66

cea

DAPNIA

DAPNIA-04-86 04/2004

Image decomposition via the combination of sparse representations and a variational approach.

J.L. Starck, M. Elad, D.L. Donoho

Submitted to IEEE Transactions on Image Processing

Département d'**A**strophysique, de Physique des **P**articules, de Physique **N**ucléaire et de l'**I**nstrumentation **A**ssociée

DSM/DAPNIA, CEA/Saclay F - 91191 Gif-sur-Yvette Cédex
Tél : (1) 69 08 24 02 Fax : (1) 69 08 99 89
http : //www-dapnia.cea.fr

Image Decomposition Via the Combination of Sparse Representations and a Variational Approach

J.-L. Starck, M. Elad, D.L. Donoho

EDICS: 2-WAVP

J.L. Starck is with the CEA-Saclay, DAPNIA/SEDI-SAP, Service d'Astrophysique, F-91191 Gif sur Yvette, France. Email: jstarck@cea.fr.

M. Elad is with the Computer Science Department, The Technion - Israel Institute of technology, Haifa 32000 Israel. Email: elad@cs.technion.ac.il

D.L. Donoho is with the Department of Statistics, Stanford University, Sequoia Hall, Stanford, CA 94305 USA. Email: donoho@stat.stanford.edu

Abstract

The separation of image content into semantic parts plays a vital role in applications such as compression, enhancement, restoration, and more. In recent years several pioneering works suggested such separation based on a variational formulation, and others using independent component analysis and sparsity. This paper presents a novel method for separating images into texture and piecewise smooth (also referred to as "cartoon") parts, exploiting both the variational and the sparsity mechanisms, by combining the Basis Pursuit Denoising (BPDN) algorithm and the Total-Variation (TV) regularization scheme. The basic idea presented in this paper is the use of two appropriate dictionaries, one for the representation of textures, and the other for the cartoon parts, assumed to be piece-wise-smooth. Both dictionaries are chosen such that they lead to sparse representations over one type of image-content (either texture or cartoon). The use of the 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. We present several experimental results that validate the algorithm's performance. We also present a highly efficient numerical scheme to solve the combined optimization problem posed in our model.

Keywords

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

I. Introduction

The task of decomposing signals into their building 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. These kind of problems have drawn a lot of research attention in last years. 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.

An alternative approach towards the separation of image content , tailored to the problem of separating texture from non-texture parts in images, was proposed recently by Meyer [1], and followed by a practical numerical scheme devised by Vese, Osher, and others [2], [3]. This method is built on variational grounds, extending the notion of Total-Variation [4].

In this paper we follow the application posed by Meyer, Vese, Osher, and others, and focus on the decomposition of a given image into a texture and natural (piecewise smooth - cartoon) additive ingredients. The importance of such separation is for applications in image coding, and in image analysis and synthesis (see for example [5]). Figure 1 presents the desired behavior of the separation task at hand for a typical example. Note that 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 a by-product of the separation.

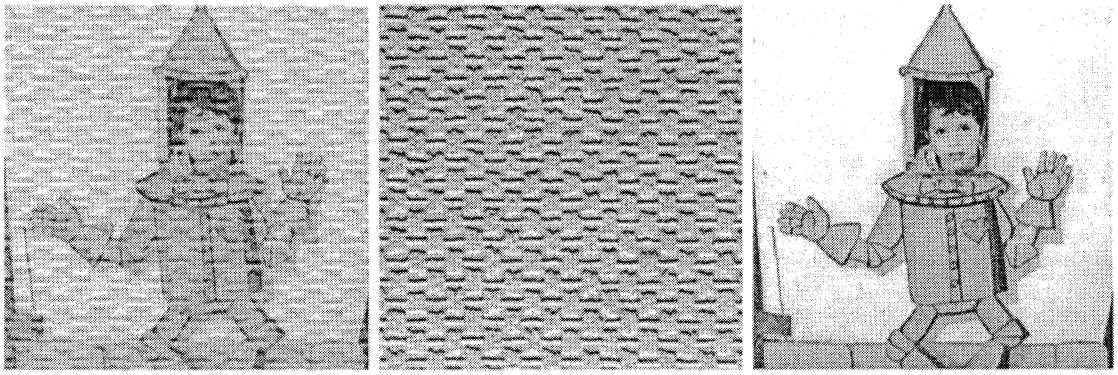

Fig. 1. Example of a separation of the form Image = texture + piecewise smooth content.

The approach we take for the separation starts with the Basis-Pursuit denoising (BPDN) algorithm, extending results from previous work [6], [7]. The basic idea behind this new algorithm is to choose two appropriate dictionaries, one for the representation of textures, and the other for the cartoon parts. Both dictionaries are to be chosen 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 framework leads to a successful separation. Further more, 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. Section 3 is devoted to a theoretic analysis of the proposed separation. In Section 4 we return to the practical side of this work and

discuss the choice of dictionaries for the texture and the cartoon parts. Section 5 addressed the numerical scheme for solving the separation problem efficiently. We present several experimental results in Section 6 and conclude in Section 7.

II. Separation of Images - Basics

A. 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 $\mathbf{T}_t \in \mathcal{M}^{N^2 \times L}$ (where typically $L \gg N^2$) such that solving

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

for any texture image $\underline{X}_t$ leads to a very sparse solution (i.e. $\|\underline{\alpha}_t^{opt}\|_0 \ll N^2$). We further assume that $\mathbf{T}_t$ is such that if the texture appears in parts of the image and otherwise zero, the representation is still sparse, implying that the dictionary employs a multi-scale and local analysis of the image content. The definition in (1) is essentially an overcomplete transform of $\underline{X}_t$, yielding a representation $\underline{\alpha}_t$, such that sparsity is maximized over the dictionary $\mathbf{T}_t$.

We further require that when this forward transform with $\mathbf{T}_t$ is applied to images containing no texture and pure piece-wise-smooth (cartoon) content, the resulting representations are non-sparse. Thus, the dictionary $\mathbf{T}_t$ plays a role of a discriminant between content types, preferring the texture over the cartoon part. A possible measure of fidelity of the chosen dictionary is the functional

$$\mathbf{T}_t^{opt} = \underset{\mathbf{T}_t}{\text{Arg min}} \ \frac{\sum_k \|\underline{\alpha}_t^{opt}(k)\|_0}{\sum_j \|\underline{\alpha}_n^{opt}(j)\|_0} \tag{2}$$

$$\text{Subject to: } \begin{array}{l} \underline{\alpha}_t^{opt}(k) = \text{Arg min}_{\underline{\alpha}_t} \ \|\underline{\alpha}_t\|_0 \ \text{ subject to: } \ \underline{X}_t(k) = \mathbf{T}_t \underline{\alpha}_t \\ \underline{\alpha}_n^{opt}(j) = \text{Arg min}_{\underline{\alpha}_n} \ \|\underline{\alpha}_n\|_0 \ \text{ subject to: } \ \underline{X}_n(j) = \mathbf{T}_t \underline{\alpha}_n. \end{array}$$

This functional of the dictionary is measuring the relative sparsity between a family of textured images $\{\underline{X}_t(k)\}_k$ and a family of cartoon content images $\{\underline{X}_n(j)\}_j$. This, or a similar measure, could be used for the design of the proper choice of $\mathbf{T}_t$, but in this paper we assume that this task is already done for us. Specifically, rather than training the dictionary based on examples, we exploit several decades of gathered experience in the field of image processing, and rely on effective representation methods found for texture images.

Similar to the above, assume that for images containing cartoon content, $\underline{X}_n$, we have a different dictionary $\mathbf{T}_n$, such that their content is sparsely represented by the above definition. Again, we assume that beyond the sparsity obtained by $\mathbf{T}_n$ for cartoon 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 and local 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 propose to seek the sparsest of all representations over the augmented dictionary containing both $\mathbf{T}_t$ and $\mathbf{T}_n$. Thus we need to solve

$$\{\underline{\alpha}_t^{opt},\ \underline{\alpha}_n^{opt}\} = \text{Arg}\min_{\{\underline{\alpha}_t,\ \underline{\alpha}_n\}} \|\underline{\alpha}_t\|_0 + \|\underline{\alpha}_n\|_0 \ \text{ subject to: } \ \underline{X} = \mathbf{T}_t\underline{\alpha}_t + \mathbf{T}_n\underline{\alpha}_n. \tag{3}$$

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 image type and being highly non-effective in representing the other.

While sensible from the point of view of the desired solution, the problem formulated in Equation (3) is non-convex and hard to solve. Its complexity grows exponentially with the number of columns in the overall dictionary. The Basis Pursuit (BP) method [6] 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

$$\{\underline{\alpha}_t^{opt},\ \underline{\alpha}_n^{opt}\} = \text{Arg}\min_{\{\underline{\alpha}_t,\ \underline{\alpha}_n\}} \|\underline{\alpha}_t\|_1 + \|\underline{\alpha}_n\|_1 \ \text{ subject to: } \ \underline{X} = \mathbf{T}_t\underline{\alpha}_t + \mathbf{T}_n\underline{\alpha}_n. \tag{4}$$

Interestingly, recent work have shown that for sparse enough solutions, the BP simpler form is accurate, also leading to the sparsest of all representations [8], [9], [10], [11]. More about this relationship is given in Section 3, where we analyze theoretically bounds on the success of such separation.

B. Complicating Factors

The above description is sensitive, and this sensitivity may hinders the success of the overall separation process. There are two complicating factors, both has to do with the assumptions made above:

• *Assumption: The image is decomposed cleanly into texture and cartoon parts.* For an arbitrary image this assumption is not true as it may also contain additive noise that is not represented well both by $\mathbf{T}_t$ and $\mathbf{T}_n$. Generally speaking, any deviation from this assumption may lead to a non-sparse pair of vectors $\{\underline{\alpha}_t^{opt},\ \underline{\alpha}_n^{opt}\}$, and with that, due to the change from ℓ^0 to ℓ^1, to a complete failure of the separation process.

• *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 cartoon images. A chosen dictionary may be inappropriate either because it does not lead to a sparse representations for the proper signals, and if this is the case, then for such images the separation will fail. More complicating scenario is obtained for dictionaries that does not discriminate well between the two contents we desire to separate. Thus, if for example, we have a dictionary $\mathbf{T}_n$ that indeed leads to sparse representations for cartoon images, but also known to lead to sparse representations for some textures, clearly, such a dictionary could not be used for a successful separation, and the result will be that part of the texture remains with the cartoon image.

A solution for the first problem could be obtained by relaxing the constraint in Equation (4) to become an approximate one. Thus, in the new form we propose solution of

$$\{\underline{\alpha}_t^{opt},\ \underline{\alpha}_n^{opt}\} = \text{Arg}\min_{\{\underline{\alpha}_t,\ \underline{\alpha}_n\}} \|\underline{\alpha}_t\|_1 + \|\underline{\alpha}_n\|_1 + \lambda\|\underline{X} - \mathbf{T}_t\underline{\alpha}_t - \mathbf{T}_n\underline{\alpha}_n\|_2^2. \tag{5}$$

This way, if an additional content exists in the image so that it is not represented sparsely by both dictionaries, the above formulation will tend to allocate this content to be the residual $\underline{X} - \mathbf{T}_t\underline{\alpha}_t - \mathbf{T}_n\underline{\alpha}_n$. This way, not only we manage to separate texture from cartoon image parts, but also succeed in removing an additive noise as a by-product. This new formulation is familiar by the name Basis Pursuit Denoising, shown in [6] to perform well for denoising tasks.

As for the second problem, we assume that this problem is more acute for the choice of $\mathbf{T}_n$. While we will choose a specific matrix $\mathbf{T}_n$ that 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 [4] to Equation (5), leading to

$$\begin{aligned}\{\underline{\alpha}_t^{opt},\ \underline{\alpha}_n^{opt}\} \quad = \quad & \text{Arg}\min_{\{\underline{\alpha}_t,\ \underline{\alpha}_n\}} \|\underline{\alpha}_t\|_1 + \|\underline{\alpha}_n\|_1 \\ & +\lambda\|\underline{X} - \mathbf{T}_t\underline{\alpha}_t - \mathbf{T}_n\underline{\alpha}_n\|_2^2 + \gamma TV\{\mathbf{T}_n\underline{\alpha}_n\}.\end{aligned} \tag{6}$$

The expression $TV\{\mathbf{T}_n\underline{\alpha}_n\}$ is essentially the computation of the image $\underline{X}_n = \mathbf{T}_n\underline{\alpha}_n$ (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, and thus support the separation process.

C. Different Problem Formulation

Assume that each of the chosen dictionaries can be composed into a set of unitary matrices such that

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

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

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

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

$$\begin{aligned}\min_{\{\underline{X}(k)_t\}_{k=1}^{L_t},\ \{\underline{X}(j)_n\}_{j=1}^{L_n}} \quad & \sum_{k=1}^{L_t}\left\|\mathbf{T}(k)_t^H\underline{X}(k)_t\right\|_1 + \sum_{j=1}^{L_n}\left\|\mathbf{T}(j)_n^H\underline{X}(j)_n\right\|_1 \\ & +\lambda\left\|\underline{X} - \sum_{k=1}^{L_t}\underline{X}(k)_t - \sum_{j=1}^{L_n}\underline{X}(j)_n\right\|_2^2 + \gamma TV\left\{\sum_{j=1}^{L_n}\underline{X}(j)_n\right\}\end{aligned} \tag{8}$$

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 [12]. This solver (see Appendix I for details) requires *only* the use of

matrix-vector multiplications with the unitary transforms and their inverses. See [13] for more details. We will return to this form of solution when we discuss numerical algorithms., and show a sub-optimal, yet very simple, algorithm for the separation task.

D. Summary of Method

In order to translate the above idea into a practical algorithm we should answer three major questions: (i) Is there a theoretical backup to the heuristic claims made here? (ii) How should we choose the dictionaries $\mathbf{T}_t$ and $\mathbf{T}_n$? and (iii) How should we numerically solve the obtained optimization problem in a traceable way? These three questions are addressed in the coming sections.

III. Theoretic Analysis of the Separation Task

Our theoretical analysis embarks from Equation (3), which stands as the basis for the separation process. This equation could also be written differently as

$$\begin{aligned} \underline{\alpha}_{all}^{opt} &= \begin{bmatrix} \underline{\alpha}_t^{opt} \\ \underline{\alpha}_n^{opt} \end{bmatrix} = \text{Arg} \min_{\{\underline{\alpha}_t,\, \underline{\alpha}_n\}} \left\| \begin{bmatrix} \underline{\alpha}_t \\ \underline{\alpha}_n \end{bmatrix} \right\|_0 \\ &\text{subject to: } \underline{X} = \begin{bmatrix} \mathbf{T}_t & \mathbf{T}_n \end{bmatrix} \begin{bmatrix} \underline{\alpha}_t \\ \underline{\alpha}_n \end{bmatrix} = \mathbf{T}_{all}\underline{\alpha}_{all}. \end{aligned} \tag{9}$$

From [9] we recall the definition of the *Spark*:

Definition 1: *Given a matrix* $\mathbf{A}$*, its Spark* ($\sigma_A = Spark\{\mathbf{A}\}$) *is defined as the minimal number of columns from the matrix that form a linearly dependent set.*

Based on this we have the following result in [9] that gives a guarantee for global optimum of (9) based on a sparsity condition:

Theorem 1: *If a candidate representation* $\underline{\alpha}_{all}$ *satisfies* $\|\underline{\alpha}_{all}\|_0 < Spark\{\mathbf{T}_{all}\}/2$*, then this solution is necessarily the global minimum of (9).*

Based on this result it is clear that the higher the value of the *Spark*, the stronger this result is. Immediate implication from the above is the following observation, referring to the success of the separation process:

Corollary 1: *If the image* $\underline{X} = \underline{X}_t + \underline{X}_n$ *is built such that* $\underline{X}_t = \mathbf{T}_t\underline{\alpha}_t$ *and* $\underline{X}_n = \mathbf{T}_n\underline{\alpha}_n$*, and* $\|\underline{\alpha}_t\|_0 + \|\underline{\alpha}_n\|_0 < Spark\{\mathbf{T}_{all}\}/2$ *is true, then the global minimum of (9) is necessarily the desired separation.*

Proof: The proof is simple deduction from Theorem 1. □

Actually, a stronger claim could be given if we assume a successful choice of dictionaries $\mathbf{T}_t$ and $\mathbf{T}_n$. Let us define a variation of the *Spark* that refers to the interface between atoms from two dictionaries:

Definition 2: *Given two matrices* $\mathbf{A}$ *and* $\mathbf{B}$*, their Inter–Spark* ($\sigma_{A\leftrightarrow B} = Spark\{\mathbf{A},\mathbf{B}\}$) *is defined as the minimal number of columns from the concatenated matrix* $[\mathbf{A},\mathbf{B}]$ *that form a linearly dependent set, and such that columns from both matrices participate in this combination.*

Suppose that for a pair of matrices $\mathbf{A}$ and $\mathbf{B}$, we have σ_A, σ_B, and $\sigma_{A\leftrightarrow B}$. Given a vector $\underline{X}$ with a representation with respect to the columns $\mathbf{A}$ with less than $\sigma_A/2$ elements, it is the sparsest possible. Similar claim could be given with respect to $\mathbf{B}$. If the representation is computed with respect to the concatenated matrix $[\mathbf{A},\mathbf{B}]$, the overall *Spark* is necessarily smaller, as the following Lemma suggests:

Lemma 1: *For a pair of matrices* $\mathbf{A}$ *and* $\mathbf{B}$ *we have that* $\sigma_{[A,B]} = \min(\sigma_A, \sigma_B, \sigma_{A\leftrightarrow B})$.

Proof: Whatever set of columns is taken from the concatenated matrix $[\mathbf{A},\mathbf{B}]$, it could be coming purely from $\mathbf{A}$, $\mathbf{B}$, or as a mixture of both. For each of these three options there is a separate bound (the three *Spark* values given). Thus, the overall bound is the weakest of them, as claimed, and this bound is essentially tight by definition. □

The minimal *Spark* for $[\mathbf{A},\mathbf{B}]$ is obtained if there is one column in $\mathbf{A}$ that appears in $\mathbf{B}$ (up to a multiplication by a constant), and this corresponds to $\sigma_{A\leftrightarrow B} = 2$. The best scenario is obtained if $\sigma_{A\leftrightarrow B} \geq \min(\sigma_A, \sigma_B)$, then the *spark* is maximal, being $\sigma_{[A,B]} = \min(\sigma_A, \sigma_B)$.

An important feature of our problem is that the goal is the successful separation of content of an incoming image and not finding the true sparse representation per each part. Thus, a stronger claim can be made:

Corollary 2: *Suppose the image* $\underline{X} = \underline{X}_t + \underline{X}_n$ *is built such that* $\underline{X}_t = \mathbf{T}_t\underline{\alpha}_t$ *and* $\underline{X}_n = \mathbf{T}_n\underline{\alpha}_n$. *If* $\|\underline{\alpha}_t\|_0 + \|\underline{\alpha}_n\|_0 < \sigma_{\mathbf{T}_t\leftrightarrow\mathbf{T}_n}/2$ *and* $\|\underline{\alpha}_t\|_0, \|\underline{\alpha}_n\|_0 > 0$ *(i.e., there is an active mixture of the two), then if the global minimum of (9) satisfies* $\|\underline{\alpha}_t^{opt}\|_0, \|\underline{\alpha}_n^{opt}\|_0 > 0$*, it is necessarily the successful separation.*

Proof: Given a mixture of columns from the two dictionaries, by the definition of the *Inter–Spark* it is clear that if there are fewer than $\sigma_{\mathbf{T}_t\leftrightarrow\mathbf{T}_n}/2$ non-zeros in such combination, it must be the unique sparsest solution. □

The new bound is higher than $Spark\{\mathbf{T}_{all}\}/2$ and therefore this result is indeed stronger. A

design goal for the choice of the two dictionaries is to get a high value of $\sigma_{\mathbf{T}_t \leftrightarrow \mathbf{T}_n}$ in order to make the best out of the above property.

Alternative approach, simpler but also weaker, towards the same analysis, could be proposed based on the notion of mutual incoherence [9], defined as

Definition 3: *Given a matrix* $\mathbf{A}$*, its Mutual–Incoherence*$\{\mathbf{A}\} = M_A$ *is defined as the maximal off-diagonal entry in the absolute Gram matrix* $|\mathbf{A}^H\mathbf{A}|$.

The *Mutual–Incoherence* is closely related to the *Spark*, and thus one can similarly define a similar notion of *Inter–*M_A. However, we leave this for future research.

So far we concentrated on Equation (3) which stands as the ideal (but impossible) tool for the separation. An interesting question is why should the ℓ^1 replacement succeed in the separation as well. We have the following result in [9]:

Theorem 2: *If the solution* $\underline{\alpha}_{all}^{opt}$ *of (9) satisfies* $\|\underline{\alpha}_{all}^{opt}\|_0 < (1/M_{\mathbf{T}_{all}} + 1)/2$*, then the* ℓ^1 *minimization alternative is guaranteed to find it.*

For the separation task, this Theorem implies that the separation via (4) is successful if it is based on sparse enough ingredients:

Corollary 3: *If the image* $\underline{X} = \underline{X}_t + \underline{X}_n$ *is built such that* $\underline{X}_t = \mathbf{T}_t\underline{\alpha}_t$ *and* $\underline{X}_n = \mathbf{T}_n\underline{\alpha}_n$*, and* $\|\underline{\alpha}_t\|_0 + \|\underline{\alpha}_n\|_0 < (1/M_{\mathbf{T}_{all}} + 1)/2$ *is true, then the solution of (4) leads to the global minimum of (9) and this is necessarily the desired separation.*

Proof: The proof is simple deduction from Theorem 2. □

We choose to stop this analysis here, as we concentrate in this paper on the applicative part. We should note that the bounds given here are quite restrictive and does not reflect truly the much better empirical results. We regard this analysis as merely supplying a theoretical motivation, rather than complete justification for the later results. We should also note that the above analysis is coming form a *worst-case* point of view (e.g., see the definition of the *Spark*), as opposed to the average case we expect to encounter empirically. Nevertheless, the ability to prove perfect separation in a stylized application without noise and with restricted success is of great benefit as a proof of concept. Further work is required to extend the theory developed here to the average case.

In order to demonstrate the gap between theoretical results and empirical evidence in Basis Pursuit separation performance, figure 2 presents a simulation of the separation task for the case of signal $\underline{X}$ of length 64, a dictionary built as the combination of the Hadamard unitary matrix

(assumed to be $\mathbf{T}_t$) and the identity matrix (assumed to be $\mathbf{T}_n$). We randomly generate sparse representations with varying number of non-zeros in the two parts of the representation vector (of length 128), and present the empirical probability (based on averaging 100 experiments) to recover correctly the separation.

For this case, Corollary 3 suggests that the number of non-zeros in the two parts should be smaller than $0.5 \cdot (1 + 1/M) = (1 + \sqrt{64})/2 = 4.5$. Actually a better result exists for this case in [9] due to the construction of the overall dictionary as a combination of two unitary matrices. Thus, the better bound is $(\sqrt{2} - 0.5))/M = 7.3$. Both these bounds are overlayed on the empirical results in the figure, and as can be seen, Basis Pursuit succeeds well beyond these bounds. Moreover, this trend is expected to strengthen as the signal size grows, since than the worst-case-scenarios (for which the bounds refer to) become of smaller probability and of less affect on the average result.

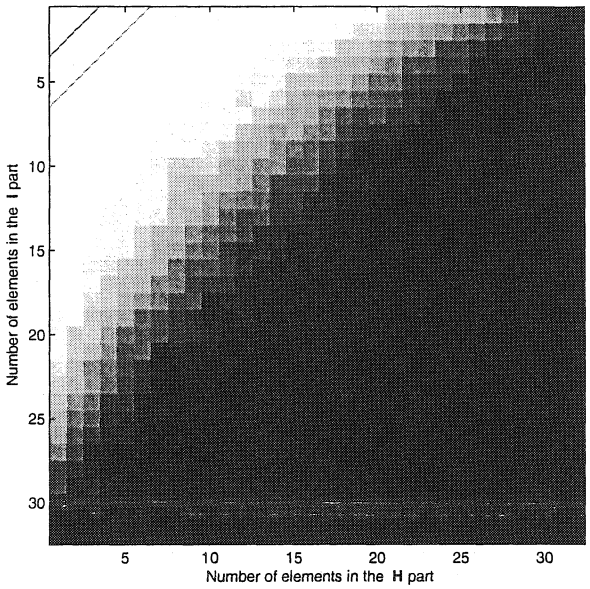

Fig. 2. The empirical probability of a success of the Basis Pursuit algorithm for separation of sources. Per every sparsity combination, 100 experiments are performed and the success rate is computed. Theoretical bounds are also drawn for comparison.

IV. Candidate Dictionaries

We are returning to the practice of separating the texture from the cartoon part, and in this section we discuss the choice of proper dictionaries.

Our approach towards the choice of $\mathbf{T}_t$ and $\mathbf{T}_n$ is to pick known transforms, and not design those optimally as we hinted earlier as a possible method. We choose transforms known for representing well either texture or piecewise smooth behaviors. For numerical reasons, we restrict our choices to the dictionaries $\mathbf{T}_t$ and $\mathbf{T}_n$ which have a fast forward and inverse implementation. In making a choice for a transform, we use experience of the user applying the separation algorithm, and the choices made may vary from one image to another. We shall start with a brief description of our candidate dictionaries.

A. Bi-Orthogonal Wavelet Transforms (OWT)

Previous work has shown that the wavelet transform is well suited for the effective (sparse) representation of piece-wise smooth images[1] [12]. The application of the OWT to image compression using the 7-9 filters [14] and the zero-tree coding [15], [16] has lead 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 scales, and there is no highly anisotropic elements [17] . 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 invariance - 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. This in turn implies that this transform is an overcomplete one, represented as a matrix with more rows that columns when multiplying a signal to be transformed. The redundancy factor (ratio between number of columns to number of rows) is $3J + 1$, J being the number of resolution layers. With the over-completeness comes the desirable shift invariance property.

[1]This is especially true if the singularities in the image are point-like, rather than organized as smooth curves.

B. The isotropic à trous algorithm

This transfrom decomposes a $N \times N$ 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 [18] 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 contains mostly (quasi-)isotropic objects, such stars or galaxies [19].

C. The Local Ridgelet Transform

The two-dimensional continuous ridgelet transform of a function is defined by:

$$\mathcal{R}_f(a,b,\theta) = \int_{x_1} \int_{x_2} \overline{\psi}_{a,b,\theta}(x_1,x_2) f(x_1,x_2) dx_1 dx_2.$$

where the ridgelet function $\overline{\psi}_{a,b,\theta}(x_1,x_2)$ is given by

$$\overline{\psi}_{a,b,\theta}(x_1,x_2) = a^{-1/2} \cdot \psi((x_1 \cos\theta + x_2 \sin\theta - b)/a); \tag{10}$$

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

It has been shown [17] 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 in finding global lines (starting and ending on the image boundaries). To detect line segments, a partitioning must be introduced [22]. The image is decomposed into smoothly overlapping blocks of side-length $b \times b$ 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 [21]. The ridgelet transform is optimal to detect line or edge segment of length equal to the block size used.

D. The Curvelet Transform

The curvelet transform, proposed in [22], [23], enables the directional analysis of the image with different scales, 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.

E. 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 blotckiness 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.

F. Dictionaries Choice - Summary

For the texture description (i.e. $\mathbf{T}_t$ dictionary), the DCT seems to have good properties. If the texture is not homogeneous, a local DCT should be preferred. The second dictionary $\mathbf{T}_n$ should be chosen depending of 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, although we expect its performance to be weaker compared to curvelets. 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. Alternatively, we can discard of the low-frequencies of the image, prior to the separation, and allocate this content to the cartoon part afterwards.

V. Numerical Considerations

A. Numerical Scheme

Returning to the separation process as posed in Equation (6), we need to solve an optimization problem of the form

$$\begin{aligned}\{\underline{\alpha}_t^{opt},\ \underline{\alpha}_n^{opt}\} &= \text{Arg}\min_{\{\underline{\alpha}_t,\ \underline{\alpha}_n\}} \|\underline{\alpha}_t\|_1 + \|\underline{\alpha}_n\|_1 \\ &\quad +\lambda\|\underline{X} - \mathbf{T}_t\underline{\alpha}_t - \mathbf{T}_n\underline{\alpha}_n\|_2^2 + \gamma TV\{\mathbf{T}_n\underline{\alpha}_n\}.\end{aligned} \tag{11}$$

Instead of solving this optimization problem, of finding two representation vectors $\underline{\alpha}_t^{opt}$ and $\underline{\alpha}_n^{opt}$, let us reformulate the problem so as to get the texture and the cartoon images, $\underline{X}_t$ and $\underline{X}_n$, as our unknowns. The reason behind this change is the obvious simplicity caused by searching shorter vectors - representation vectors are far longer than the images they represent for overcomplete dictionaries as the ones we use here.

Define $\underline{X}_t = \mathbf{T}_t\underline{\alpha}_t$ and similarly $\underline{X}_n = \mathbf{T}_n\underline{\alpha}_n$. Given $\underline{X}_t$, we can recover $\underline{\alpha}_t$ as $\underline{\alpha}_t = \mathbf{T}_t^+\underline{X}_t + \underline{r}_t$ where $\underline{r}_t$ is an arbitrary vector in the null-space of $\mathbf{T}_t$. Put these back into (6) we obtain

$$\begin{aligned}\{\underline{X}_t^{opt},\ \underline{X}_n^{opt}\} &= \text{Arg}\min_{\{\underline{X}_t,\ \underline{X}_n,\ \underline{r}_t,\ \underline{r}_n\}} \|\mathbf{T}_t^+\underline{X}_t + \underline{r}_t\|_1 + \|\mathbf{T}_n^+\underline{X}_n + \underline{r}_n\|_1 \\ &\quad +\lambda\|\underline{X} - \underline{X}_t - \underline{X}_n\|_2^2 + \gamma TV\{\underline{X}_n\} \\ &\quad \text{Subject to: } \mathbf{T}_t\underline{r}_t = 0\ ,\ \mathbf{T}_n\underline{r}_n = 0.\end{aligned} \tag{12}$$

The term $\mathbf{T}_t^+\underline{X}_t$ is an overcomplete linear transform of the image $\underline{X}_t$. Assume hereafter that we use the DCT (actually several local versions of it, with varying block sizes) for this texture part. Similarly, $\mathbf{T}_n^+\underline{X}_n$ is an overcomplete linear transform of the cartoon part - in our experiments it is chosen as the curvelet transform.

In our attempts to replace the representation vectors as unknowns, we see that we have a pair of residual vectors to be found as well. If we choose (rather arbitrarily at this stage) to assign those vectors to be zeros we obtain

$$\{\underline{X}_t^{opt},\ \underline{X}_n^{opt}\} = \text{Arg}\min_{\{\underline{X}_t,\ \underline{X}_n\}} \|\mathbf{T}_t^+\underline{X}_t\|_1 + \|\mathbf{T}_n^+\underline{X}_n\|_1 + \lambda\|\underline{X}-\underline{X}_t-\underline{X}_n\|_2^2 + \gamma TV\{\underline{X}_n\}. \quad (13)$$

We can justify the choice $\underline{r}_t = \underline{0},\ \underline{r}_n = \underline{0}$ in several ways:

Bounding function: Consider the function posed on (12) as a function of $\underline{X}_t,\ \underline{X}_n$, where per every possible values of those two images we optimize with respect to $\underline{r}_t,\ \underline{r}_n$. Comparing this function to the one we have suggested in (13), the new function could be referred to as an upper bounding surface to the true function. Thus, in minimizing it instead, we can guarantee that the true function to be minimized is of even lower value.

Relation to the Block-Coordinate-Relaxation algorithm: Comparing (13) to the case discussed in Equation (8), we see close resemblance, if we assume that the dictionaries involved contain just one unitary part. In this case we get a complete equivalence between solving (12) and (13). In a way we may refer to the approximation we have made here as a method to generalize the block-coordinate-relaxation method for the non-unitary case.

Relation to MAP: The expression written as penalty function in (13) has a Maximal-A-Posteriori estimation flavor to it. It suggests that the given image $\underline{X}$ is known to originate from a linear combination of the form $\underline{X}_t+\underline{X}_n$, contaminated by Gaussian noise - this part comes from the likelihood term $\|\underline{X}-\underline{X}_t-\underline{X}_n\|_2^2$. For the texture image part there is the assumption that it comes from a Gibbs distribution of the form $\text{Const}\cdot\exp\left(-\beta_t\|\mathbf{T}_t^+\underline{X}_t\|_1\right)$. Similarly, the cartoon part is assumed to originate from a prior of the form $\text{Const}\cdot\exp\left(-\beta_n\|\mathbf{T}_n^+\underline{X}_n\|_1-\gamma_n TV\{\underline{X}_n\}\right)$. While different from our original point of view, these assumptions are reasonable and not far from the Basis Pursuit approach.

The bottom line to all this discussion is that we have chosen an approximation to our true minimization task, and with it managed to get a simplified optimization problem, for which an effective algorithm can be proposed. Our minimization task is thus given by

$$\min_{\{\underline{X}_t,\ \underline{X}_n\}} \left\|\mathbf{T}_t^+\underline{X}_t\right\|_1 + \left\|\mathbf{T}_n^+\underline{X}_n\right\|_1 + \lambda\left\|\underline{X}-\underline{X}_t-\underline{X}_n\right\|_2^2 + \gamma TV\left\{\underline{X}_n\right\}. \quad (14)$$

The algorithm we use is based on the Block-Coordinate-Relaxation method [13] (see Appendix

I), with some required changes due to the non-unitary transforms involved, and the additional TV term. The algorithm is given below:

1. Initialize $L_{\max}$, number of iterations per layer N, and threshold $\delta = \lambda \cdot L_{\max}$.
2. Perform N times:
   - Part A - Update of $\underline{X}_n$ assuming $\underline{X}_t$ is fixed:
     - Calculate the residual $\underline{R} = \underline{X} - \underline{X}_t - \underline{X}_n$.
     - Calculate the curvelet transform of $\underline{X}_n + \underline{R}$ and obtain $\underline{\alpha}_n = \mathbf{T}_n^+(\underline{X}_n + \underline{R})$.
     - Soft threshold the coefficient $\underline{\alpha}_n$ with the δ threshold and obtain $\hat{\underline{\alpha}}_n$.
     - Reconstruct $\underline{X}_n$ by $\underline{X}_n = \mathbf{T}_n\hat{\underline{\alpha}}_n$.
   - Part B - Update of $\underline{X}_t$ assuming $\underline{X}_n$ is fixed:
     - Calculate the residual $\underline{R} = \underline{X} - \underline{X}_t - \underline{X}_n$.
     - Calculate the local DCT transform of $\underline{X}_t + \underline{R}$ and obtain $\underline{\alpha}_t = \mathbf{T}_t^+(\underline{X}_t + \underline{R})$.
     - Soft threshold the coefficient $\underline{\alpha}_t$ with the δ threshold and obtain $\hat{\underline{\alpha}}_t$.
     - Reconstruct $\underline{X}_t$ by $\underline{X}_t = \mathbf{T}_t\hat{\underline{\alpha}}_t$.
   - 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.

Algorithm 1 - The numerical algorithm for minimizing (14).

In the above algorithm, soft threshold is used due to our formulation of 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.

We chose this numerical scheme over the Basis Pursuit interior-point approach in [6], because it presents two major advantages: (i) 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. Also, (ii) 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 is the same on 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.

B. TV and Undecimated Haar Transform

The link between the TV constraint and the undecimated Haar wavelet soft thresholding has been studied in [6] and later in [24]. It has been shown that iterating using the undecimated Haar wavelet soft thresholding with just one resolution leads to the same results as the TV constraint. In light of this interpretation, we can change part C in the above algorithm this way:

> Part C - TV Consideration: Apply the TV correction by using the undecimated Haar wavelet transform $\mathcal{H}$ and a soft thresholding:
> - Calculate the undecimated Haar wavelet transform of X_n and obtain $\underline{\alpha}_h = \mathcal{H}\underline{X}_n$.
> - Soft threshold the coefficient $\underline{\alpha}_h$ with the γ threshold and obtain $\underline{\hat{\alpha}}_h$.
> - Reconstruct $\underline{X}_n$ by $\underline{X}_n = \mathcal{H}^{-1}\underline{\hat{\alpha}}_h$.

Algorithm 2 - The alternative to TV based on Haar wavelet.

This method is expected to lead to better results compared to the regular TV one as it introduces multi-resolution TV due to the several layers in the undecimated Haar wavelet transform employed.

C. Noise Consideration

The case of noisy data – additive noise, white, and independent of the texture and the cartoon parts – can be easily considered in our framework, and merged into the algorithm such that we get a three-way separation to texture, cartoon, and additive noise.

For simplicity, we assume that the two transforms $\mathbf{T}_t^+$ and $\mathbf{T}_n^+$ are normalized, so that for a given noise realization $\underline{V}$ with zero mean and a standard deviation equals to 1, $\alpha_n = \mathbf{T}_n^+\underline{V}$ and $\alpha_t = \mathbf{T}_t^+\underline{V}$ have also a standard deviation equals to 1.

Then, only the last step of the algorithm need to to changed. Indeed, by just replacing the stopping criterion $\delta > \lambda$ by $\delta > k\sigma$, where σ is the noise standard deviation and k a value in the interval $[3, 4]$. This ensures that non-significant coefficients (i.e. coefficients with an absolute value lower than $k\sigma$) are never taken into account. The final image is therefore be decomposed in three images,

$$\underline{X} = \underline{X}_t + \underline{X}_n + \underline{V},$$

where $\underline{V}$ is the residual, considered as our noise estimation.

VI. Experimental Results

A. Image Decomposition

We start the description of our experiments with a synthetically generated image composed of a cartoon and a texture, where we have the ground truth parts to compare with. We implemented the proposed algorithm with the curvelet transform (five resolution levels) for the cartoon part, and a global DCT transform for the texture. We have used the soft thresholding version of the TV, as described in previous section. The TV parameter γ has been fixed to 2. In this example, we got better results if the very low frequency components of the image is first subtracted from it, and then added to $\underline{X}_n$ after the separation. The results are shown in Figure 3.

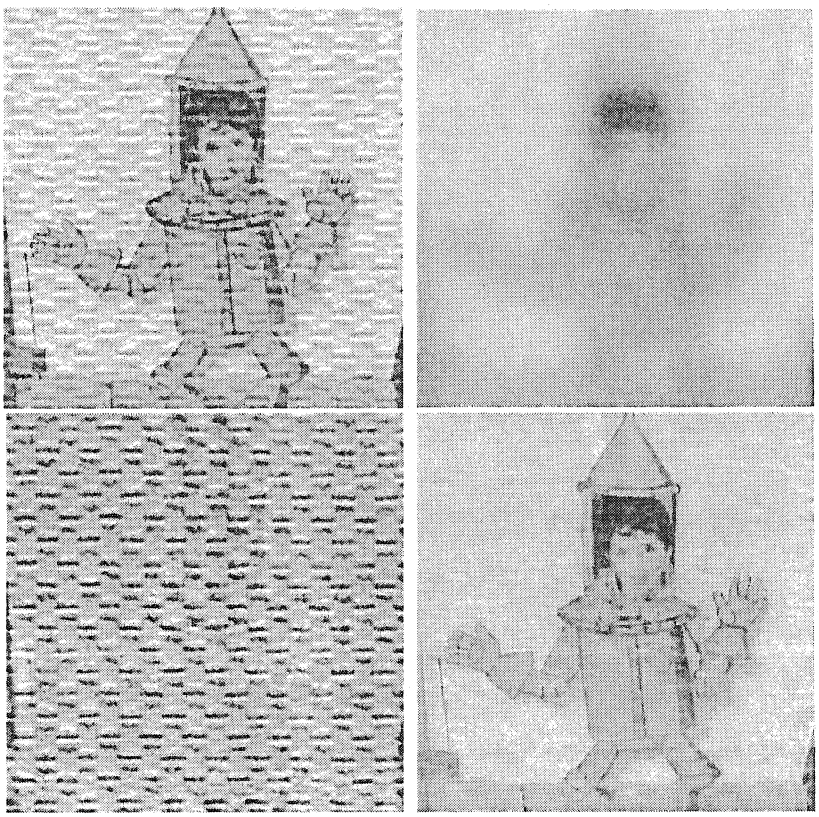

Fig. 3. Top left - the original combination image, Top right - the low frequency part taken out. Bottom left - separated texture part, Bottom right - separated cartoon part.

Figure 3 shows the original combined image at the top left part. The low-frequency content removed prior to the separation is shown in the top right part of the figure. The separated texture component $\underline{X}_t$ and the cartoon part $\underline{X}_n$ are shown at the bottom. As we can see, the separation is reproduced rather well.

Figure 4 shows the results on the same image as in Figure 3. In this experiment a Gaussian noise ($\sigma = 10$) was added to the original image. As we can, even in the presence of noise, the method is able to perform a separation between the texture and cartoon very well. Moreover, the additive noise is separated successfully.

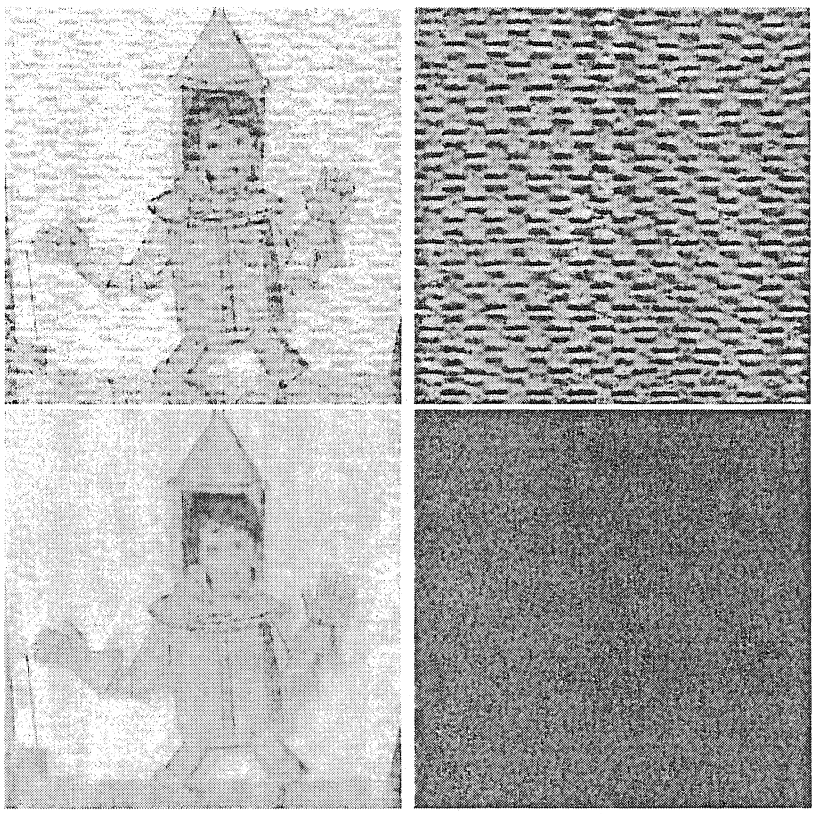

Fig. 4. Top left - original image containing a texture part, cartoon, and a Gaussian noise (sigma=10). Top right - separated texture part. Bottom right - separated cartoon part, Bottom left - noise component estimation.

We have also applied our method to the `Barbara` (512x512) image. We have used the curvelet transform algorithm described in [21] with the five resolution levels, and overlapping DCT transform with a block size equals to 32. The TV parameter γ has been fixed to 0.5.

Figure 5 shows respectively the original `Barbara` image, the reconstructed cosine component $\underline{X}_t$ and the reconstructed curvelet component $\underline{X}_n$. Figure 6 top left and right shows a magnified part of these two components. For comparison, the bottom part shows the separated components reconstructed by Vese-Osher approach [2]. As can e seen, the results are comparable, with some differences we attribute mostly to parameter setup. As discussed in Appendix II, these two

alternative methods are very much alike, although developed from different origins.

Fig. 5. Top - original Barbara image (512x512), Bottom left - reconstructed DCT (texture) component, Bottom right - reconstructed curvelet (cartoon) component.

B. Non Linear Approximation

The efficiency of a given decomposition can be estimated by the non-linear approximation (NLA) scheme, where one represents the signal based on the leading coefficients (in size and not location!) and see how the representation error behaves. 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 coefficients of the decomposition.

For example, using the wavelet expansion of a function f (smooth away from a discontinuity across a C^2 curve), the best m-terms approximation $\tilde{f}_m^W$ obeys [25], [26] $\| f - \tilde{f}_m^W \|_2^2 \asymp m^{-1}, \quad m \to \infty$, while for a Fourier expansion, we have $\| f - \tilde{f}_m^F \|_2^2 \asymp m^{-\frac{1}{2}}, \quad m \to \infty$. Using the algorithm described in the previous section, we decompose the image $\underline{X}$ into two components $\underline{X}_t$ and $\underline{X}_n$

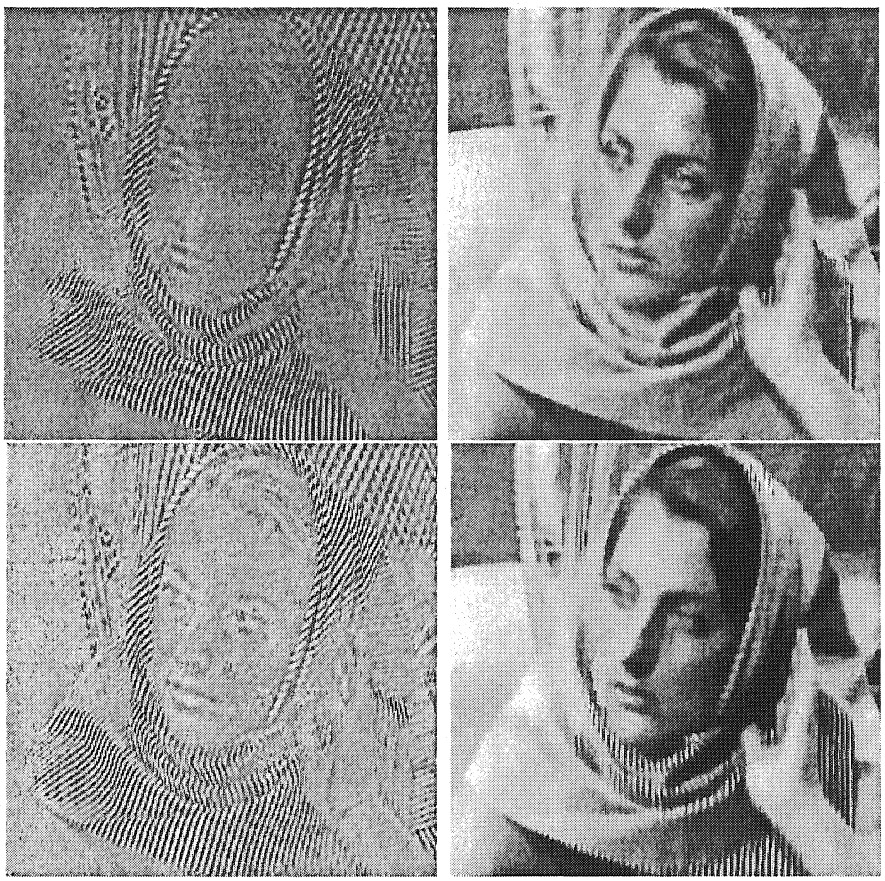

Fig. 6. Top - reconstructed DCT and curvelet components by our method, Bottom, texture and cartoon components using Vese and Osher's algorithm.

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

$$\underline{\tilde{X}}_m = \mathbf{T}_t\underline{\tilde{\alpha}}_t + \mathbf{T}_t\underline{\tilde{\alpha}}_n.$$

The curves in Figure 7 show the representation error standard deviation as a function of m (i.e. $\mathcal{E}(m) = \sigma(\underline{X} - \underline{\tilde{X}}_m)$). We see that for $m < 15$ %, our representation lead to a better non linear

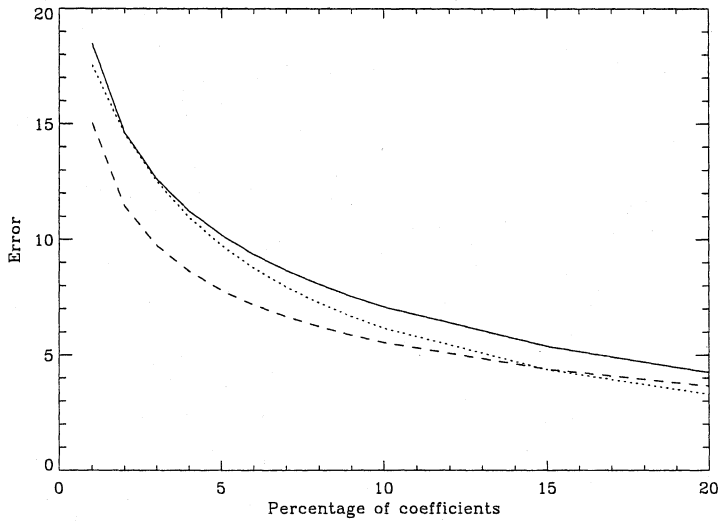

Fig. 7. Standard deviation of the error of reconstructed `Barbara` image versus the m largest coefficients used in the reconstruction. Full line, DCT transform, dotted line orthogonal wavelet transform, and dashed line our signal/texture decomposition.

approximation than both the DCT and the OWT separately.

C. Basic Applications

The ability to separate the image as we show has many applications. We sketch here two such simple experiments to illustrate the importance of a successful separation.

Edge detection is a crucial processing step in many vision applications. When the texture is highly contrasted, most of the detected edges are due the texture rather than to the cartoon part. By separating first the two components, texture and cartoon part, we can detect the edges on the cartoon component. Figure 8 shows the edges detected by the Canny algorithm on both the original image (see Figure 1) and the curvelet reconstructed component (see figure 3 bottom right).

Fig. 9 upper left shows a galaxy, imaged with the GEMINI-OSCIR instrument at 10 μm. The data is contaminated by a noise and a stripping artifact (assumed to be the texture in the image) due to the instrument electronics. As the galaxy is isotropic, we have preferred to use the isotropic wavelet transform instead of the curvelet transform. Fig. 9 summarizes the results of the separation where we see a successful isolation of the galaxy, the textured disturbance, and the additive noise. We can refer to the entire separation process as a noise removal algorithm,

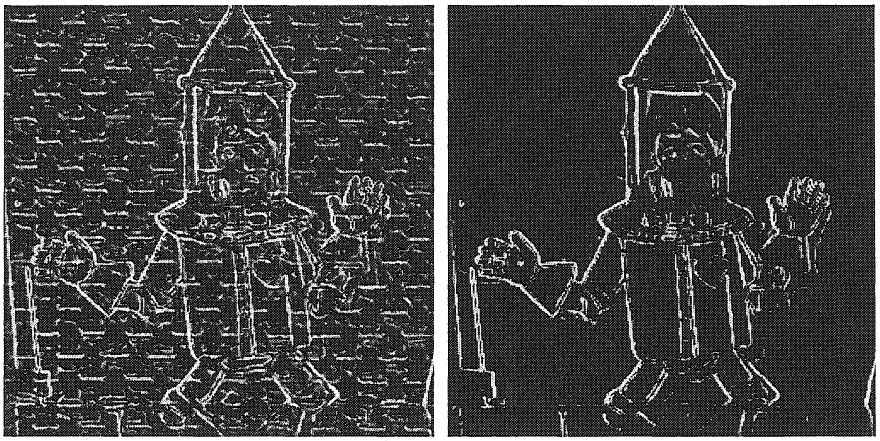

Fig. 8. Left - detected edges on the original image (see Fig 1), Right - detected edges on the curvelet reconstruct component.

where we remove both purely random noise, and some structural noise caused by the instruments used.

VII. Discussion

In this paper we have presented a novel method for separating an image into its texture and piece-wise smooth ingredients. Our method is based on the ability to represent these content types as sparse combinations of atoms of predetermined dictionaries. The proposed approach is a fusion of the Basis Pursuit algorithm and the Total-Variation regularization scheme, both merged in order to direct the solution towards a successful separation.

This paper offers a theoretical analysis of the separation idea with the Basis Pursuit algorithm, and shows that a perfect decomposition of image content could be found in principle. While the theoretical bounds obtained for a perfect decomposition are rather weak, they serve both as a starting point for future research, and as a motivating results for the practical sides of the work.

In going from the pure theoretic view to the implementation, we managed to extend the model to treat additive noise – essentially any content in the image that does not fit well with either texture or piece-wise-smooth contents. We also made a change in the problem formulation, departing from the Basis Pursuit, and getting closer to a Maximum A-Posterioi estimation method. The new formulation leads to smaller memory requirements, and more constraints can easily added on each of the components. This gives more flexibility for a given application. Simulation results show consistently promising results.

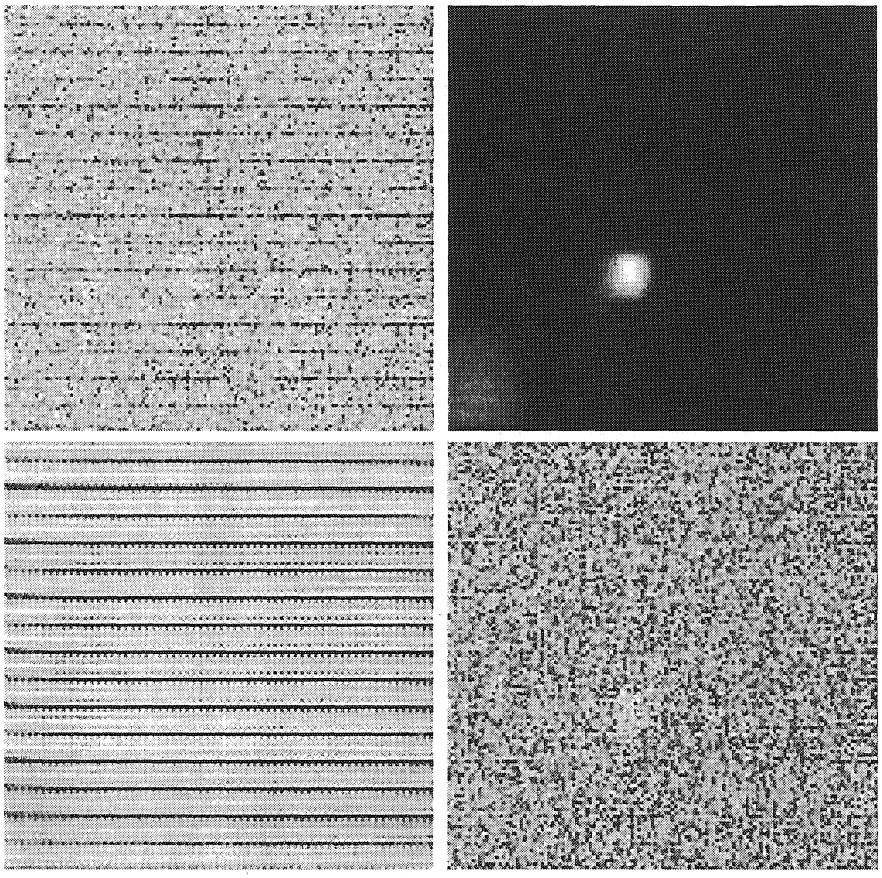

Fig. 9. Upper left: the original image, Upper left: the reconstructed wavelet component. Bottom left - the DCT reconstructed component, Bottom right - the residual, being the noise.

A natural comparison should be made between this work and the pioneering work recently published by Vese and Osher, on the same separation task. Appendix II attempts to present such a discussion. We feel that further work is required to fully understand the bridge between these two methodologies.

ACKNOWLEDGMENTS

The authors would like to thank Prof. Stanley Osher and Prof. Luminita Vese for helpful discussions, and for sharing their results to be presented in this paper.

APPENDIX I - THE BLOCK-COORDINATE-RELAXATION METHOD

In Section II-C we have seen an alternative formulation to the separation task, built on the assumption that the involved dictionaries are concatenations of unitary matrices. Thus, we need to minimize (7), given (after a simplification) as

$$\min_{\{\underline{\alpha}(k)\}_{k=1}^{L}} \quad \sum_{k=1}^{L} \|\underline{\alpha}(k)\|_1 + \lambda \left\| \underline{X} - \sum_{k=1}^{L} \mathbf{T}(k)\underline{\alpha}(k) \right\|_2^2 . \tag{I-1}$$

Note that we have discarded the TV part for the discussion given here. We also simply assume that the unknowns $\underline{\alpha}(k)$ contain both the texture and the piece-wise-smooth parts.

Minimizing such a penalty function was shown by Bruce, Sardy and Tseng [13] to be quite simple, as it is based on the shrinkage algorithm due to Donoho and Johnston [12]. In what follows we briefly describe this algorithm and its properties.

Property 1: *Referring to (I-1) as a function of $\{\underline{\alpha}(k)\}_{k_0}$, assuming all other unknowns as known, there is a closed-form solution for the optimal $\{\underline{\alpha}(k)\}_{k_0}$, given by*

$$\{\underline{\alpha}(k)\}_{k_0}^{opt} = \text{sign}(\underline{Z}) \cdot \left(|\underline{Z}| - \frac{1}{2\lambda} \right)_+ \tag{I-2}$$

for $\underline{Z} = \mathbf{T}(k_0)^H \left[\underline{X} - \sum_{k=1,\ k\neq K_0}^{L} \mathbf{T}(k)\underline{\alpha}(k) \right]$.

Proof: Rewriting (I-1) assuming that $\{\underline{\alpha}(k)\}_{k_0}$ are known, we have

$$\min_{\underline{\alpha}(k_0)} \quad \|\underline{\alpha}(k_0)\|_1 + \lambda \|\underline{Z} - \mathbf{T}(k_0)\underline{\alpha}(k_0)\|_2^2 . \tag{I-3}$$

Due to the fact that $\mathbf{T}(k_0)$ is unitary and the fact that the ℓ^2 norm is unitary invariant we can rewrite this penalty term as

$$\min_{\underline{\alpha}(k_0)} \quad \|\underline{\alpha}(k_0)\|_1 + \lambda \left\| \mathbf{T}(k_0)^H \underline{Z} - \underline{\alpha}(k_0) \right\|_2^2 , \tag{I-4}$$

which in turn, can be written as

$$\min_{\alpha(1),\alpha(2),\ldots,\alpha(N)} \quad \sum_{n=1}^{N} \left(|\alpha(n)| + \lambda [\alpha(n) - z_t(n)]^2 \right) . \tag{I-5}$$

This function is a sum of N (the dimension of $\underline{\alpha}(k_0)$) scalar and independent convex optimization problems. The term $z_t(n)$ represents the n^{th} entry of the inverse transform ($\mathbf{T}(k_0)$) of the vector $\underline{Z}$. The solution for this problem is given by the shrinkage operator mentioned above [12]. □

This property is the source of the simple numerical scheme of the Block-Coordinate-Relaxation Method. The idea is to sweep through the vectors $\underline{\alpha}(k)$ one at a time repeatedly, fixing all others, and solving for each.

Property 2: *Sweeping sequentially through k and updating $\underline{\alpha}(k)$ as in Property 1, the Block-Coordinate-Relaxation Method is guaranteed to converge to the optimal solution of (I-1).*

Proof: The proof is given in [13], along with practical implementation ideas. □

Appendix II - Relation to the Variational Method of Vese-Osher

Whereas piece-wise smooth images u are assumed to belong to the Bounded-Variation (BV) family of functions $u \in BV(\mathcal{R}^2)$, texture is known to behave differently. A different approach has recently been proposed for separating the texture v from the signal f $(= u + v)$ [2], based on a model proposed by Meyer [1]. This model suggests that a texture image v is to belong to a different family of functions denoted as $v \in BV^*(\mathcal{R}^2)$. This notation implies the existence of two functions $g_1, g_2 \in L^\infty(\mathcal{R}^2)$ such that $v(x,y) = \partial_x g_1(x,y) + \partial_y g_2(x,y)$. The BV^* norm is defined using the two functions g_1, g_2 as $\|v\|_{BV^*} = \|(|g_1(x)|^2 + |g_2(x)|^2)^{\frac{1}{2}}\|_\infty$.

Based on this model, a variational minimization problem was set by Vese and Osher to recover u, g_1, g_2 from a given mixture f. This approach essentially searches for the solution of

$$\inf_{(u,\ g_1,\ g_2)} \left[\|u\|_{BV} + \lambda\|v\|_{BV^*} + \lambda\|f - u + v\|_2^2\right] \tag{II-1}$$

A numerical algorithm to solve this problem is described in [2], with encouraging simulation results.

Let us return to our method and draw attention to several similar features between the two proposed approaches for separation of texture from a piece-wise smooth content. We refer to our formulation in (14) with the choice $\gamma = 0$,

$$\min_{\{\underline{X}_t,\ \underline{X}_n\}} \|\mathbf{T}_n^+ \underline{X}_n\|_1 + \left\|\mathbf{T}_t^+ \underline{X}_t\right\|_1 + \lambda \|\underline{X} - \underline{X}_t - \underline{X}_n\|_2^2 . \tag{II-2}$$

We have mentioned earlier that there is an equivalence between TV regularization and the soft thresholding using the undecimated Haar transform with a single scale [24]. It is also well known that the image obtained by soft thresholding is the solution of the minimization of the ℓ^1 norm. Therefore, we can write that $\|u\|_{BV} = \|\mathcal{H}u\|_1$ where $\mathcal{H}$ is the undecimated Haar transform (i.e. $\mathcal{H} = T_n^+$ in our original notations). Thus there is a similarity between the effects of the first terms in both (II-1) and (II-2).

Furthermore, we may argue that images with sparse representations in the DCT domain (local with varying block sizes and block overlap) present strong oscillations and therefore could be considered as textures, belonging to the Banach space $BV^*(\mathcal{R}^2)$. This suggests that $\|v\|_{BV^*}$ could also be written using a ℓ^1 norm as $\|\mathcal{D}v\|_1$ where $\mathcal{D}$ is the DCT transform (i.e. $\mathcal{D} = T_t^+$ in our notations). Now we get a similarity between the second terms on the two optimization problems posed in (II-1) and (II-2).

Since the third expression is exactly the same in (II-1) and (II-2), we see a close relation between our model and the one proposed by Meyer as adopted and used by Vese and Osher. However, there are also differences that should be mentioned.

In our implementation we do not use the undecimated Haar with just one resolution, but rather use the complete pyramid. This implies that a multi-scale TV is actually employed. We have also seen that in some cases, we replace the Haar with a more effective transform such as the the curvelet. Several reasons justify such a change. Among them is the fact that curvelet succeeds in distinguishing edges in noise much better than Haar wavelets. Moreover, Haar does not represent well edges, leading to a transfer of faint edges to the texture component, which may explain the results in Figure 6. Another important difference is the way the texture is modelled. Our method does not search for the implicit g_1, g_2 supposed to be the origin of the texture, but rather searches directly the texture part by an alternative and simpler model based on the local DCT.

Finally, we should note that the methodology presented in the paper is not limited to the separation of texture and cartoon in an image. Here we concentrated on the basic idea of separation of signals to different content types, leaning on the idea that each of the ingredients have a sparse representation with a proper choice of dictionaries. This may lead to other applications, and different implementations. We leave this for future research.

REFERENCES

[1] Y. Meyer, "Oscillating patterns in image processing and non linear evolution equations," *University Lecture Series, AMS* **22**, 2002.

[2] L. Vese and S. Osher, "Modeling textures with total variation minimization and oscillating patterns in image processing," *Journal of Scientific Computing* , 2003. in press.

[3] 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.

[4] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation noise removal algorithm," *Physica D* **60**, pp. 259–268, 1992.

[5] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *to appear in IEEE Trans. on Image Processing* , 2003.

[6] S. Chen, D. Donoho, and M. Saunder, "Atomic decomposition by basis pursuit," *SIAM Journal on Scientific Computing* **20**, pp. 33–61, 1998.

[7] J.-L. Starck, E. Candes, and D. Donoho, "Astronomical image representation by the curvelet tansform," *Astronomy and Astrophysics* **398**, pp. 785–800, 2003.

[8] D. Donoho and X. Huo, "Uncertainty Principles and Ideal Atomic Decomposition," *IEEE Transactions on Information Theory* **47**(7), pp. 2845–2862, 2001.

[9] D. L. Donoho and M. Elad, "Maximal sparsity representation via l_1 minimization," *the Proc. Nat. Aca. Sci.* **100**, pp. 2197–2202, 2003.

[10] 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.

[11] R. Gribonval and M. Nielsen, "Some remarks on nonlinear approximation with schauder bases," *East J. on Approx.* **7**(2), pp. 267–285, 2001.

[12] D. Donoho and I. Johnstone, "Ideal spatial adaptation via wavelet shrinkage," *Biometrika* **81**, pp. 425–455, 1994.

[13] A. Bruce, S. Sardy, and P. Tseng, "Block coordinate relaxation methods for nonparametric signal de-noising," *Proceedings of the SPIE - The International Society for Optical Engineering* **3391**, pp. 75–86, 1998.

[14] M. Antonini, M. Barlaud, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," *IEEE Transactions on Image Processing* **1**, pp. 205–220, 1992.

[15] J. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Transactions on Signal Processing* **41**, pp. 3445–3462, 1993.

[16] 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.

[17] 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.

[18] J.-L. Starck, F. Murtagh, and A. Bijaoui, *Image Processing and Data Analysis: The Multiscale Approach*, Cambridge University Press, 1998.

[19] J.-L. Starck and F. Murtagh, *Astronomical Image and Data Analysis*, Springer-Verlag, 2002.

[20] E. J. Candès, "Harmonic analysis of neural netwoks," *Applied and Computational Harmonic Analysis* **6**, pp. 197–218, 1999.

[21] 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.

[22] 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.

[23] 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.

[24] G. Steidl, J. Weickert, T. Brox, P. Mrzek, and M. Welk, "On the equivalence of soft wavelet shrinkage, total variation diffusion, total variation regularization, and sides," Tech. Rep. 26, Department of Mathematics, University of Bremen, Germany, 2003.

[25] 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.

[26] M. Vetterli, "Wavelets, approximation, and compression," *IEEE Signal Processing Magazine* **18**(5), pp. 59–73, 2001.