Image Decomposition: Separation of Texture from Piecewise Smooth Content

J.-L. Starck [a], M. Elad [b], and D.L. Donoho [c]

[a] DAPNIA/SEDI-SAP, Service d'Astrophysique,
CEA/Saclay, 91191 Gif sur Yvette, France
[b] Computer Science Department, Stanford University,
Stanford, CA, 94305, USA
[c] Statistic Department, Stanford University,
Stanford, CA, 94305, USA

ABSTRACT

This paper presents a novel method for separating images into texture and piecewise smooth parts. The proposed approach is based on a combination of the Basis Pursuit Denoising (BPDN) algorithm and the Total-Variation (TV) regularization scheme. The basic idea promoted in this paper is the use of two appropriate dictionaries, one for the representation of textures, and the other for the natural scene parts. Each dictionary is designed for sparse representation of a particular type of image-content (either texture or piecewise smooth). The use of BPDN with the two augmented dictionaries leads to the desired separation, along with noise removal as a by-product. As the need to choose a proper dictionary for natural scene is very hard, a TV regularization is employed to better direct the separation process. Experimental results validate the algorithm's performance.

Keywords: Basis Pursuit Denoising, Total Variation, Sparse Representations, Piecewise Smooth, Texture, Wavelet, Local DCT, Ridgelet, Curvelet.

1. INTRODUCTION

The task of decomposing signals into simpler atoms is of great interest for many applications. In such problems a typical assumption is made that the given signal is a linear mixture of several source signals of more coherent origin. This kind of problem has attracted considerable research attention recently. Independent Component Analysis (ICA) and sparsity methods are typically used for the separation of signal mixtures with varying degrees of success. A classic example is the cocktail party problem where a sound signal containing several concurrent speakers is to be decomposed into the separate speakers. In image processing, a parallel situation is encountered for example in cases of photographs containing transparent layers.

Following the work of Vese, Osher, and others,[1,2] in this paper we focus on the decomposition of a given image into a texture and natural (piecewise smooth) additive ingredients. Such separation is important for applications in image coding, and in image analysis and synthesis (see[3]). Figure 1 presents the desired behavior of the separation task at hand for two examples. As can be seen, we aim at separating these two parts on a pixel-by-pixel basis, such that if the texture appears on parts of the spatial support of the image, the separation should succeed in finding a masking map as well.

The approach we take for the separation is based on the Basis-Pursuit denoising (BPDN) algorithm, extending results from previous work.[4,5] The basic idea behind this new algorithm is to choose two appropriate dictionaries, one for the representation of textures, and the other for the natural scene parts. Both dictionaries are to be designed such that each leads to sparse representations over the images it is serving, while yielding non-sparse representations on the other content type. Thus, when combined to an overall dictionary, the BPDN is expected to lead to the proper separation, as it seeks for the overall sparsest solution, and this should align with the sparse representation for each part separately. We show experimentally how indeed the BPDN

Further author information: Send correspondence to jstarck@cea.fr

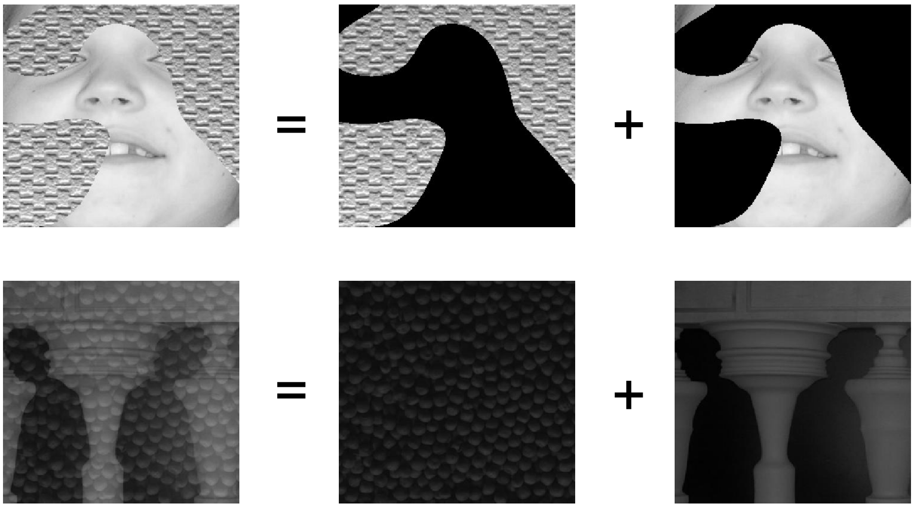

Figure 1. Our goal: Schematic illustrating ideal separation of texture from piecewise smooth contents in images.

framework leads to a successful separation. Furthermore, we show how to strengthen the BPDN paradigm, overcoming inappropriate dictionary choices, leaning on the Total-Variation (TV) regularization scheme.

The rest of the paper is organized as follows: Section 2 presents the separation method via the BPDN and the way to combine the TV for its improvement. In Section 3 we discuss the choice of dictionaries for the texture and the natural scene parts. Section 4 addressed the numerical scheme for solving the separation problem efficiently. We present several experimental results in Section 5 and conclude in Section 6.

2. SEPARATION OF IMAGES - BASICS

2.1. Model Assumption

Assume that the input image to be processed is of size $N \times N$. We represent this image as a 1D vector of length N^2 by simple reordering. For such images $\underline{X}_t$ that contain *only* pure textures we propose an over-complete representation matrix $\mathbf{T}_t \in \mathcal{M}^{N^2 \times L}$. While $\underline{X}_t = \mathbf{T}_t \underline{\alpha}_t$ has no unique solution because of over-completeness, we suppose that

$$\underline{\alpha}_t^{opt} = ArgMin_{\underline{\alpha}_t} \ \|\underline{\alpha}_t\|_0 \ \text{ subject to: } \ \underline{X}_t = \mathbf{T}_t \underline{\alpha}_t \tag{1}$$

for any texture image $\underline{X}$ leads to a very sparse solution (i.e. $\|\underline{\alpha}_t^{opt}\|_0$ is very small). We further assume that $\mathbf{T}_t$ is such that if the texture appears in parts of the image, the representation is still sparse, implying that the dictionary offers a spatially localized multi-scale representation of the image content. In words, the definition (1) yields a representation $\underline{\alpha}_t$ with maximal sparsity. Empirically we have found that if indeed the proper choice of a dictionary is made, we get very sparse representations for textures and non-sparse representations for different image types (e.g. natural scenes).

Similar to the above, assume that for images containing piecewise smooth content, $\underline{X}_n$, we have a different dictionary $\mathbf{T}_n$, such that their content is sparsely represented by a minimum ℓ^0-norm decomposition. Again, we assume that beyond the sparsity obtained by $\mathbf{T}_n$ for natural images, we can further assume that texture images are represented very inefficiently (i.e. non-sparsely), and also assume that the analysis applied by this dictionary is of multi-scale nature, enabling it to detect pieces of the desired content.

For an arbitrary image $\underline{X}$ containing both texture and piecewise smooth content (overlayed, side-by-side, or both), we seek the sparsest of all representations over the combined dictionary containing both $\mathbf{T}_t$ and $\mathbf{T}_n$. Thus we need to solve

$$\{\underline{\alpha}_t^{opt},\ \underline{\alpha}_n^{opt}\} = ArgMin_{\{\underline{\alpha}_t,\ \underline{\alpha}_n\}}\ \|\underline{\alpha}_t\|_0 + \|\underline{\alpha}_n\|_0 \ \text{ subject to: } \ \underline{X} = \mathbf{T}_t\underline{\alpha}_t + \mathbf{T}_n\underline{\alpha}_n. \tag{2}$$

We explore in this paper the proposition that this optimization task is likely to lead to a successful separation of the image content, such that $\mathbf{T}_t\underline{\alpha}_t$ is mostly texture and $\mathbf{T}_n\underline{\alpha}_n$ is mostly piecewise smooth. The reason for this expectation relies on the assumptions made earlier about $\mathbf{T}_t$ and $\mathbf{T}_n$ being very efficient in representing one phenomenon and being highly non-effective in representing the other image type.

While conceptually appealing, the problem formulated in Equation (2) is computationally intractable in general. Its complexity grows exponentially with the number of columns in the overall dictionary. The Basis Pursuit (BP) method[4] suggests the replacement of the ℓ^0-norm with an ℓ^1-norm, thus leading to a solvable optimization problem (Linear Programming) of the form

$$\{\underline{\alpha}_t^{opt},\ \underline{\alpha}_n^{opt}\} = ArgMin_{\{\underline{\alpha}_t,\ \underline{\alpha}_n\}}\ \|\underline{\alpha}_t\|_1 + \|\underline{\alpha}_n\|_1 \ \text{ subject to: } \ \underline{X} = \mathbf{T}_t\underline{\alpha}_t + \mathbf{T}_n\underline{\alpha}_n. \tag{3}$$

Interestingly, recent work shows that when (2) admits sufficiently sparse solutions, the BP form also leads to the sparsest of all representations.[6–9]

2.2. Complicating Factors and Their Treatment

The above description depends on qualitative assumptions whose failure may hinder the success of the overall separation process.

- *Assumption: The image is decomposed cleanly into texture and natural (piecewise smooth) parts.* For an arbitrary image this assumption may fail, as the image may contain additive noise that is not represented well by $\mathbf{T}_t$ or $\mathbf{T}_n$. Generally speaking, any deviation from this assumption may lead to a non-sparse pair of vectors $\{\underline{\alpha}_t^{opt},\ \underline{\alpha}_n^{opt}\}$.
- *Assumption: The chosen dictionaries are appropriate.* It is very hard to propose a dictionary that leads to sparse representations for a wide family of signals. Such is the case for textures which may come in many forms, and such is definitely the case for representing natural scenes. Also, dictionaries may not discriminate well between the two phenomenon we desire to separate. Thus, if for example, we have a dictionary $\mathbf{T}_n$ that leads to sparse representations for natural scenes, but is also known to lead to sparse representations for some textures, clearly, such a dictionary could not be used for a successful separation.

A solution for the first problem could be obtained by relaxing the equality constraint in Equation (3) to become an approximate one like so:

$$\{\underline{\alpha}_t^{opt},\ \underline{\alpha}_n^{opt}\} = ArgMin_{\{\underline{\alpha}_t,\ \underline{\alpha}_n\}}\ \|\underline{\alpha}_t\|_1 + \|\underline{\alpha}_n\|_1 + \lambda \left\|\underline{X} - \mathbf{T}_t\underline{\alpha}_t - \mathbf{T}_n\underline{\alpha}_n\right\|_2^2. \tag{4}$$

This way, if an additional content exist in the image so that it is not represented sparsely by both dictionaries, (4) may accommodate this as the residual $\underline{X} - \mathbf{T}_t\underline{\alpha}_t - \mathbf{T}_n\underline{\alpha}_n$. Hopefully, we manage to separate texture from natural scene parts, and also manage to remove additive noise as a by-product. This new formulation is familiar by the name Basis Pursuit Denoising, shown in[4] to perform well for denoising tasks.

As for the second problem, we assume that this problem is more acute for the choice of $\mathbf{T}_n$ since textures are less diverse, and a proper dictionary for such images is easier to obtain. While we will choose a specific matrix $\mathbf{T}_n$ which is generally well suited for separating piecewise smooth images from textures, we should require that the image $\mathbf{T}_n\underline{\alpha}_n$ is indeed only piecewise smooth, throwing away any other content. This is achieved by adding a TV penalty[10] to Equation (4), leading to

$$\{\underline{\alpha}_t^{opt},\ \underline{\alpha}_n^{opt}\} = ArgMin_{\{\underline{\alpha}_t,\ \underline{\alpha}_n\}}\ \|\underline{\alpha}_t\|_1 + \|\underline{\alpha}_n\|_1 + \lambda \left\|\underline{X} - \mathbf{T}_t\underline{\alpha}_t - \mathbf{T}_n\underline{\alpha}_n\right\|_2^2 + \gamma TV\{\mathbf{T}_n\underline{\alpha}_n\}. \tag{5}$$

The expression $TV\{\mathbf{T}_n\underline{\alpha}_n\}$ is essentially the computation of the image $\mathbf{T}_n\underline{\alpha}_n$ (supposed to be piecewise smooth), computing its absolute gradient field and summing it (ℓ^1-norm). Penalizing with TV, we force the image $\mathbf{T}_n\underline{\alpha}_n$ to be closer to a piecewise smooth image. We note that Candes et. al. have had a good success combining TV with other optimization schemes to enhance separation.

2.3. Different Problem Formulation

Assume that each of the chosen dictionaries is a concatenation of unitary matrices:

$$\mathbf{T}_t = [\mathbf{T}(1)_t,\ \mathbf{T}(2)_t,\ \ldots\ ,\mathbf{T}(L_t)_t] \qquad \mathbf{T}_n = [\mathbf{T}(1)_n,\ \mathbf{T}(2)_n,\ \ldots\ ,\mathbf{T}(L_n)_n]$$

and

$$\begin{aligned}\mathbf{T}(1)_t^H\mathbf{T}(1)_t &= \mathbf{T}(2)_t^H\mathbf{T}(2)_t = \cdots = \mathbf{T}(L_t)_t^H\mathbf{T}(L_t)_t \\ &= \mathbf{T}(1)_n^H\mathbf{T}(1)_n = \mathbf{T}(2)_n^H\mathbf{T}(2)_n = \cdots = \mathbf{T}(L_n)_n^H\mathbf{T}(L_n)_n = \mathbf{I}.\end{aligned}$$

In that case we could slice $\underline{\alpha}_t$ and $\underline{\alpha}_n$ into corresponding parts L_t and L_n, and obtain a new formulation of the problem

$$\begin{aligned}\min_{\{\underline{\alpha}(k)_t\}_{k=1}^{L_t},\ \{\underline{\alpha}(j)_n\}_{j=1}^{L_n}} \quad & \sum_{k=1}^{L_t}\|\underline{\alpha}(k)_t\|_1 + \sum_{j=1}^{L_n}\|\underline{\alpha}(j)_n\|_1 \\ & +\lambda\left\|\underline{X} - \sum_{k=1}^{L_t}\mathbf{T}(k)_t\underline{\alpha}(k)_t - \sum_{j=1}^{L_n}\mathbf{T}(j)_n\underline{\alpha}(j)_n\right\|_2^2 + \gamma TV\left\{\sum_{j=1}^{L_n}\mathbf{T}(j)_n\underline{\alpha}(j)_n\right\}.\end{aligned} \tag{6}$$

Defining $\underline{X}(k)_t = \mathbf{T}(k)_t\underline{\alpha}(k)_t$ and similarly $\underline{X}(j)_n = \mathbf{T}(j)_n\underline{\alpha}(j)_n$, we can reformulate the problem as

$$\begin{aligned}\min_{\{\underline{X}(k)_t\}_{k=1}^{L_t},\ \{\underline{X}(j)_n\}_{j=1}^{L_n}} \quad & \sum_{k=1}^{L_t}\left\|\mathbf{T}(k)_t^H\underline{X}(k)_t\right\|_1 + \sum_{j=1}^{L_n}\left\|\mathbf{T}(j)_n^H\underline{X}(j)_n\right\|_1 \\ & +\lambda\left\|\underline{X} - \sum_{k=1}^{L_t}\underline{X}(k)_t - \sum_{j=1}^{L_n}\underline{X}(j)_n\right\|_2^2 + \gamma TV\left\{\sum_{j=1}^{L_n}\underline{X}(j)_n\right\}\end{aligned} \tag{7}$$

and the unknowns become images, rather then representation coefficients. For this problem structure there exist a fast numerical solver called *Block-Coordinate Relaxation Method*, based on the shrinkage method.[11] This solver requires *only* the use of matrix-vector multiplications with the unitary transforms and their inverses. See[12] for more details. We remark that the same algorithm can be applied with non-unitary transforms although theoretical validation may require more challenging analysis than in the block-unitary case. In practice, block coordinate methods work well in the non-unitary cases we have explored.

To summarize so far, in order to translate the above idea into a practical algorithm we should answer two major questions: (i) How should we choose the dictionaries $\mathbf{T}_t$ and $\mathbf{T}_n$? and (ii) How should we numerically solve the obtained optimization problem in a tractable way? These two questions are addressed in the coming sections.

3. CANDIDATE DICTIONARIES

Our approach towards the choice of $\mathbf{T}_t$ and $\mathbf{T}_n$ is to pick transforms known for representing well either texture or piecewise smooth behaviors. For numerical reasons, we restrict our choices to dictionaries $\mathbf{T}_t$ and $\mathbf{T}_n$ which have a fast forward and inverse implementation. We shall start with a brief description of our candidate dictionaries:

- *Bi-Orthogonal Wavelet Transforms (OWT)*: Previous work has shown that the wavelet transform is well suited for the effective (sparse) representation of natural scenes[11] . The application of the OWT to image compression using the 7-9 filters[13] and the zero-tree coding[14, 15] has led to impressive results compared to previous methods like JPEG.

  The OWT implementation requires $O(n^2)$ operations for an image with $n \times n$ pixels, both for the forward and the inverse transforms. Represented as a matrix-vector multiplication, this transform is a square matrix, either unitary, or non-unitary with accompanying inverse matrix of a similar simple form.

  The OWT presents only a fixed number of directional elements independent of scale, and there are no highly anisotropic elements[16] . For instance, the Haar 2D wavelet transform is optimal to find features with a ratio $length/width = 2$, in a horizontal, vertical, or diagonal orientation. Therefore, we naively expect the OWT to be non-optimal for detection of highly anisotropic features. Moreover, the OWT is non-shift invariant - a property that may cause problems in signal analysis.

  The undecimated version (UWT) of the OWT is certainly the most popular transform for data filtering. It is obtained by skipping the decimation and implies that this is an overcomplete transform represented as a matrix with more columns that rows when multiplying a signal to be transform. The redundancy factor (ratio between number of columns to number of rows) is $3J + 1$ - where J being the number of resolution layers. With the over-completeness comes a shift invariance property that we desire.

- *The isotropic à trous algorithm*: The isotropic "à trous" wavelet transform algorithm decomposes an $n \times n$ image I as a superposition of the form $I(x, y) = c_J(x, y) + \sum_{j=1}^{J} w_j(x, y)$, where c_J is a coarse or smooth version of the original image I and w_j represents 'the details of I' at scale 2^{-j}, see[17] for more information. Thus, the algorithm outputs $J + 1$ sub-band arrays of size $n \times n$. (The present indexing is such that $j = 1$ corresponds to the finest scale (high frequencies)). This wavelet transform is very well adapted to the detection of isotropic features, and this explains the reason of its success for astronomical image processing, where the data contain mostly (quasi-)isotropic objects, such stars or galaxies.[18]

- *The Local Ridgelet Transform*: The two-dimensional continuous ridgelet transform of a function is defined by:

  $$\mathcal{R}_f(a, b, \theta) = \int \overline{\psi}_{a,b,\theta}(x) f(x) dx.$$

  where the ridgelet function $\psi_{a,b,\theta}$ is given by

  $$\psi_{a,b,\theta}(x) = a^{-1/2} \cdot \psi((x_1 \cos\theta + x_2 \sin\theta - b)/a); \tag{8}$$

  with $\int \psi(t)dt = 0$, $a > 0$, $b \in \mathbf{R}$ and each $\theta \in [0, 2\pi)$.

  It has been shown[16] that the ridgelet transform is precisely the application of a 1-dimensional wavelet transform to the slices of the Radon transform where the angular variable θ is constant and t is varying.

  The ridgelet transform is optimal to find only global lines (starting and ending on the image boundaries). To detect non-global line segments, a localization partitioning must be introduced.[19] The image is decomposed into smoothly overlapping blocks of side-length b pixels in such a way that the overlap between two vertically adjacent blocks is a rectangular array of size $b \times b/2$; we use overlap to avoid blocking artifacts. For a $n \times n$ image, we count $2n/b$ such blocks in each direction. The partitioning introduces redundancy (over-completeness), as each pixel belongs to 4 neighboring blocks.

  The ridgelet transform requires $O(n^2 \log_2 n)$ operations. More details on the implementation of the digital ridgelet transform can be found in.[20] The ridgelet transform is optimal to detect line or edge segment of length equal to the block size used.

- *The Curvelet Transform*: The curvelet transform, proposed by Candes and Donoho[21, 22] enables the possibility to directionally analyze an image with different angular resolutions in a single and effective transform. The idea is to first decompose the image into a set of wavelet bands, and to analyze each band with a local ridgelet transform. The block size is changed at each scale level, such that different levels of the multi-scale ridgelet pyramid are used to represent different sub-bands of a filter bank output.

The side-length of the localizing windows is doubled *at every other* dyadic sub-band, hence maintaining the fundamental property of the curvelet transform, which says that elements of length about $2^{-j/2}$ serve for the analysis and synthesis of the j-th sub-band $[2^j, 2^{j+1}]$. The curvelet transform is also redundant, with a redundancy factor of $16J+1$ whenever J scales are employed.Its complexity is of the $O(n^2 \log_2 n$, as with ridgelet. This method is best for the detection of anisotropic structures of different lengths.

- *The (Local) Discrete Cosine Transform (DCT)*: The DCT is a variant of the Discrete Fourier Transform, replacing the complex analysis with real numbers by a symmetric signal extension. The DCT is an orthonormal transform, known to be well suited for stationary signals. Its coefficients essentially represents frequency content, similar to the one obtained by Fourier analysis. When dealing with non-stationary sources, DCT is typically applied in blocks. Such is indeed the case in the JPEG image compression algorithm. Choice of overlapping blocks is preferred for analyzing signals while preventing blotchiness effects. In such a case we get again an overcomplete transform with redundancy factor of 4 for an overlap of 0.5. A fast algorithm with complexity of $n^2 \log_2 n$ exists for its computation. The DCT is appropriate for a sparse representation of smooth or periodic behaviors.

3.1. Dictionaries Choice - Summary

For texture description (i.e. $\mathbf{T}_t$ dictionary), the DCT seems to have good properties. If the texture is not spatially homogeneous, a local DCT should be preferred. The second dictionary $\mathbf{T}_n$ should be chosen depending on the content of the image. If it contains lines of a fixed size, the local ridgelet transform will be good. More generally the curvelet transform represents well edges in an images, and should be a good candidate in many cases. The un-decimated wavelet transform could be used as well. Finally, for images containing isotropic features, the isotropic à trous wavelet transform is the best. In our experiments, we have chosen images with edges, and decided to apply the texture/signal separation using the DCT and the curvelet transform.

Note that when choosing a transform, we may want to prune some of the representation coefficients for better selectivity. For example, using the DCT (for the texture part) along with the wavelet transform (for the piecewise smooth part) implies some overlap between the two, when smooth content exists in the image. Thus, the low-resolution coefficients of the DCT could be simply discarded for a better definition of the separation process.

4. NUMERICAL CONSIDERATIONS

Assume hereafter that we use the DCT (actually several local versions of it, with varying block sizes) for the texture - we denote this multi-scale dictionary by $\mathcal{D}$. Assume further that given the representation coefficients of this transform, we have a reconstruction from these DCT coefficients, denoted as $\mathcal{D}^+$. In such an inversion we refer to the frame approach that generalizes the inverse by a pseudo-inverse. Similarly, we choose the curvelet transform for the natural scene part, denote the analysis operator by $\mathcal{C}$, and the synthesis operator by $\mathcal{C}^+$.

Returning to the separation process as posed in Equation (7), we have two unknowns - $\underline{X}_t$ and $\underline{X}_n$ - the texture and the piecewise smooth images. The optimization problem to be solved is

$$\min_{\{\underline{X}_t,\ \underline{X}_n\}} \quad \|\mathcal{D}\underline{X}_t\|_1 + \|\mathcal{C}\underline{X}_n\|_1 + \lambda \left\|\underline{X} - \underline{X}_t - \underline{X}_n\right\|_2^2 + \gamma TV\left\{\underline{X}_n\right\}. \tag{9}$$

The algorithm we use is the following:

1. Initialize a coefficient $L_{\max}$, the number of iterations per layer N, and threshold $\delta = \lambda \cdot L_{\max}$.
2. Perform N times:
   - Part A - Update of $\underline{X}_n$:
     - Calculate the residual $\underline{R} = \underline{X} - \underline{X}_t - \underline{X}_n$.
     - Calculate the curvelet transform $\mathcal{C}$ of $\underline{X}_n + \underline{R}$ and obtain $\underline{\alpha}_n = \mathcal{C}(\underline{X}_n + \underline{R})$.

  - Soft* threshold the coefficient $\underline{\alpha}_n$ with the δ threshold and obtain $\hat{\underline{\alpha}}_n$.
  - Reconstruct $\underline{X}_n$ by $\underline{X}_n = \mathcal{C}^+\hat{\underline{\alpha}}_n$.

- Part B - Update of $\underline{X}_t$:
  - Calculate the residual $\underline{R} = \underline{X} - \underline{X}_t - \underline{X}_n$.
  - Calculate the local DCT transform $\mathcal{D}$ of $\underline{X}_t + \underline{R}$ and obtain $\underline{\alpha}_t = \mathcal{D}(\underline{X}_t + \underline{R})$.
  - Soft threshold the coefficient $\underline{\alpha}_t$ with the δ threshold and obtain $\hat{\underline{\alpha}}_t$.
  - Reconstruct $\underline{X}_t$ by $\underline{X}_t = \mathcal{D}^+\hat{\underline{\alpha}}_t$.
- Part C - TV Consideration: Apply the TV correction by $\underline{X}_n = \underline{X}_n - \mu\gamma\frac{\partial TV\{\underline{X}_n\}}{\partial \underline{X}_n}$. The parameter μ is chosen either by a line-search minimizing the overall penalty function, or as a fixed step-size of moderate value that guarantees convergence.

3. Update the threshold by $\delta = \delta - \lambda$.

4. If $\delta > \lambda$, return to Step 2. Else, finish.

The described algorithm uses the *Block-Coordinate Relaxation Method* discussed earlier, but employs it even though the involved dictionaries are non-unitary. Further work is required to establish the convergence of this algorithm to the desired solution of the penalty function in Equation (9). We chose this numerical scheme over the Basis Pursuit interior-point approach in[4] because it presents two major advantages:

- We do not need to keep all the transformations in memory. This is particularly important when we use redundant transformations such the un-decimated wavelet transform or the curvelet transform.
- We can add different constraints on the components. Here we applied only the *TV* constraint on one of the components, but other constraints, such as positivity, can easily be added as well.

If the texture extends across the whole image, then a global DCT should be preferred to a local DCT. Our method allows us to build easily a dedicated algorithm which takes into account the *a priori* knowledge we have on the solution for a specific problem.

5. EXPERIMENTAL RESULTS

5.1. Image Decomposition

We start the description of our experiments with a synthetically–generated image composed of a natural scene and a texture, where we have the ground truth parts to compare with. We implement the proposed algorithm with the curvelet transform (five resolution levels) for the natural scene part, and a global DCT transform for the texture. The TV parameter γ has been fixed to 0.5.

Figure 2 shows the original image (top), the two original parts the image was composed from (middle left and right) and the separated texture part (bottom left) and the separated natural part (bottom right).

The texture reconstructed component $\underline{X}_t$ (middle left), the natural scene part (middle right), and the two original parts the image was composed from (bottom left and right). As we can see, the separation is reproduced rather well.

We have also applied our method to the `Barbara` (512x512) image. We have used the curvelet transform algorithm described in[20] with the five resolution levels, and overlapping DCT transform with block size 32. The TV parameter γ has been fixed again to 0.5. Figure 3 top left and right shows respectively the `Barbara` and its upper right part. Figure 3 middle shows the reconstructed Cosine component $\underline{X}_t$ and Figure 3 bottom the reconstructed curvelet component $\underline{X}_n$.

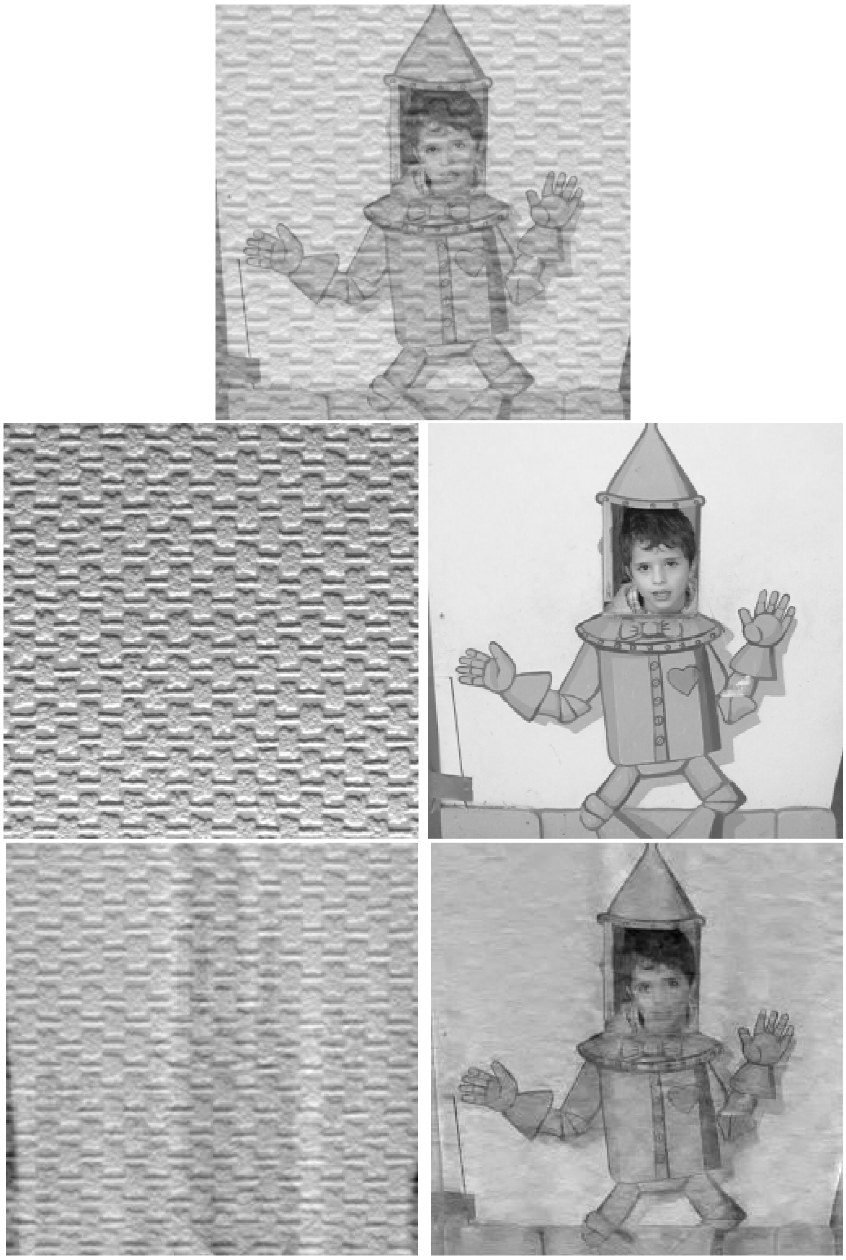

Figure 2. Original simulated image and DCT reconstructed component. Top - the combination image, Middle left - original texture part, Middle right - Original natural part, Bottom left - separated texture part, Bottom right - separated natural part.

5.2. Non–Linear Approximation

The efficiency of a given decomposition can be estimated by the non-linear approximation (NLA) scheme. Indeed, a sparse representation implies a good approximation of the image with only few coefficients. An NLA-curve is obtained by reconstructing the image from the m best term of the decomposition. For example, using

*Soft threshold is used for comparability with the ℓ^1 sparsity penalty term. However, as we have explained earlier, the ℓ^1 expression is merely a good approximation for the desired ℓ^0 one, and thus, replacing the soft by a hard threshold towards the end of the iterative process may lead to better results.

Figure 3. Top, original `Barbara` image (512x512) and upper right part of it. Middle, reconstructed DCT component. Bottom, reconstructed curvelet component.

the wavelet expansion of a function f (smooth away from a discontinuity across a C^2 curve), the best m-term approximation $\tilde{f}_m^W$ obeys[23, 24]:

$$\| f - \tilde{f}_m^W \|_2^2 \asymp m^{-1}, \quad m \to \infty, \tag{10}$$

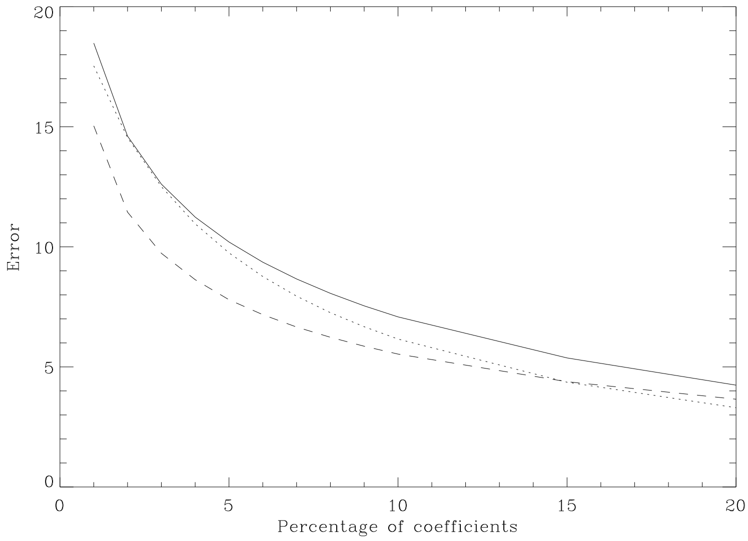

Figure 4. Standard deviation of the error of reconstructed `Barbara` image versus the m largest coefficients used in the reconstruction. Full line, DCT transform, dotted line orthogonal wavelet transform, and dashed line our signal/texture decomposition.

while for a Fourier expansion, we have

$$\| f - \tilde{f}_m^F \|_2^2 \asymp m^{-\frac{1}{2}}, \quad m \to \infty \tag{11}$$

Using the algorithm described in the previous section, we decompose the image $\underline{X}$ into two components $\underline{X}_t$ and $\underline{X}_n$ using the overcomplete transforms $\mathbf{T}_t$ and $\mathbf{T}_n$. While the decomposition is (very) redundant, the exact overall representation $\underline{X}$ may require a relatively small number of coefficients due to the promoted sparsity, and essentially yield a better NLA-curve.

Figure 4 presents the NLA-curves for the image `Barbara` using (i) the wavelet transform (OWT), (ii) the DCT, and (iii) the results of the novel algorithm discussed here, based on the OWT-DCT combination. Denoting the wavelet transform as $\mathcal{W}$ and the DCT one as $\mathcal{D}$, the representation we use includes the m largest coefficients from $\{\underline{\alpha}_t, \ \underline{\alpha}_n\} = \{\mathcal{D}\underline{X}_t, \ \mathcal{W}\underline{X}_n\}$. Using these m values we reconstruct the image by

$$\underline{\tilde{X}}_m = \mathcal{D}^{-1}\underline{\tilde{\alpha}}_t + \mathcal{W}^{-1}\underline{\tilde{\alpha}}_n.$$

The curves in Figure 4 show the representation error standard deviation as a function of m (i.e. $\mathcal{E}(m) = \sigma(\underline{X} - \underline{\tilde{X}}_m)$) relative to the `Barbara` image when using the DCT transform (continuous line), the OWT transform (dotted line) and our method (dashed line). We see that for $m < 15$ %, our representation lead to a better non linear approximation than both the DCT and the OWT separately.

6. DISCUSSION.

We have seen that our method presents several advantages compared to the standard basis pursuit. The algorithm requires less memory and additive constraint can easily added on each of the components, which gives more flexibility for a given application. Finally we have shown that it allows us to obtain very good non linear approximation when we keep only a few coefficients.

Another approach has recently been proposed[1, 25] for separating the texture v from the signal f ($f = u + v$) by using the following model $\inf_{(u,v)}(\| u \|_{BV} + \lambda \| v \|_{BV^*}$, where $f = u + v$, $u \in BV(\mathcal{R}^2)$, and $v \in BV^*(\mathcal{R}^2)$

which is the Banach space $BV^*(\mathcal{R}^2)$ containing signals with strong oscillations, and defined[1] as the space of functions $v(x, y)$ which can be written as

$$v(x, y) = \partial_x g_1(x, y) + \partial_y g_2(x, y), g_1, g_2 \in L^\infty(\mathcal{R}^2), \tag{12}$$

and

$$\| v \|_{BV^*} = \| (| g_1(x) |^2 + | g_2(x) |^2)^{\frac{1}{2}} \|_\infty \tag{13}$$

Even if nice numerical experiments have been presented in,[1] we see two drawbacks in this model:

- the λ parameter is linked to the period of the oscillations. This means that a texture containing two kinds of features with two different periods will not be well described by this model.
- non linearity: consider two images $f_1 = u + v$ and $f_2 = u + \alpha v$ where v is the texture, the minimization will not produce the same u for the two images.

These drawbacks can be considered as serious limitations of the model in practical issues.

These problems do not appear with our modelisation. Furthermore, our approach seems very promising in many practical situations, where we want to separate two or more components, each of them being sparse in a given domain associated to a transformation/reconstruction. Indeed, we address in this paper the texture/signal separation based on the DCT/curvelet construction, but this method can be seen as a specific case of a more general approach. This will be inverstigate in the future.

REFERENCES

1. L. Vese and S. Osher, "Modeling textures with total variation minimization and oscillating patterns in image processing," *Journal of Scientific Computing* , 2003. in press.
2. J. Aujol, G. Aubert, L. Blanc-Feraud, and A. Chambolle, "Image decomposition: Application to textured images and SAR images," Tech. Rep. ISRN I3S/RR-2003-01-FR, INRIA - Project ARIANA, Sophia Antipolis, 2003.
3. M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *to appear in IEEE Trans. on Image Processing* , 2003.
4. S. Chen, D. Donoho, and M. Saunder, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing* **20**, pp. 33–61, 1998.
5. J.-L. Starck, E. Candes, and D. Donoho, "Astronomical image representation by the curvelet tansform," *Astronomy and Astrophysics* **398**, pp. 785–800, 2003.
6. D. Donoho and X. Huo, "Uncertainty Principles and Ideal Atomic Decomposition," *IEEE Transactions on Information Theory* **47**(7), pp. 2845–2862, 2001.
7. D. L. Donoho and M. Elad, "Maximal sparsity representation via l_1 minimization," *Proc. Nat. Acad. Sci.* **100**, pp. 2197–2202, 2003.
8. M. Elad and A. Bruckstein, "A generalized uncertainty principle and sparse representation in pairs of bases," *IEEE Transactions on Information Theory* **48**, pp. 2558–2567, 2002.
9. R. Gribonval and M. Nielsen, "Some remarks on nonlinear approximation with schauder bases," *East J. on Approx.* **7**(2), pp. 267–285, 2001.
10. L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation noise removal algorithm," *Physica D* **60**, pp. 259–268, 1992.
11. D. Donoho and I. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biometrika* **81**, pp. 425–455, 1994.
12. A. Bruce, S. Sardy, and P. Tseng, "Block coordinate relaxation methods for nonparametric signal denoising," *Proceedings of the SPIE - The International Society for Optical Engineering* **3391**, pp. 75–86, 1998.

13. M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing* **1**, pp. 205–220, 1992.
14. J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing* **41**, pp. 3445–3462, 1993.
15. A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchival trees," *IEEE Transactions on Circuits and Systems for Video Technology* **6**, pp. 243–250, 1996.
16. E. Candès and D. Donoho, "Ridgelets: the key to high dimensional intermittency?," *Philosophical Transactions of the Royal Society of London A* **357**, pp. 2495–2509, 1999.
17. J.-L. Starck, F. Murtagh, and A. Bijaoui, *Image Processing and Data Analysis: The Multiscale Approach*, Cambridge University Press, 1998.
18. J.-L. Starck and F. Murtagh, *Astronomical Image and Data Analysis*, Springer-Verlag, 2002.
19. E. J. Candès, "Harmonic analysis of neural netwoks," *Applied and Computational Harmonic Analysis* **6**, pp. 197–218, 1999.
20. J.-L. Starck, E. Candès, and D. Donoho, "The curvelet transform for image denoising," *IEEE Transactions on Image Processing* **11**(6), pp. 131–141, 2002.
21. D. Donoho and M. Duncan, "Digital curvelet transform: strategy, implementation and experiments," in *Proc. Aerosense 2000, Wavelet Applications VII*, H. Szu, M. Vetterli, W. Campbell, and J. Buss, eds., **4056**, pp. 12–29, SPIE, 2000.
22. E. J. Candès and D. L. Donoho, "Curvelets – a surprisingly effective nonadaptive representation for objects with edges," in *Curve and Surface Fitting: Saint-Malo 1999*, A. Cohen, C. Rabut, and L. Schumaker, eds., Vanderbilt University Press, (Nashville, TN), 1999.
23. E. J. Candès and D. L. Donoho, "Recovering edges in ill-posed inverse problems: Optimality of curvelet frames," tech. rep., Department of Statistics, Stanford University, 2000.
24. M. Vetterli, "Wavelets, approximation, and compression," *IEEE Signal Processing Magazine* **18**(5), pp. 59–73, 2001.
25. Y. Meyer, "Oscillating patterns in image processing and non linear evolution equations," *University Lecture Series, AMS* **22**, 2002.