Abstract

The property of sparse representations concerning capability for information storage is discussed. It is shown that this feature can be used, for instance, for an application that we term Image Folding. The proposed procedure is applicable by means of any suitable transformation. However, it is also the aim of this paper to draw attention in regard to the gain in the sparsity of an image representation achieved by combination of Discrete Cosine and Dirac dictionaries.

1 Introduction

The problem of reducing the dimensionality of a piece of data without losing their information content is of paramount importance in signal processing. Well established transforms, from classical Fourier and Cosine Transforms to Wavelets, Wavelet Packets, and Lapped Transforms, just to mention the most popular ones, are usually applied for generating the transformed domain where the processing tasks are realized. Signals amenable to transformation into a data set of smaller cardinality are said to be compressible. Natural images, for instance, provide a typical example of compressible data.

In the last fifteen years emerging techniques for signal representation are addressing the matter by highly nonlinear methodologies which decompose the signal as a superposition of vectors, normally called ‘atoms’, selected from a large redundant set called a ‘dictionary’ [1]. The representation qualifies to be sparse if the number of atoms for a satisfactory signal approximation is considerably smaller than the dimension of the original data. Available methodologies for highly nonlinear approximations are known as Pursuit Strategies. This comprises Bases Pursuit [2] and Matching Pursuit like greedy algorithms, including Orthogonal Matching Pursuit (OMP) and variations of this method [3–6]. The other ingredient of highly non linear approximations is, of course, the dictionary from where atoms are selected. In this respect, Gabor dictionaries have been shown useful for image and video processing [7,8]. Combined dictionaries, arising by merging for instance orthogonal bases, have received consideration in relation to the theoretical analysis of Pursuit Strategies [9–13]. From a different perspective, other approaches are based on dictionaries learned from large data sets [14,15].

In this Communication we would like to discuss a property intrinsic to sparse representations; since that type of representation entails a projection onto a subspace of lower dimensionality, it generates a null space. This feature suggests the possibility of using the created
space for embedding information. In particular, we discuss an application involving the null space yielded by the sparse representation of an image, to store part of the image itself. We term this application Image Folding. The proposed procedure can be carried out through any transformation giving rise to a sparse representation. However, here we focus on a particular one. We use mixed dictionaries composed of a Discrete Cosine (DC) dictionary and a discrete Dirac basis. Both dictionaries are considered in [2], nevertheless, to the best of our knowledge the significant gain achieved by a combination of these dictionaries, in regard to sparse image representation, has not been reported. Thus, we would also like to draw attention about the suitability of mixed DC and Dirac dictionaries for image representation. As far as sparsity is concerned, at the visual acceptable level of 40 PSNR, they may render a significant improvement in relation to established transforms such as Discrete Cosine Transform (DCT) and Discrete Wavelets Transform (DWT). An additional advantage of these dictionaries is that the OMP methodology for selecting the atoms can be implemented at a reduced complexity cost by means of the DCT.

The paper is organized as follows: In Sec. 2 we discuss the mixed DC-Dirac dictionaries and the implementation of the OMP method based on DCT calculations. In Sec. 3 we discuss ‘something else’. By this we mean to alert that a sparse representation can be used for embedding information. Based on such a possibility, a scheme for image folding and a simple encryption procedure fully implemented by data processing are discussed. The conclusions are presented in Sec. 4.

2 OMP with mixed 2D DC-Dirac Dictionaries

Let us consider the set $D_a$ defined as

$$D_a = \{v_i ; v_i = p_i \cos(\frac{\pi(2j-1)(i-1)}{2M}), j = 1, \ldots, N\}_{i=1}^{M},$$

with $p_i$, $i = 1, \ldots, M$ normalization factors. For $M = N$ this set is a DC orthonormal basis for the Euclidean space $\mathbb{R}^N$. For $M = 2lN$, with $l$ a positive integer, the set is a DC dictionary with redundancy $2l$.

We further consider the set $D_b$, which is a discrete Dirac Basis, i.e.,

$$D_b = \{e_i \in \mathbb{R}^N \text{ with the } i\text{-th component equal to 1 and the other components equal to 0}\}_{i=1}^{N}.$$

A redundant dictionary $D$ for $\mathbb{R}^N \otimes \mathbb{R}^N$ is then obtained as the tensor product $D = D_a \otimes D_b$. We denote by $d_n \in \mathbb{R}^N \otimes \mathbb{R}^N$, $n = 1, \ldots, J$, where $J = 2MN + M^2 + N^2$, the elements of dictionary $D$ and use them to construct the atomic decomposition of an image $I \in \mathbb{R}^N \otimes \mathbb{R}^N$ as

$$I = \sum_i^K c_i^K \cdot d_{i},$$  

where the atoms $d_i$, $i = 1, \ldots, K$ are chosen from the dictionary $D$ by the OMP strategy.

2.1 Dedicated implementation of the OMP approach

The OMP strategy for decomposing $I \in \mathbb{R}^N \times \mathbb{R}^N$ using a dictionary $\{d_{i}\}_{i=1}^{J}$ evolves as follows: Setting $R^1 = I$ at iteration $k + 1$ the OMP algorithm selects the atom, $d_{k+1}$ say, as the one
minimizing the absolute value of the inner products \( \langle d_i, R^k \rangle \), \( i = 1, \ldots, J \), i.e.,

\[
d_{k+1} = \arg \max_{i=1,\ldots,J} |\langle d_i, R^k \rangle| \quad \text{where} \quad R^k = I - \sum_{i=1}^k c^k_i d_i.
\] (2)

The coefficients \( c^k_i \), \( i = 1, \ldots, J \) in (2) are such that \( \| R^k \|^2 \) is minimum, which is equivalent to requesting \( R^k = \hat{P}_{S_k} I \), where \( \hat{P}_{S_k} \) is the orthogonal projection operator onto the span of the \( k \) selected atoms i.e., \( S_k = \text{span}\{d_i\}_{i=1}^k \). We base our implementation for determining the coefficients \( c^k_i \), \( i = 1, \ldots, k \) on recursive biorthogonalization [5,6]. At each iteration we update the vectors

\[
w^{k+1}_i = w^k_i - w^{k+1}_i \langle d_{k+1}, w^k_i \rangle, \quad i = 1, \ldots, k,
\] (3)

where \( w^{k+1}_i = q_{k+1}/\|q_{k+1}\|^2 \), with \( q_{k+1} = d_{k+1} - \hat{P}_{S_k} d_{k+1} \) and \( q_1 = d_{t_1} \). The projector \( \hat{P}_{S_k} \) is here computed as \( \hat{P}_{S_k} = Q_k Q_k^* \) where the \( k \)-columns of matrix \( Q_k \) are the vectors \( q_i/\|q_i\|, i = 1, \ldots, k \) and \( Q^*_k \) indicates the transpose conjugate of \( Q_k \). However, to calculate the coefficients of the linear superposition we express the projectors as \( \hat{P}_{S_k} = A_k B_k^* \) where the \( k \)-columns of matrix \( A_k \) are the selected vectors and the \( k \)-columns of matrix \( B_k \) are the vectors \( w^k_i, i = 1, \ldots, k \). Thus, the required coefficients arise from the inner products \( c^k_i = \langle w^k_i, I \rangle, \quad i = 1, \ldots, k \). The code for this type of implementation available is at [16].

The complexity of the OMP approach is dominated by the calculation of the quantities \( \langle d_i, R^k \rangle \), \( i = 1, \ldots, J \) in (2) at each iteration step. For the present dictionaries these quantities can be evaluated by fast DCT. In order to discuss the matter let us re-name the dictionary atoms as follows

- for \( n = 1, \ldots, M^2 \) \( d_n \rightarrow v_i \otimes v_j, i = 1, \ldots, M, j = 1, \ldots, M \)
- for \( n = M^2 + 1, \ldots, M^2 + MN \) \( d_n \rightarrow v_i \otimes e_j, i = 1, \ldots, M, j = 1, \ldots, N \)
- for \( n = M^2 + MN + 1, \ldots, M^2 + 2MN \) \( d_n \rightarrow e_i \otimes v_j, i = 1, \ldots, N, j = 1, \ldots, M \)
- for \( n = M^2 + 2MN, \ldots, J \) \( d_n \rightarrow e_i \otimes e_j, i = 1, \ldots, N, j = 1, \ldots, N \).

Hence, the inner products \( \langle d_i, R^k \rangle \), \( i = 1, \ldots, J \) are calculated as

\[
\langle v_i \otimes v_j, R^k \rangle = p_i p_j \sum_{s,r=1}^N R^k(s,r) \cos(\frac{\pi(2s-1)(i-1)}{2M}) \cos(\frac{\pi(2r-1)(j-1)}{2M})
\]

(4)

\[
\langle v_i \otimes e_j, R^k \rangle = p_i \sum_{s=1}^N R^k(s,j) \cos(\frac{\pi(2s-1)(i-1)}{2M})
\]

(5)

\[
\langle e_i \otimes v_j, R^k \rangle = p_j \sum_{r=1}^N R^k(i,r) \cos(\frac{\pi(2r-1)(j-1)}{2M})
\]

(6)

\[
\langle e_i \otimes e_j, R^k \rangle = R^k(i,j)
\]

(7)

If \( M = N \) (4) is the 2D DCT of the residual \( R^k \) whilst (5) and (6) are the 1D DCT of the rows and columns of \( R^k \), respectively. If \( M = 2lN \), for some positive integer \( l \), the calculations can also be carried out through fast DCT by zero padding. Then with the dictionaries we are considering the complexity required for evaluation of inner products in (2) is \( O(M^2 \log_2 M) + O(M \log_2 M) \).

The images we have represented with the above outline procedure are the 6 test images of Fig. I. For the actual processing we divide each image into blocks of 16 \( \otimes \) 16 pixels. The
Figure 1: The six test images from left to right, top to bottom: Boat, Bridge, Film Clip, Lena, Mandril, Peppers.

| Image   | DCT$2 \cup$ Dirac | DCT  | WT  |
|---------|-------------------|------|-----|
| Boat    | 6.03              | 3.63 | 3.65|
| Bridge  | 3.7               | 2.06 | 2.2 |
| Film    | 8.06              | 4.53 | 4.8 |
| Lena    | 10.06             | 6.5  | 6.97|
| Mandril | 3.32              | 1.91 | 1.90|
| Peppers | 7.74              | 4.36 | 3.39|

Table 1: Compression ratio achieved by the mixed DC-Dirac dictionary and those yielded by DCT and DWT.
The sparsity measure we use is the compression ratio defined as

\[
CR = \frac{\text{number or total pixels}}{\text{divided the number of total coefficients}}
\]

In all the cases the number of coefficients are determined so as to produce a PSNR of 40 dB by the image reconstruction and the dictionary is a mixed DC redundancy 2 and Dirac basis. The results are given in the first column of Table I. For comparison the second column of this table shows results produced by fast DCT implemented using the same blocking scheme, and for further comparison the results produced by the Cohen-Daubechies-Feauveau 9/7 DWT (applied on the whole image at once) are displayed in the last column of Table I. Notice that, while for the fixed PSNR of 40 dB the DCT and DWT yield comparable \( CR \) the corresponding \( CR \) obtained by the mixed dictionaries, for all the images, is significantly higher.

3 Something else

By the title of this section we try to emphasize the fact that, since a sparse representation involves a projection onto a lower dimension subspace, it also creates room for storing ‘something else’. Certainly, if with \( K \)-dictionary’s atoms \( \{d_i\}_{i=1}^{K} \) rendering a sparse representation of an image we construct matrix \( A_K \) having those atoms as columns and matrix \( B_K \) the columns of which are the dual atoms \( \{w^K_i\}_{i=1}^{K} \) defined in the previous section, as already mentioned, the approximated image is obtained as \( I^K = A_K B_K^* I \). Thus the coefficients in the atomic decomposition of \( I^K \) are calculated as the inner products

\[
c^K_i = \langle w^K_i, I \rangle, \ i = 1, \ldots, K.
\]  

(8)

The calculation of the duals atoms as in (3) guarantees that they span the same space \( S_K \) as the selected atoms \( \{d_i\}_{i=1}^{K} \) [5]. The sparsity property of a representation implies that \( S_K \) is a subspace of the image space \( \mathbb{R}^N \otimes \mathbb{R}^N \). Hence, by denoting as \( S_K^\perp \) the orthogonal complement of \( S_K \) in \( \mathbb{R}^N \otimes \mathbb{R}^N \) we have \( \mathbb{R}^N \otimes \mathbb{R}^N = S_K \oplus S_K^\perp \) where \( \oplus \) indicates orthogonal sum. Now, if we take a vector \( F \in S_K^\perp \) and add it to the image forming the vector \( G = I + F \) to replace \( I \) in (8), since \( F \) is orthogonal to the duals \( \{w^K_i\}_{i=1}^{K} \), we still have

\[
\langle w^K_{\ell_i}, G \rangle = \langle w^K_{\ell_i}, I + F \rangle = \langle w^K_{\ell_i}, I \rangle = c^K_{\ell_i}, \ i = 1, \ldots, K.
\]  

(9)

This suggests the possibility of using the sparse representation of an image to embed the image with additional information stored in the vectors \( F \in S_K^\perp \). In order to do this, we apply the earlier proposed scheme to embed redundant representations [17], which in this case operates as described below.

**Embedding Scheme:** Consider that \( I^K = A_K B_K^* I \) is the reconstruction of a sparse representation of an image \( I \). We embed \( L = N^2 - K \) numbers \( h_i, \ i = 1, \ldots, L \) into a vectors \( F \in S_K^\perp \) as prescribed below.

- Take an orthonormal basis \( u_i, \ i = 1, \ldots, L \) for \( S_K^\perp \) and form vector \( F \) as the linear combination
  \[
  F = \sum_{i=1}^{L} h_i u_i.
  \]

- Add \( F \) to \( I^K \) to obtain \( G = I^K + F \)
Information Retrieval:

Given $G$ retrieve the numbers $h_i, i = 1, \ldots, L$ as follows.

- Use $G$ to compute the coefficients of the sparse representation of $I$ as in (9). Use this coefficients to reconstruct the image $\tilde{I}^K = \sum_{i=1}^{K} c_i^K v_i$

- Obtain $F$ from the given $G$ and the reconstructed $\tilde{I}^K$ as $F = G - \tilde{I}^K$. Use $F$ and the orthonormal basis $u_i, i = 1, \ldots, L$ to retrieve the embedded numbers $h_i, i = 1, \ldots, L$ as $h_i = \langle u_i, F \rangle, i = 1, \ldots, L$

One can encrypt the embedding procedure simply by randomly controlling the order of the orthogonal basis $u_i, i = 1, \ldots, L$ or by applying some random rotation to the basis, requiring a key for generating it. An example is given in the next section.

3.1 Application to image folding

We apply now the above discussed embedding scheme to fold an image. For this we simple process the image by dividing it into, say $q$, blocks $I_q$ of $N_q \otimes N_q$ pixels. We find the representation of each block with the DC Dirac dictionary (in the examples presented here the blocks are of $16 \otimes 16$ and we use a DC redundancy 2 and Dirac basis dictionary). Then we embed the coefficients of the sparse representation of some of the blocks into vectors $F_q$, that we add to the pixels of the other blocks to obtain $G_q = I_q + F_q$. The number of blocks we embed depends of the sparsity of the representation. The higher the sparsity is the less blocks are needed to embed the rest of them, thereby resulting in a more effective folding.

In this example we have built the orthogonal basis $u_i, i = 1, \ldots, L$ by randomly generating vectors $y_i \in R^{N} \otimes R^{N}, i = 1, \ldots, L$ using a ‘public’ initialization for the random generator. We then compute vectors $\tilde{y}_i = y_i - \hat{P}_{S_K}y_i$ that we further orthonormalize to have the vectors $o_i, i = 1, \ldots, L$. Now, setting an initialization key, which remains unknown for an unauthorized user, we apply a random permutation $\Pi_{\text{key}}$ on vectors $o_i$ to finally generate the orthonormal basis $u_i = \Pi_{\text{key}}o_i, i = 1, \ldots, L$, for embedding the coefficients of the remaining blocks. We illustrate the results on the image of Lena, which by a DC-Dirac dictionary has a compression ratio of approximately 10, so that we have folded most of the image into the small piece of image shown at the top of Fig. 2. The middled picture is the unfolded image obtained without using the right security key. Finally the last picture is the unfolded image using the permutation matrix initialized by the correct key. In Fig. 3 we present the same process but using the key to generate a random rotation of the basis (instead of a permutation).

Remark 1. Notice that within this framework in order to store the folded image we need to quantize the blocks of pixels $G_q$ to convert them into integer numbers, which implies some lost of information due to the rounding operation. However, the quantization step does not prevent us from recovering the coefficients corresponding to the folded pixels with enough accuracy to produce a good representation of those blocks of image. The PSNR of the recovered image $\tilde{I}^K$ (after folding it) is 40.56 while the PSNR of the original image $I^K$ is 41.49.
Figure 2: The small picture at the top is the folded Image. The middle picture is the unfolded image without knowledge of the private key to initialize the permutation. The bottom picture is the unfolded picture when the correct key is used.
Figure 3: Same description as in Fig. 2 but using random rotation instead of a permutation.
4 Conclusions

A bonus of sparse image representation has been discussed. Namely the capability for information storage. It was shown that, in particular, this feature of sparse representations can be used, for instance, for Image Folding. While the proposed procedure is applicable through any transformation, including DCT and DWT, it was illustrated, by a number of examples, that a combination of Dirac and DC dictionaries may produce a significant gain in the sparsity of an image representation.

The performance, with regard to sparsity, of highly nonlinear techniques from redundant dictionaries encourages the application of image folding, and vise versa. The fact that storage and encryption may be implemented by simple processing steps is certainly a very interesting feature worth being further investigated.

Acknowledgements

Support from EPSRC, UK, grant EP/D062632/1, is gratefully acknowledged.

References

[1] S. Mallat. A wavelet tour of signal processing. Academic Press, London, 1998.

[2] S.S. Chen, D.L. Donoho, and M.A. Saunders. Atomic decomposition by basis pursuit. SIAM Journal on Scientific Computing, 20:33–61, 1998.

[3] S. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. IEEE Transactions on Signal Processing, 41:3397–3415, 1993.

[4] Y.C. Pati, R. Rezaifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In Proceedings of the 27th Annual Asilomar Conference in Signals, System and Computers, volume 1, pages 40–44, 1993.

[5] L. Rebollo-Neira and D. Lowe. Optimized orthogonal matching pursuit approach. IEEE Signal Processing Letters, 9:137–140, 2002.

[6] M. Andrle and L. Rebollo-Neira. A swapping-based refinement of orthogonal matching pursuit strategies. Signal Processing, 86:480–495, 2006.

[7] S. Fischer, G. Cristobal, R. Redondo, R. Sparse overcomplete Gabor wavelet representation based on local competitions. IEEE Transactions on Image Processing 15: 265–272, 2006.

[8] R. Figueras i Ventura, P. Vanderghynst, P. Frossard Low-rate and flexible image coding with redundant representations IEEE Transactions on Image Processing, 15: 726–739, 2006.

[9] D. Donoho and X. Huo. Uncertainty principles and ideal atomic decomposition. IEEE Transactions on Information Theory, 47:2845–2862, 2001.

[10] M. Eldar and A.M. Bruckstein, A generalized uncertainty principle and sparse representations of pairs of bases. IEEE Transactions on Information Theory, 48: 2558–2567, 2002.
[11] A. Feurer and A. Nemirosky, On sparse represeation in pairs of basis. *IEEE Transactions on Information Theory*, 49: 1579–1581, 2003.

[12] R. Gribonval and M. Nielsen. Sparse representations in unions of bases. *IEEE Transactions on Information Theory*, 49:3320–3325, 2003.

[13] Greed is good: algorithmic results for sparse approximation *IEEE Transactions on Information Theory*, 50: 2231–2242, 2004.

[14] B.A. Olshausen and B.J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, 1997.

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

[16] Highly nonlinear approximations for sparse signal representation. http://www.ncrg.aston.ac.uk/Projects/HNLAprox, 2009.

[17] J.Miotke, L. Rebollo-Neira Oversampling of Fourier Coefficients for Hiding Messages, Applied and Computational Harmonic Analysis, 16,3, (2004) 203–207.
