conclusions

High quality approximation of 3D images has been considered within the context of data reduction. A remarkable improvement in sparsity achieved by the simultaneous approximation of multiple channels has been illustrated through numerical experiments of different natures. Firstly it was demonstrated that a standard data set of RGB images can be approximated at high quality using far fewer elementary components if each image is treated as a very thin 3D array instead of as 3 independent 2D arrays. Secondly the full power of the approach was demonstrated through the approximation of hyper-spectral images. For the hyper- spectral natural images the sparsity is remarkably higher if the approx- imation is realized in the wavelet domain. For the remote sensing images the domain of approximation has less influence because, as opposed to natural images, these images are not as sparse in the wavelet domain as natural images are.

Taking into account the major reduction of dimensionality demon- strated by the numerical examples in this work, we feel confident that the proposed approach will be of assistance to the broad range of image processing applications which rely on a transformation for data reduction as a first step of further processing.

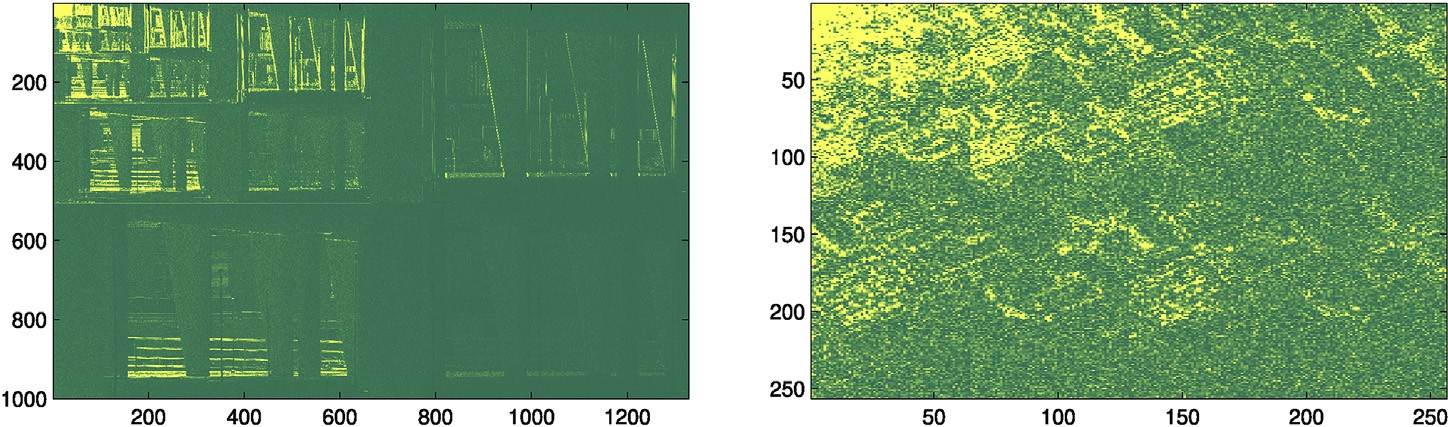


Fig. 9. Absolute value of the wavelet transform of a plane in the Col. image (left graph) and in the University of Pavia image (right graph).

Declaration of Competing Interest

The authors declare no conflict of interest.

Acknowledgments

We are also indebted to P. Getreuer, for making available the wave- letcdf97 MATLAB function that we have used for the transformation of each single channel image to the wavelet domain.

References

1. [Wright J, Ma Yi, Mairal J, Sapiro G, Huang TS, Yan S. Sparse representation for](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref1) [computer vision and pattern recognition. Proc of the IEEE 2010;98:1031](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref1)–[44](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref1).
2. [Elad M. Sparse and redundant representations: from theory to applications in signal](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref2) [and image processing. Springer; 2010](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref2).
3. [Zhang Z, Xu Y, Yang J, Li X, Zhang D. A survey of sparse representation: algorithms](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref3) [and applications. IEEE access; 2015](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref3).
4. [Mairal J, Eldar M, Sapiro G. Sparse representation for color image restoration. IEEE](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref4) [Trans Image Process 2008;17:53](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref4)–[69](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref4).
5. [Dong W, Zhang L, Shi G, Li X. Nonlocally centralized sparse representation for](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref5) [image restoration. IEEE Trans Image Process 2013;22:1620](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref5)–[30](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref5).
6. Gou J, Hou B, Ou W, Mao Q, Yang H, Liu Y. Several robust extensions of collaborative representation for image classification. Neurocomputing 2018. <https://doi.org/10.1016/j.neucom.2018.06.089>.
7. [Gou J, Yi Z, Zhang D, Zhan Y, Shen X, Du L. Sparsity and geometry preserving graph](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref7) [embedding for dimensionality reduction. IEEE Access 2018;6:75748](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref7)–[66. 10.1109/](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref7) [ACCESS.2018.2884027](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref7).
8. Gou J, Wang L, Yi Z, Lv J, Mao Q, Yuan Y-H. A new discriminative collaborative neighbor representation method for robust face recognition. IEEE Access 2018;6: 74713–27. <https://doi.org/10.1109/ACCESS.2018.2883527>. 2018.
9. [Wright J, Yang AY, Ganesh A. Robust face recognition via sparse representation.](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref9) [IEEE Trans Pattern Anal Mach Intell 2009;31:210](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref9)–[27](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref9).
10. [Yuan XT, Liu X, Yan S. Visual classi](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref10)fi[cation with multitask joint sparse](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref10) [representation. IEEE Trans Image Process 2012;21:4349](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref10)–[60](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref10).
11. [Yang J, Wright J, Huang T. Image super-resolution via sparse representation. IEEE](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref11) [Trans Image Process 2010;19:2861](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref11)–[73](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref11).
12. [Zhang Y, Liu J, Yang W, Guo Z. Image super-resolution based on structure-](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref12) [modulated sparse representation. IEEE Trans Image Process 2015;9:2797](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref12)–[810](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref12).
13. [Dabov K, Foi A, Katkovnik V. Image denoising by sparse 3-D transform-domain](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref13) [collaborative](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref13) fi[ltering. IEEE Trans Image Process 2007;16:2080](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref13)–[95](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref13).
14. [Caiafa CF, Cichocki A. Computing sparse representations of multidimensional](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref14) [signals Using Kronecker Bases. Neural Comput 2013;25:186](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref14)–[220](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref14).
15. [Cichocki A, Mandic D, De Lathauweri L, Zhou G, Zhao Q, Caiafai C, Phan HA.](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref15) [Tensor decompositions for signal processing applications: from two-way to](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref15) [multiway component analysis. IEEE Signal Process Mag 2015;32:145](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref15)–[63](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref15).
16. [Dai Q, Yoo S, Kappeler A. Sparse representation-based multiple frame video super-](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref16) [resolution”. IEEE Trans Image Process 2017;26:2080](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref16)–[95](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref16).
17. [Mousavi HS, Monga V. Sparsity-based color image super resolution via exploiting](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref17) [cross channel constraints. IEEE Trans Image Process 2017;26:5094](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref17)–[106](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref17).
18. [Rebollo-Neira L, Bowley J. Sparse representation of astronomical images. J Opt Soc](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref18) [Am 2013;30:758](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref18)–[68](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref18).
19. [Donoho DL, Tsaig Y, Drori I, Starck J. Stagewise orthogonal matching pursuit. IEEE](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref19) [Trans Inf Theory 2006;58:1094](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref19)–[121](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref19).
20. [Needell D, Tropp JA. CoSaMP: Iterative signal recovery from incomplete and](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref20) [inaccurate samples”. Appl Comput Harmon Anal 2009;26:301](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref20)–[21](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref20).
21. [Eldar Y, Kuppinger P, Bi](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref21)o€[lcskei H. Block-sparse signals: uncertainty relations and](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref21)

[ef](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref21)fi[cient recovery. IEEE Trans Signal Process 2010;58:3042](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref21)–[54](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref21).

1. [Rebollo-Neira L, Matiol R, Bibi S. Hierarchized block wise image approximation by](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref22) [greedy pursuit strategies. IEEE Signal Process Lett 2013;20:1175](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref22)–[8](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref22).
2. L. Rebollo-Neira, <http://www.nonlinear-approx.info/examples/node09.html> (Last access April 2019)..
3. [Tao JianWen, Wen Shiting, Hu Wenjun. Robust domain adaptation image](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref24) [classi](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref24)fi[cation via sparse and low rank representation. J Vis Commun Image](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref24) [Represent 2015;33:134](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref24)–[48](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref24).
4. [Nie Wei-Zhi, Liu An-An, Su Yu-Ting. 3D object retrieval based on sparse coding in](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref25) [weak supervision. J Vis Commun Image Represent 2016;37:40](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref25)–[5](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref25).
5. [Gao Z, Li SH, Zhu YJ, Wang C, Zhang H. Collaborative sparse representation leaning](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref26) [model for RGBD action recognition. J Vis Commun Image Represent 2017;48:](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref26) [442](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref26)–[52](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref26).
6. [Zhang Youqiang, Guo Cao, Li Xuesong, Wang Bisheng. Cascaded random forest for](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref27) [hyperspectral image classi](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref27)fi[cation. IEEE JSelect Top Appl Earth Observ Remote Sens](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref27) [2018;11:1082](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref27)–[94](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref27).
7. [Li Guiji, Peng Manman, Ke Nai, Li Zhiyong, Li Keqin. Visual tracking via context-](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref28) [aware local sparse appearance model. J Vis Commun Image Represent 2018;56:](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref28) [92](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref28)–[105](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref28).
8. [Mallat S, Zhang Z. Matching pursuit with time-frequency dictionaries. IEEE Trans](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref29) [Signal Process 1993;41(12):3397](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref29)–[415](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref29).
9. [Pati YC, Rezaiifar R, Krishnaprasad PS. Orthogonal matching pursuit: recursive](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref30) [function approximation with applications to wavelet decomposition,” Proc. of the](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref30) [27th ACSSC, vol. 1; 1993. p. 40](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref30)–[4](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref30).
10. [Rebollo-Neira L, Lowe D. Optimized orthogonal matching pursuit approach. IEEE](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref31) [Signal Process Lett 2002;9:137](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref31)–[40](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref31).
11. [Rebollo-Neira L, Bowley J, Constantinides A, Plastino A. Self contained encrypted](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref32) [image folding. Physica A 2012;391:5858](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref32)–[70](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref32).
12. Rebollo-Neira L. Effective sparse representation of X-Ray medical images. Int J Num Method Biomed Eng 2017. <https://doi.org/10.1002/cnm.2886>.
13. Rebollo-Neira L, Rozloˇzník M, Sasmal P. Analysis of a low memory implementation

of the Orthogonal Matching Pursuit greedy strategy. [http://arxiv.org/abs/1](http://arxiv.org/abs/1609.00053) [609.00053](http://arxiv.org/abs/1609.00053); 2017.

1. [Kreutz-Delgado K, Murray JF, Rao BD, Engan K, Lee Te-Won, Sejnowski TJ.](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref35) [Dictionary learning algorithms for sparse representation. Neurocomputing 2003;15:](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref35) [349](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref35)–[96](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref35).
2. [Aharon M, Elad M, Bruckstein A. K-SVD: an algorithm for designing overcomplete](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref36) [dictionaries for sparse representation. IEEE Trans Signal Process 2006;54:4311](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref36)–[22](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref36).
3. [Rubinstein R, Zibulevsky M, Elad M. Double sparsity: learning sparse dictionaries](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref37) [for sparse signal approximation. IEEE Trans Signal Process 2010;58:1553](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref37)–[64](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref37).
4. [Tosi´](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref38)c [I, Frossard P. Dictionary Learning: what is the right representation for my](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref38)

[signal? IEEE Signal Process Mag 2011;28:27](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref38)–[38](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref38).

1. [Zepeda J, Guillemot C, Kijak E. Image compression using sparse representations and](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref39) [the iteration-tuned and aligned dictionary. IEEE J Select Top Signal Process 2011;5:](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref39) [1061](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref39)–[73](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref39).
2. [Hawe S, Seibert M, Kleinsteuber M. Separable dictionary learning. In: Proceedings](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref40) [of the 2013 IEEE conference on computer vision and pattern recognition; 2013.](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref40)

[p. 438](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref40)–[45](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref40).

1. [Srinivas M, Naidu RR, Sastry CS, Krishna Mohana C. Content based medical image](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref41) [retrieval using dictionary learning. Neurocomputing 2015;168:880](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref41)–[95](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref41).
2. [Wen B, Ravishankar S, Bresler Y. Structured overcomplete sparsifying transform](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref42) [learning with convergence guarantees and applications. Int J Comput Vis 2015;114:](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref42) [137](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref42)–[67](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref42).
3. Rebollo-Neira L. A competitive scheme for storing sparse representation of X-Ray medical images. PLoS One 2018. <https://doi.org/10.1371/journal.pone.0201455>.
4. [Cohen A, Daubechies I, Feauveau JC. Biorthogonal bases of compactly supported](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref44) [wavelets. Commun Pure Appl Math 1992;45:485](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref44)–[560](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref44).
5. [Andrle M, Rebollo-Neira L. Cardinal B-spline dictionaries on a compact interval.](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref45) [Appl Comput Harmon Anal 2005;18:336](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref45)–[46](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref45).
6. [Rebollo-Neira L, Xu Z. Adaptive non-uniform B-spline dictionaries on a compact](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref46) [interval. Signal Process 2010;90:2308](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref46)–[13](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref46).
7. [Wang Z, Bovik AC, Sheikh HR, Simoncelli EP. Image quality assessment: from error](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref47) [visibility to structural similarity. IEEE Trans Image Process 2004;13:600](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref47)–[12](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref47).
8. I. Kowalik-Urbaniak, D. Brunet, J. Wang, D. Koff, N. Smolarski-Koff, E. Vrscay, B. Wallace, and Z. Wang, “The quest for ‘diagnostically lossless’ medical image compression: a comparative study of objective quality metrics for compressed medical images”, Proc. SPIE 9037, Medical Imaging 2014: Image Perception, Observer Performance, and Technology Assessment, 903717 (March 11, 2014); doi: 10.1117/12.2043196.
9. R. Franzen, <http://r0k.us/graphics/kodak/>(Last access April 2019)..
10. [Foster DH, Nascimento SMC, Amano K. Information limits on neural identi](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref50)fi[cation](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref50) [of colored surfaces in natural scenes. Vis Neurosci 2004;21:331](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref50)–[6](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref50).
11. [Foster DH, Amano K, Nascimento SMC, Foster MJ. Frequency of metamerism in](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref51) [natural scenes. J Opt Soc Am 2006;23:2359](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref51)–[72](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref51).
12. [Nascimento SMC, Amano K, Forster D. Spatial distributions of local illumination](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref52) [color in natural scenes. Vis Res 2016;120:39](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref52)–[44](http://refhub.elsevier.com/S2590-0056(19)30001-3/sref52).
13. D. H. Foster, [https://personalpages.manchester.ac.uk/staff/d.h.foster/Hyperspectra](https://personalpages.manchester.ac.uk/staff/d.h.foster/Hyperspectral_images_of_natural_scenes_04.html) [l\_images\_of\_natural\_scenes\_04.html](https://personalpages.manchester.ac.uk/staff/d.h.foster/Hyperspectral_images_of_natural_scenes_04.html) (Last access April 2019)..
14. D. H. Foster, [http://personalpages.manchester.ac.uk/staff/d.h.foster/Local\_Ill](http://personalpages.manchester.ac.uk/staff/d.h.foster/Local_Illumination_HSIs/Local_Illumination_HSIs_2015.html) [umination\_HSIs/Local\_Illumination\_HSIs\_2015.html](http://personalpages.manchester.ac.uk/staff/d.h.foster/Local_Illumination_HSIs/Local_Illumination_HSIs_2015.html) (Last access April 2019)..
15. <http://lesun.weebly.com/hyperspectral-data-set.html> (Last access April 2019)..