Available online at www.sciencedirect.com
SCIENCE DIRECT®
Appl. Comput. Harmon. Anal. 19 (2005) 340–358
Applied and Computational Harmonic Analysis
www.elsevier.com/locate/acha

Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)

M. Elad [a,*], J.-L. Starck [b], P. Querre [b], D.L. Donoho [c]

[a] *Computer Science Department, The Technion—Israel Institute of Technology, Haifa 32000, Israel*
[b] *CEA-Saclay, DAPNIA/SEDI-SAP, Service d'Astrophysique, F-91191 Gif sur Yvette, France*
[c] *Department of Statistics, Stanford University, Sequoia Hall, Stanford, CA 94305, USA*

Received 27 October 2004; revised 6 March 2005; accepted 9 March 2005

Available online 15 August 2005

Communicated by Charles K. Chui

Abstract

This paper describes a novel inpainting algorithm that is capable of filling in holes in overlapping texture and cartoon image layers. This algorithm is a direct extension of a recently developed sparse-representation-based image decomposition method called MCA (morphological component analysis), designed for the separation of linearly combined texture and cartoon layers in a given image (see [J.-L. Starck, M. Elad, D.L. Donoho, Image decomposition via the combination of sparse representations and a variational approach, IEEE Trans. Image Process. (2004), in press] and [J.-L. Starck, M. Elad, D.L. Donoho, Redundant multiscale transforms and their application for morphological component analysis, Adv. Imag. Electron Phys. (2004) 132]). In this extension, missing pixels fit naturally into the separation framework, producing separate layers as a by-product of the inpainting process. As opposed to the inpainting system proposed by Bertalmio et al., where image decomposition and filling-in stages were separated as two blocks in an overall system, the new approach considers separation, hole-filling, and denoising as one unified task. We demonstrate the performance of the new approach via several examples.

Keywords: Basis pursuit; Total variation; Sparse representation; Cartoon; Texture; Inpainting

* Corresponding author.
E-mail addresses: elad@cs.technion.ac.il (M. Elad), jstarck@cea.fr (J.-L. Starck), donoho@stat.stanford.edu (D.L. Donoho).

1063-5203/$ – see front matter
doi:10.1016/j.acha.2005.03.005

1. Introduction

Filling-in 'holes' in images is an interesting and important inverse problem with many applications. Removal of scratches in old photos, removal of overlaid text or graphics, filling-in missing blocks in unreliably transmitted images, scaling-up images, predicting values in images for better compression, and more, are all manifestations of the above problem. In recent years this topic attracted much interest, and many contributions have been proposed for the solution of this interpolation task. Common to these many techniques is the understanding that classic interpolation methods (such as polynomial-based approaches) are not satisfying; indeed nonlinear strategies and local adaptivity seem crucial.

Among the numerous approaches to fill in holes in images, variational methods are very attractive; these were pioneered by Guillermo Sapiro and his collaborators [6,20,21], and followed by Chan and Shen [7]. These techniques were coined *Inpainting* as a reminder of the recovery process museums experts do for old and deteriorating artwork. In their work, Sapiro et al. motivate the filling-in algorithms by geometrical considerations: one should fill in by a smooth continuation of isophotes. This principle leads to one or another nonlinear partial differential equation (PDE) model, propagating information from the boundaries of the holes while guaranteeing smoothness of some sort. In a series of publications, the geometric principle has been implemented through several different PDEs, aiming to get the most convincing outcome.

The variational approach has been shown to perform well on piecewise smooth images. Here and below we call such images *cartoons*, and think of them as carrying only geometric information. Real images also contain textured regions, and variational methods generally fail in such settings. On the other hand, local statistical analysis and prediction have been shown to perform well at filling in texture content [3,13,29].

Of course real images contain both geometry and texture; they demand approaches that work for images containing both cartoon and texture layers. In addition, approaches based on image segmentation—labeling each pixel as either cartoon or texture—are to be avoided, since some areas in the image contain contributions from both layers. Instead, a method of additively decomposing the image into layers would be preferred, allowing a combination of layer-specific methods for filling in.

This motivated the approach in [2]. Building on the image decomposition method by Vese, Osher, and others [1,28], the image was separated into cartoon and texture images. The inpainting was done separately in each layer, and the completed layers were superposed to form the output image. The layer decomposition, a central component in this approach, was built on variational grounds as well, extending the notion of total-variation [23], based on a recent model for texture images by Meyer [22]. An interesting feature of this overall system is that even if the image decomposition is not fully successful, the final inpainting results can be still quite good, since the expected failures are in areas where the assignment to cartoon/texture contents is mixed, where both inpainting techniques perform rather well.

In previous papers we presented an alternative approach to layer decomposition, optimizing the sparsity of each layer's representation [25,26]. The central idea is to use two adapted dictionaries, one adapted to represent textures, and the other to represent cartoons. The dictionaries are mutually incoherent; each leads to sparse representations for its intended content type, while yielding nonsparse representations on the other content type. These are amalgamated into one combined dictionary, and the basis-pursuit denoising (BPDN) algorithm [8] is relied upon for proper separation, as it seeks the combined sparsest solution, which should agree with the sparse representation of each layer separately. This algorithm was shown to perform well, and was further improved by imposing total-variation (TV) regularization as an

additional constraint. A nice feature to this algorithm is its ability to handle additive noise as a third content type, and separate the given image into three components, achieving denoising as a by-product.

Naturally, one could deploy such a separation technique in the block diagram strategy of [2], obtaining an alternative inpainting algorithm. However, separation-by-sparsity offers a fundamentally different strategic option, *integrated* inpainting. Indeed, in this paper we propose an inpainting algorithm capable of filling in holes in either texture or cartoon content, or any combinations thereof. This new algorithm extends the sparsity-seeking layer separation method of [25,26] mentioned above. In effect, we show that missing pixels fit naturally into the layer-separation framework. As a result, layer separation and denoising of the image are integral by-products of the inpainting process.

As opposed to the inpainting system proposed in [2], where the image decomposition and the filling-in stages were separated, our approach recombines the two ingredients in one. Our model is general and has several desirable features:

(1) The image is allowed to include additive white noise;
(2) The image is allowed to have missing pixels; and
(3) The image is assumed to be a sparse combination of atoms from the two dictionaries.

Whereas the two first features refer to the measurements of the problem, as manifested in the likelihood function, the last one plays the role of regularization, proposing a prior knowledge on the unknown image.

The inpainting method proposed in [18,19] is closely related to our technique, being also based on sparse representations. Our method seems to offer substantial advantages, including: (i) the use of general overcomplete representations which are better suited for typical image content; (ii) a global treatment of the image, rather than a local block-based analysis; (iii) a coherent modeling of the overall problem as an optimization, rather than the presentation of a numerical scheme; and, perhaps most important of all, (iv) the ability to treat overlapping texture and cartoon layers, due to our separation abilities. We will return to these issues in more depth after describing our algorithm in Section 3.

In the next section we briefly describe the image separation method as presented in [25,26]. In Section 3 we show how this should be extended to treat missing parts, and discuss the numerical algorithm that should be employed for the solution of the new optimization task posed. We describe some experimental results in Section 4 and conclude in Section 5.

2. Image decomposition using the MCA approach

Let the input image, containing N total pixels, be represented as a 1D vector of length N by lexicographic ordering. To model images $\underline{X}_t$ containing *only* texture, we assume that a matrix $\mathbf{T}_t \in \mathcal{M}^{N\times L}$ (where typically $L \gg N$) allows sparse decomposition, written informally as

$$\underline{X}_t = \mathbf{T}_t \underline{\alpha}_t, \quad \alpha_t \text{ is sparse.} \tag{1}$$

Here sparsity can be quantified by any of several different quasi-norms including the ℓ_0 norm $\|\alpha\|_0 = \#\{i\colon \alpha(i) \neq 0\}$ and ℓ_p-norms $\|\alpha\|_p = (\sum |\alpha(i)|^p)^{1/p}$ with $p < 1$, with small values of any of these indicating sparsity. Sparsity measured in ℓ_0 norm implies that the texture image can be a linear combination of relatively few columns from $\mathbf{T}_t$.

There are two more technical assumptions. First, localization: the representation matrix $\mathbf{T}_t$ is such that if the texture appears in parts of the image and is otherwise zero, the representation is still sparse, implying that this dictionary employs a multi-scale and local analysis of the image content. Second, incoherence: $\mathbf{T}_t$ should not be able to represent cartoon images sparsely. We require that when (1) is applied to images containing cartoon content, the resulting representations are nonsparse. Thus, the dictionary $\mathbf{T}_t$ plays a role of a discriminant between content types, preferring texture geometry.

Turn now to the geometric layer. Converse to the above, we assume there is a dictionary $\mathbf{T}_n$, such that a cartoon image $\underline{X}_n$ is sparsely represented by the above definition. We further assume that texture images are represented very nonsparsely by T_n, and also assume that the analysis applied by this dictionary is of multi-scale and local nature, enabling it to represent localized pieces of the desired content.

For an arbitrary image $\underline{X}$ containing both texture and piecewise smooth content (superposed or segmented), we propose to seek a sparse representations over the combined dictionary containing both $\mathbf{T}_t$ and $\mathbf{T}_n$. If we work with the ℓ_0 norm as a definition of sparsity, we need to solve

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

It would be very desirable to obtain the solution of this optimization task. Intuitively, it should lead to a successful separation of the image content, with $\mathbf{T}_t\underline{\alpha}_t$ containing the texture and $\mathbf{T}_n\underline{\alpha}_n$ containing the cartoon. This expectation relies on the assumptions made earlier about $\mathbf{T}_t$ and $\mathbf{T}_n$ being able to sparsely represent one content type while being highly noneffective in sparsifying the other.

While sensible as a general goal, the problem formulated in Eq. (2) is nonconvex and seemingly intractable. Its complexity grows exponentially with the number of columns in the overall dictionary. The basis pursuit (BP) method [8] suggests the replacement of the ℓ^0-norm with an ℓ^1-norm, thus leading to a tractable convex optimization problem, in fact being reducible to linear programming:

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

Interestingly, recent work has shown that, for certain dictionaries and for objects that have sufficiently sparse solutions, the BP approach can actually produce the sparsest of all representations [9,10,14,16].

If the image is noisy it cannot be cleanly decomposed into sparse texture and cartoon layers. We therefore propose a noise-cognizant version of BP

$$\left\{\underline{\alpha}_t^{\text{opt}}, \underline{\alpha}_n^{\text{opt}}\right\} = \arg\min_{\{\underline{\alpha}_t, \underline{\alpha}_n\}} \|\underline{\alpha}_t\|_1 + \|\underline{\alpha}_n\|_1 \quad \text{subject to:} \quad \|\underline{X} - \mathbf{T}_t\underline{\alpha}_t - \mathbf{T}_n\underline{\alpha}_n\|_2 \leqslant \varepsilon. \tag{4}$$

This way, the decomposition of the image is only approximate, leaving some error to be absorbed by content that is not represented well by both dictionaries. The parameter ε stands for the noise level in the image $\underline{X}$. Alternatively, the constrained optimization in (4) can be replaced by an unconstrained penalized optimization. Both noise-cognizant approaches have been analyzed theoretically, providing conditions for a sparse representation to be recovered accurately [11,27].

Also useful in the context of sparsity-based separation is the imposition of a total variation (TV) penalty [23]. This works particularly well in recovering piecewise smooth objects with pronounced edges—i.e., when applied to the cartoon layer. It is most conveniently imposed as a penalty in an unconstrained optimization:

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

Here the total variation of an image I, $\mathrm{TV}(I)$ is essentially the ℓ^1 norm of the gradient. Penalizing with TV forces the image $\mathbf{T}_n\underline{\alpha}_n$ to have a sparser gradient, and hence to be closer to a piecewise smooth image. More on TV and how to use it can be found in [23].

As to the actual choice of $\mathbf{T}_t$ and $\mathbf{T}_n$, our approach in this work is to choose known transforms. For texture content we may use transforms such as local DCT, Gabor or wavelet packets (oscillatory ones to fit texture behavior). For the cartoon content we can use wavelet, curvelet, ridgelets, contourlets, and there are several more options. In both cases, the proper choice of dictionaries depends on the actual content of the image to be treated. At this writing, the best choice of transform will depend on the user's experience; choices made may vary from one image to another. For numerical reasons, we restrict our choices to dictionaries $\mathbf{T}_t$ and $\mathbf{T}_n$ that have fast forward, inverse, and adjoint transforms. More details on these issues can be found in [25,26].

Figure 1 illustrates the layer separation result for the `Barbara` image, as obtained by the above described algorithm. Many more such results are given in [25,26]. This separation was obtained using the curvelet transform with five resolution levels at $\mathbf{T}_n$, and 50% overlapping discrete cosine transform with a block size 32×32 as $\mathbf{T}_t$.

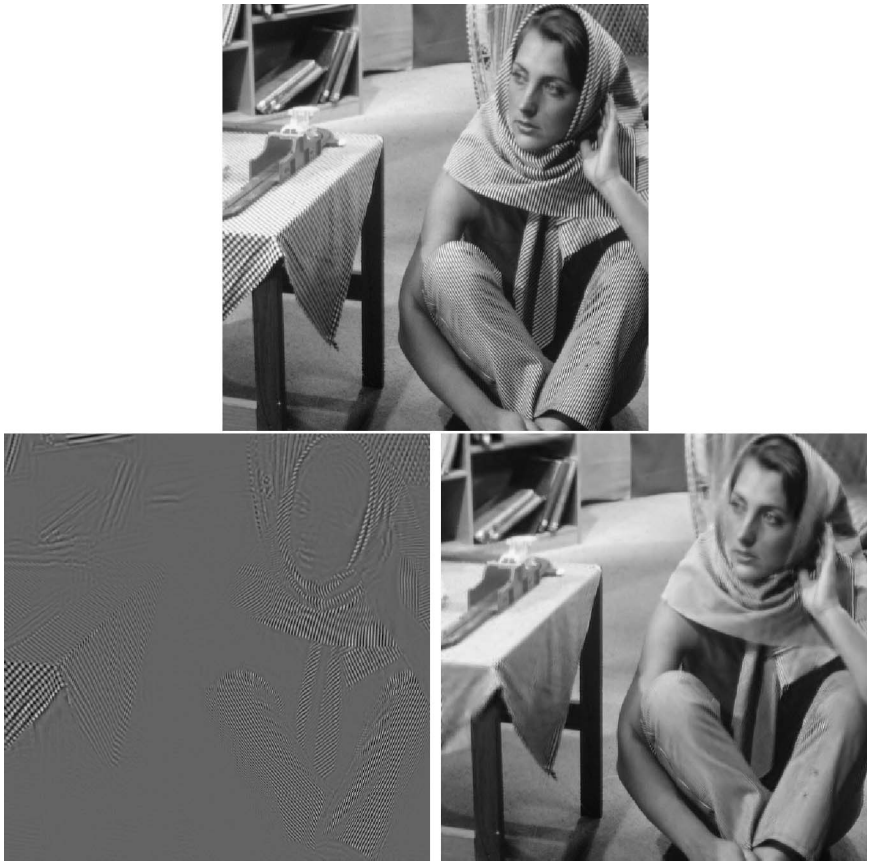

Fig. 1. The original `Barbara` image (top), the separated texture (bottom left), and the separated cartoon (bottom right).

3. Image inpainting using MCA

Assume that the missing pixels are indicated by a diagonal 'mask' matrix $\mathbf{M} \in \mathcal{M}^{N\times N}$. The main diagonal of $\mathbf{M}$ encodes the pixel status, namely '1' for an existing pixel and '0' for a missing one. Thus, in the model (5) we can incorporate this mask by

$$\left\{\underline{\alpha}_t^{\text{opt}}, \underline{\alpha}_n^{\text{opt}}\right\} = \arg\min_{\{\underline{\alpha}_t,\underline{\alpha}_n\}} \|\underline{\alpha}_t\|_1 + \|\underline{\alpha}_n\|_1 + \lambda\left\|\mathbf{M}(\underline{X} - \mathbf{T}_t\underline{\alpha}_t - \mathbf{T}_n\underline{\alpha}_n)\right\|_2^2 + \gamma\,\text{TV}\{\mathbf{T}_n\underline{\alpha}_n\}. \tag{6}$$

This way, we desire an approximate decomposition of the input image $\underline{X}$ to texture and cartoon parts, $\mathbf{T}_t\underline{\alpha}_t$ and $\mathbf{T}_t\underline{\alpha}_t$, respectively, and the fidelity of the representation is measured with respect to the existing measurements only, disregarding missing pixels. The idea is that once $\mathbf{T}_t\underline{\alpha}_t$ and $\mathbf{T}_n\underline{\alpha}_n$ are recovered, those represent entire images, where holes are filled in by the two dictionaries' basis functions.

Interestingly, if we simplify the above model by using a single unitary transform $\mathbf{T}$, the model becomes

$$\underline{\hat{X}} = \mathbf{T}\cdot\underline{\alpha}^{\text{opt}} = \mathbf{T}\cdot\arg\min_{\underline{\alpha}}\left\{\|\underline{\alpha}\|_1 + \lambda\left\|\mathbf{M}(\underline{X} - \mathbf{T}\underline{\alpha})\right\|_2^2\right\} = \arg\min_{\underline{Z}}\left\{\left\|\mathbf{T}^H\underline{Z}\right\|_1 + \lambda\left\|\mathbf{M}(\underline{X} - \underline{Z})\right\|_2^2\right\}, \tag{7}$$

and this is essentially the model underlying the method presented in [18,19]. In his work, Guleryuz describes an iterated numerical scheme that effectively minimizes the above function. While the above model leads to a simpler inpainting method, it is a weaker version of the one proposed here in Eq. (6) for several reasons:

- The model in (6) uses general overcomplete representations. This allows to better match natural image content by choosing the transform to strengthen the sparsity assumption, which is at the heart of the two methods.
- Using a pair of dictionaries, the algorithm can cope with the combination of linearly combined texture and cartoon content overlapped in the image.
- The total-variation penalty in (6) suppresses the typical ringing artifacts encountered in using linear transforms. This can be crucial near sharp edges, where ringing artifacts are strongly visible.
- While the above models (both) consider the image as a whole, the approach taken in [18,19] is local and block-based. Thus, multi-scale relations that exist in the image and could be exploited are overlooked. Still, the formulation of (7) allows $\mathbf{T}$ to be chosen as an orthonormal multi-scale transform that operates on the entire image (e.g., wavelet), and then improved results could be obtained.

On the other hand, we should mention that Guleryuz's block-based approach is much simpler than the one proposed here, and so has a strong appeal despite the above drawbacks.

Returning to the model in (6), instead of solving this optimization problem directly and finding two representation vectors $\{\underline{\alpha}_t^{\text{opt}}, \underline{\alpha}_n^{\text{opt}}\}$, let us reformulate this 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 lower-dimensional vectors—representation vectors are much longer than the image they represent for overcomplete dictionaries as the ones we use here.

Defining $\underline{X}_t = \mathbf{T}_t\underline{\alpha}_t$, 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 $\mathbf{T}_t$. A similar structure exists for $\underline{X}_n = \mathbf{T}_n\underline{\alpha}_n$, with a residual vector $\underline{r}_n$ in the null-space of $\mathbf{T}_n$. Put these back into (6) we obtain

$$\left\{\underline{X}_t^{\text{opt}}, \underline{X}_n^{\text{opt}}\right\} = \underset{\{\underline{X}_t, \underline{X}_n, \underline{r}_t, \underline{r}_n\}}{\arg\min} \|\mathbf{T}_t^+ \underline{X}_t + \underline{r}_t\|_1 + \|\mathbf{T}_n^+ \underline{X}_n + \underline{r}_n\|_1 + \lambda \left\|\mathbf{M}(\underline{X} - \underline{X}_t - \underline{X}_n)\right\|_2^2 + \gamma \text{TV}\{\underline{X}_n\}$$
$$\text{subject to:} \quad \mathbf{T}_t \underline{r}_t = 0, \ \mathbf{T}_n \underline{r}_n = 0. \tag{8}$$

The terms $\mathbf{T}_t^+ \underline{X}_t$ and $\mathbf{T}_n^+ \underline{X}_n$ are overcomplete linear transforms of the images $\underline{X}_t$ and $\underline{X}_n$, respectively. For tight frames, these are equivalent to the multiplication by the adjoint of the original dictionaries $\mathbf{T}_t$ and $\mathbf{T}_n$.

In the spirit of the simplification done in [26], we assume $\underline{r}_t = \underline{r}_n = 0$. Thus we find a suboptimal solution to the problem posed in (8). The resulting minimization task becomes

$$\min_{\{\underline{X}_t, \underline{X}_n\}} \|\mathbf{T}_t^+ \underline{X}_t\|_1 + \|\mathbf{T}_n^+ \underline{X}_n\|_1 + \lambda \left\|\mathbf{M}(\underline{X} - \underline{X}_t - \underline{X}_n)\right\|_2^2 + \gamma \text{TV}\{\underline{X}_n\}. \tag{9}$$

There are several ways to justify this choice of $\underline{r}_t = \underline{r}_n = 0$ made above:

- The function minimized in (9) could be perceived as a simplified upper-bound function to the one in (8). Indeed, per every choice of the pair $\{\underline{X}_t, \underline{X}_n\}$, the value of the function in (9) is higher than the one obtained in (8) when optimized with respect to $\underline{r}_t$ and $\underline{r}_n$. Replacing the original objective with an upper bound of it makes sense here, since the new formulation is much easier to solve, as it's unknowns are of substantially smaller dimension. A crucial question that remains is how far could the optimal solutions $\{\underline{X}_t, \underline{X}_n\}$ be, when passing from (8) to (9). While we do not explicitly answer this question here, we show experimentally that the solutions obtained from (9) are of worth. Also, the next explanations shed some light on the fact that the two are expected to be quite close in general.
- Interestingly, it is relatively easy to see that if the dictionaries $\mathbf{T}_t$ and $\mathbf{T}_n$ are square and nonsingular matrices (leading to a complete, rather than overcomplete, representations), then (8) and (9) are equivalent, implying that the choice $\underline{r}_t = \underline{r}_n = 0$ loses nothing. Similarly, if the ℓ^1-norms in (8) and (9) are replaced with ℓ^2 norms, the two formulations are again equivalent, regardless of the dictionary sizes. When we depart from those two simplified cases and consider ℓ^1-norm and overcomplete representations, we know that the two are different, but expect this difference to be relatively small. The reason is that we are interested in the images $\underline{X}_t$ and $\underline{X}_n$, and not their representations. While $\underline{r}_t$ and $\underline{r}_n$ may be different from zero, their effect on the final outcome is reduced as we multiply by the dictionaries $\mathbf{T}_t$ and $\mathbf{T}_n$ to obtain the separated images.
- The formulation in (9) has a solid Bayesian interpenetration, independent of the source formulation in (8). The new problem format has maximum a posteriori probability structure, with a log-likelihood term being $\|\mathbf{M}(\underline{X} - \underline{X}_t - \underline{X}_n)\|_2^2$, and prior terms for the cartoon and the texture parts. The priors are analysis-based, with a promotion of sparsity of the filtered images, $\mathbf{T}_t^+ \underline{X}_t$ and $\mathbf{T}_n^+ \underline{X}_n$. In addition, spatial piece-wise smoothness in the cartoon image is promoted by the TV term. Note, however, that this change implies a change in the sparsity assumption underlying our method.

The algorithm we use to solve this optimization problem is based on the block-coordinate-relaxation method with some required changes due to the nonunitary transforms involved, and the additional TV term [4,25]. Also, the mask matrix $\mathbf{M}$ should be taken into consideration. The MCA algorithm is given below:

1. Initialization:
   - Choose parameters: $L_{\max}$—threshold factor, N—number of iterations, and the parameters λ, γ.
   - Initialize $\underline{X}_n = \underline{X}$ and $\underline{X}_t = 0$.
   - Set $\delta = \lambda \cdot L_{\max}$.
2. Perform N times:
   Part A—Update $\underline{X}_n$ with $\underline{X}_t$ fixed:
   - Calculate the residual $\underline{R} = \mathbf{M}(\underline{X} - \underline{X}_t - \underline{X}_n)$.
   - Calculate the curvelet transform of $\underline{X}_n + \underline{R}$: $\underline{\alpha}_n = \mathbf{T}_n^+(\underline{X}_n + \underline{R})$.
   - 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 $\underline{X}_t$ with $\underline{X}_n$ fixed:
   - Calculate the residual $\underline{R} = \mathbf{M}(\underline{X} - \underline{X}_t - \underline{X}_n)$.
   - Calculate the local-DCT transform of $\underline{X}_t + \underline{R}$: $\underline{\alpha}_t = \mathbf{T}_t^+(\underline{X}_t + \underline{R})$.
   - 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 penalization:
   - Apply TV correction by

$$\underline{X}_n = \underline{X}_n - \mu \frac{\partial \mathrm{TV}\{\underline{X}_n\}}{\partial \underline{X}_n} = \underline{X}_n - \mu \nabla \cdot \left(\frac{\nabla \underline{X}_n}{|\nabla \underline{X}_n|} \right)$$

   (see [23] for more details about this derivative). 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.[a]
3. Update the threshold by $\delta = \delta - \lambda/N$.
4. If $\delta > \lambda$, return to Step 2. Else, finish.

[a] This is were γ influences the algorithm's outcome.

The numerical algorithm for minimizing (9).[1]

As can be seen, by replacing the mask matrix by the identity operator we obtain the very same algorithm as proposed in [25,26] for the task of image decomposition. Thus, this algorithm is a simple modification of the separation one proposed earlier.

The rationale behind the way the mask is taken into account here is the following: suppose that after several rounds we have a rough approximation of $\underline{X}_t$ and $\underline{X}_n$. In order to update $\underline{X}_n$ we assume that $\underline{X}_t$ is fixed and compute the residual image $\underline{R} = \mathbf{M}(\underline{X} - \underline{X}_t - \underline{X}_n)$. In existing pixels (where the mask value is '1') this residual has a content that can be attributed to texture, cartoon, and/or noise content. On the missing pixels (where the mask is '0') the residual value is forced to zero by the multiplication with the mask. Thus, the image $\underline{R} + \underline{X}_n$ does not contain holes. An analysis of this image—transforming it to curvelet coefficients, nulling small entries, and reconstructing it—is able to absorb some of the cartoon

[1] Notice that in turning from the formulation (9) to the algorithm described here, we have changed the role of λ. In the algorithm is it used as a weight that multiplies the ℓ^1-norm terms. This change was made to better fit the soft-thresholding description, and it has no impact on the way the problem formulation acts.

content that exists in $\underline{R}$. This way the updated $\underline{X}_n$ takes some of the cartoon content that exists in the residual, and the new residual image energy becomes smaller.

In the language of numerical optimization, the above algorithm could be described as a block-coordinate descent algorithm, where one image (say $\underline{X}_t$) is fixed while the other (say $\underline{X}_n$) is updated, and vice-versa. Within each such update stage there are two parts (disregarding the TV treatment): The first minimizes the penalty $\|\mathbf{M}(\underline{X} - \underline{X}_t - \underline{X}_n)\|_2^2$ by assigning $\underline{X}_n^{\text{new}} \leftarrow \underline{X}_n^{\text{old}} + \mathbf{M}(\underline{X} - \underline{X}_t - \underline{X}_n^{\text{old}})$. This causes this penalty to be nulled,

$$\left\|\mathbf{M}\left(\underline{X} - \underline{X}_t - \underline{X}_n^{\text{new}}\right)\right\|_2^2 = \left\|\mathbf{M}\left(\underline{X} - \underline{X}_t - \underline{X}_n^{\text{old}} - \mathbf{M}\left(\underline{X} - \underline{X}_t - \underline{X}_n^{\text{old}}\right)\right)\right\|_2^2 = 0,$$

since $\mathbf{M}^2 = \mathbf{M}$. The second part decreases the penalty $\|\mathbf{T}_n^+ \underline{X}_n\|_1$ while maintaining proximity between the outcome and the updated $\underline{X}_n$. This is achieved by soft-thresholding. Merged together, these two steps cause a decrease in the overall penalty as a function of $\underline{X}_n$, if the thresholding is moderate enough. The same applies to the update of $\underline{X}_t$.

Why should this work?

In this section we started from the desire to fill-in missing pixels in an image, and concluded with the claim that a proper way to achieve this goal is the solution of the minimization problem posed in (9). In the path from the objective to its solution, we have used various assumptions and conjectures, without which the overall inpainting process is doomed to fail. Let us list those assumptions and show how they build the eventual inpainting algorithm:

- *Sparse and overcomplete model assumption*: We assume that an image could be modeled as a sparse linear combination of atom images. Furthermore, we assume that general images could be described as such sparse compositions over two dictionaries, one responsible for the texture and the other for the cartoon content. These assumptions are at the roots of this work. We cannot justify such claims theoretically, and in fact, it is unclear whether this is at all possible. Instead, we can rely on recent years' results on the role of sparsity and over-completeness in signal and image processing, with respect to the wavelet transform, and its advanced versions such as the curvelet [24], and more. We can also pose these as assumptions we build upon, and see whether the results agree.
  An additional assumption here is the existence of such two dictionaries for the cartoon and the texture, and our ability to get them. In this work we have chosen specific known transforms, exploiting their known tendency to sparse compositions. Further work is needed to replace this stage by a training method that evaluates the dictionaries from examples. As above, the results of the MCA algorithm will either support such assumptions or stand as a contradiction.
- *Sparsity can be handled with* ℓ^1: Considering the above assumptions as true, we need to find the sparsest representation that fits the data. This process, as posed in (2), is known as *atomic decomposition*. Since this is a complex combinatorial problem, it has been relaxed with an ℓ^1 formulation. Results gathered in the past four years support such a relaxation, with a reasonable guarantee of successful recovery of the desired representation, if it is sparse enough to begin with. Representative work along these lines can be found in [5,10–12,15,17,27], where both the exact and the noisy cases are considered.

- *Treatment of missing samples*: Missing pixels in the image are handled by the weight matrix **M** introduced in Eq. (4). Considering a simplified version of (4), without the TV term and with an exact decomposition rather than an inaccurate one, we get

$$\underline{\alpha}^{\text{opt}} = \arg\min_{\underline{\alpha}} \|\underline{\alpha}\|_1 \quad \text{subject to:} \quad \mathbf{M}\underline{X} = \mathbf{MT}\underline{\alpha}. \tag{10}$$

The core question remains: assuming that there is indeed a sparse $\underline{\alpha}$ such that $\underline{X} = \mathbf{T}\underline{\alpha}$, will the formulation in (10) be successful in recovering it? How does this depend on the sparsity of $\underline{\alpha}$ and the amount of missing pixels marked in **M**? Clearly, if $\underline{\alpha}$ is recovered successfully, then by multiplication by the dictionary we get the filling-in effect we desire.

These questions and their generalization to the approximate representation case (where the constraint $\mathbf{M}\underline{X} = \mathbf{MT}\underline{\alpha}$ is replaced by a penalty $\|\mathbf{M}\underline{X} - \mathbf{MT}\underline{\alpha}\|_2^2$) can be analyzed. Putting things into perspective, the constraint in (10) essentially states $\underline{\tilde{X}} = \tilde{\mathbf{T}}\underline{\alpha}$, where we define $\underline{\tilde{X}} = \mathbf{M}\underline{X}$ and $\tilde{\mathbf{T}} = \mathbf{MT}$. This linear set of equations has a subset of the rows in the original $\underline{X} = \mathbf{T}\underline{\alpha}$. Thus, previous analysis in the study of uniqueness of sparse representations and equivalence when using ℓ^1 are all applicable. Thus, a study of the decay of the mutual incoherence as a function of the rows removed could be of help here (see [5,10,12,17]). We will not show this study here (we are currently working on this problem and we hope to show some theoretical results soon). Instead we demonstrate the expected behavior of the above problem via a synthetic experiment.

We use a maximally incoherent two random and orthonormal dictionaries $\mathbf{T} = [\mathbf{\Phi}, \mathbf{\Psi}]$ of size 64×128 [5]. We use a random and sparse representation $\underline{\alpha}$ with $n \in [1, 10]$ nonzero entries in random

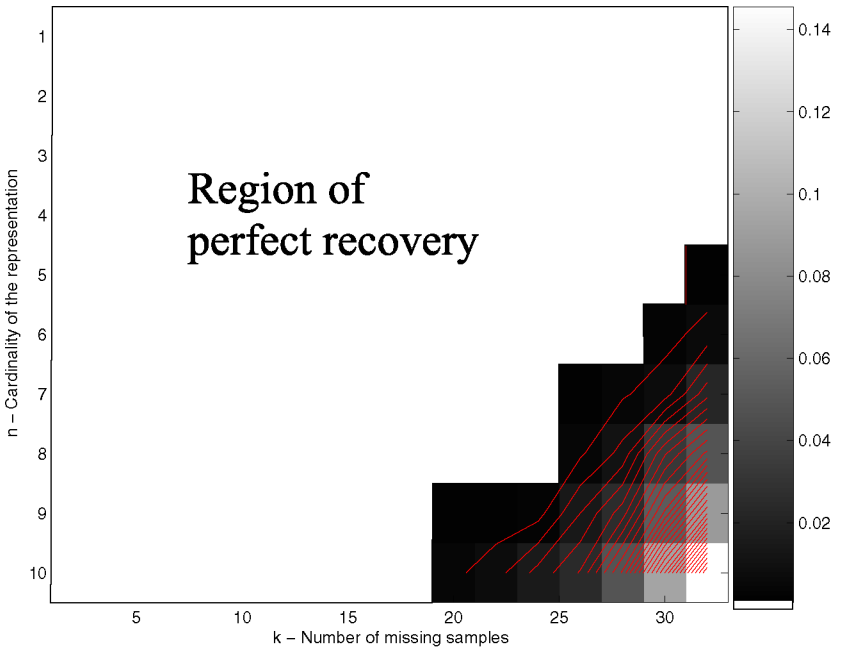

Fig. 2. A synthetic experiment showing the relative error in the recovery of missing samples as a function of their number k and the original representation's cardinality n. The overlaid curves are the contour plot of the same data, showing a growth toward the bottom right corner. The masked area corresponds to a perfect recovery.

locations and with zero mean Gaussian i.i.d. entries, and compute $\underline{X} = \mathbf{T}\underline{\alpha}$. We generate a random missing pattern of k samples with $k \in [0, 32]$ missing samples, and solve (10). Finally, we compare the obtained result $\mathbf{T}\underline{\alpha}^{\text{opt}}$ to the original signal $\underline{X}$, using the following formula: $\|\underline{X} - \underline{X}_{\text{opt}}\|_2^2/(\|\underline{X}\|_2^2 - \|\underline{X}_{\text{opt}}\|_2^2)$. Since the noncanceled entries in $\underline{X}$ are unaffected and are the same as those in $\underline{X}_{\text{opt}}$, the denominator in the above measure gives the energy of the missing values. Thus, the error obtained is a relative error, being 1 for a simple interpolation that fills the missing values by zeros.
Figure 2 presents this relative error as a function of k (the number of missing samples) and n (the original number of nonzeros in the representation). Per every (k, n) pair a set of 1000 experiments were performed and averaged. As can be seen, for sparse enough representations and with small enough number of missing samples, the process yields perfect recovery (the top left masked area). The results deteriorate as the two grow, but as can be seen, even for $\|\underline{\alpha}\|_0 = 10$ and 32 missing samples, the relative error is still reasonable, being approximately 0.14. As was said above, a theoretical analysis of this behavior is currently under study.

- *From synthesis to analysis formulation*: The last brick in the wall of assumptions made to solve the inpainting problem, is the transition from the formulation posed in (8) to (9). Several explanations to justify this change have been already given. Further work is required to relate the two formulations and bound the difference between their solutions.

4. Experimental results

We present here six experiments demonstrating the separation, inpainting, and denoising obtained. In these experiments we have used the following parameters: $\lambda = 1$, $L_{\max} = 255$, $N \in [30, 200]$ (number of iterations), and $\gamma \in [0.5, 2]$. Note that the computational complexity of the MCA inpainting process is governed mostly by the number of iterations (inner and outer) $NL_{\max}$ and the complexity in applying the two forward and the inverse transforms.

Experiment 1. *Synthetic noiseless*: Figure 3 shows the `Adar` image with two cases of missing data (left). The `Adar` image is a synthetic combination of cartoon and texture (see [25,26] for more details). The results of the MCA-inpainting method using curvelet and global DCT are shown in Fig. 3 (right). Both results show no trace of the original holes, and look near-perfect.

Experiment 2. *Synthetic with additive noise*: In order to show that the proposed algorithm is capable of denoising as a by-product of the separation and inpainting, we added a zero mean white Gaussian noise ($\sigma = 10$) to the image `Adar` and then applied the MCA algorithm. Figure 4 shows the inpainting result and the residual. Notice that the residual is almost feature-less, implying that the noise was removed successfully, without taking true texture of cartoon content.

Experiment 3. *Barbara*: Figure 5 presents the `Barbara` image and its inpainting results for two different patterns of missing data as before. The MCA-inpainting method applied here used Wavelet and homogeneous decomposition level Wavelet Packets to represent the cartoon and the texture, respectively. Again, the results show no trace to the original holes, and look natural and artifact-free.

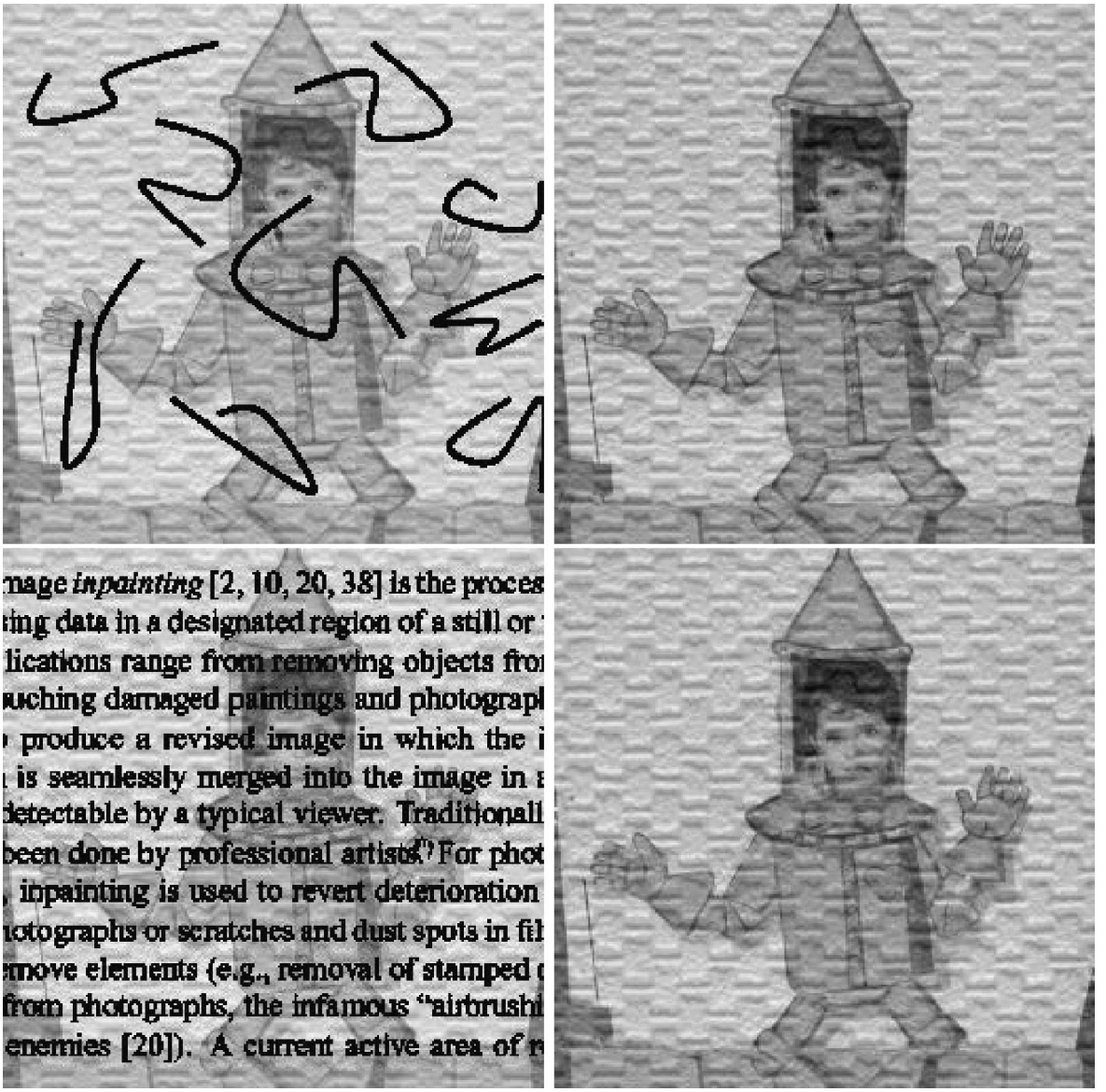

Fig. 3. Two synthetic `Adar` images (top and bottom left) with combined cartoon and texture, and imposed missing pixels. The results of the MCA inpainting are given in the top and bottom right.

Experiment 4. *Random mask*: Figure 6 presents the `Barbara` image and its filled-in results for three random patterns of 20%, 50%, and 80% missing pixels. The unstructured form of the mask makes the reconstruction task easier. These results are tightly related to the removal of salt-and-pepper noise in images. As before, the MCA-inpainting method applied here used Wavelet and Wavelet Packets to represent the cartoon and the texture respectively, and again, the results look natural and artifact-free.

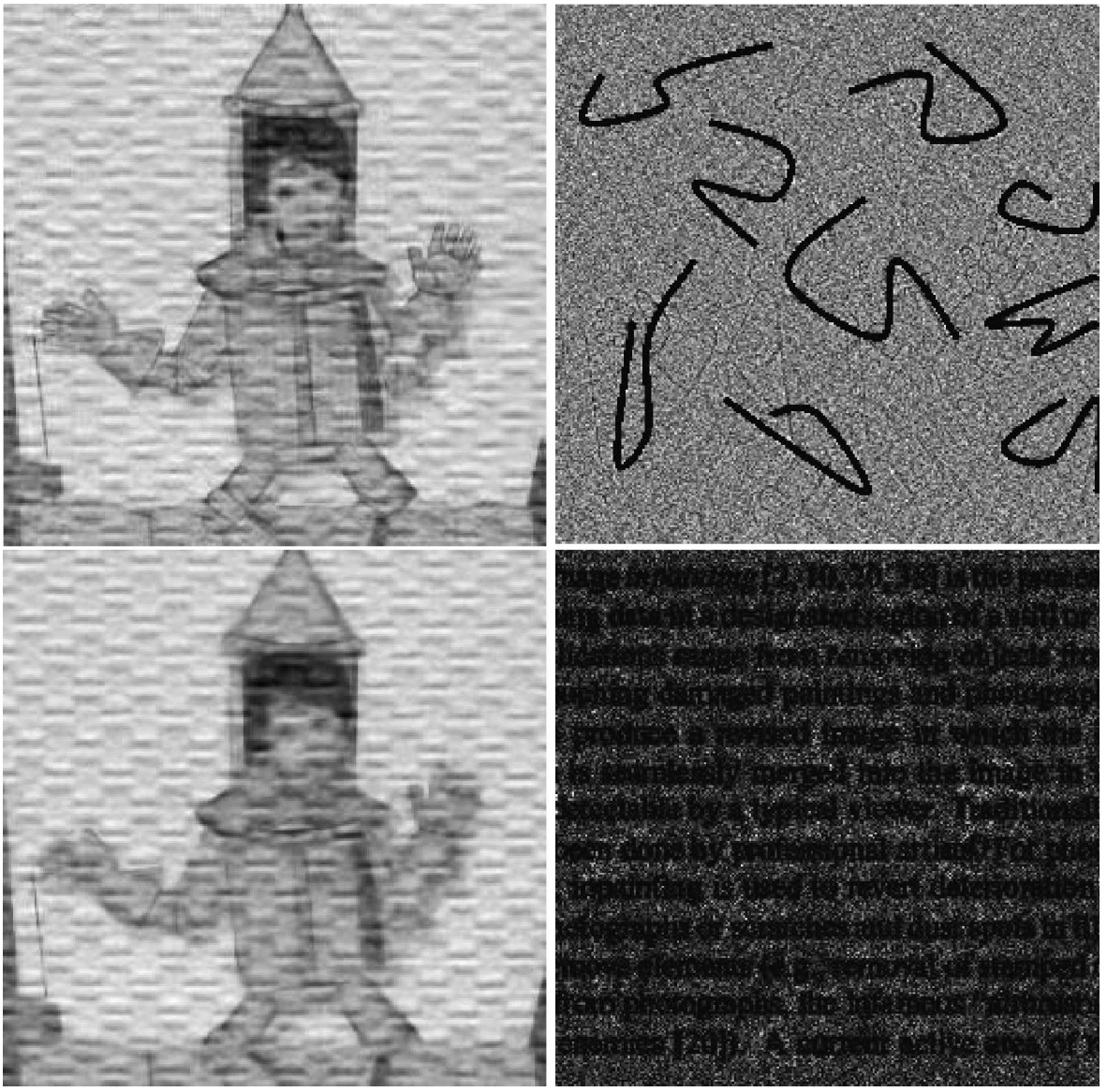

Fig. 4. MCA inpainting results for the `Adar` image (with two missing pixels masks—curves (top) and text (bottom)) contaminated by additive noise. Left: The inpainting result. Right: The residual.

Experiment 5. *Growing mask*: Figure 7 presents the `Barbara` image and its filled-in results for three patterns of missing pixels (9 blocks of size 8×8, 16×16, and 32×32 pixels). As before, the MCA-inpainting method applied here used Wavelet and Wavelet Packets to represent the cartoon and the texture, respectively. We see that as the regions of missing pixels grow, the recovery deteriorates, as expected, and smooth behavior is enforced.

Fig. 5. Two `Barbara` images (top and bottom left) and imposed missing pixels. The results of the MCA inpainting are given in the top and bottom right.

Experiment 6. *WMAP data*: Figure 8 shows real WMAP cosmic microwave background (CMB) data (see http://lambda.gsfc.nasa.gov/product/map for more details about this data), and imposed missing values (uniform gray areas represent missing data). Such masking is frequently encountered in actual cosmic data gathering, due to foreground components contamination. The CMB field is known to be stationary random field. We have used the global-DCT and the wavelet transform in our MCA-inpainting method and the results are shown in Fig. 8.

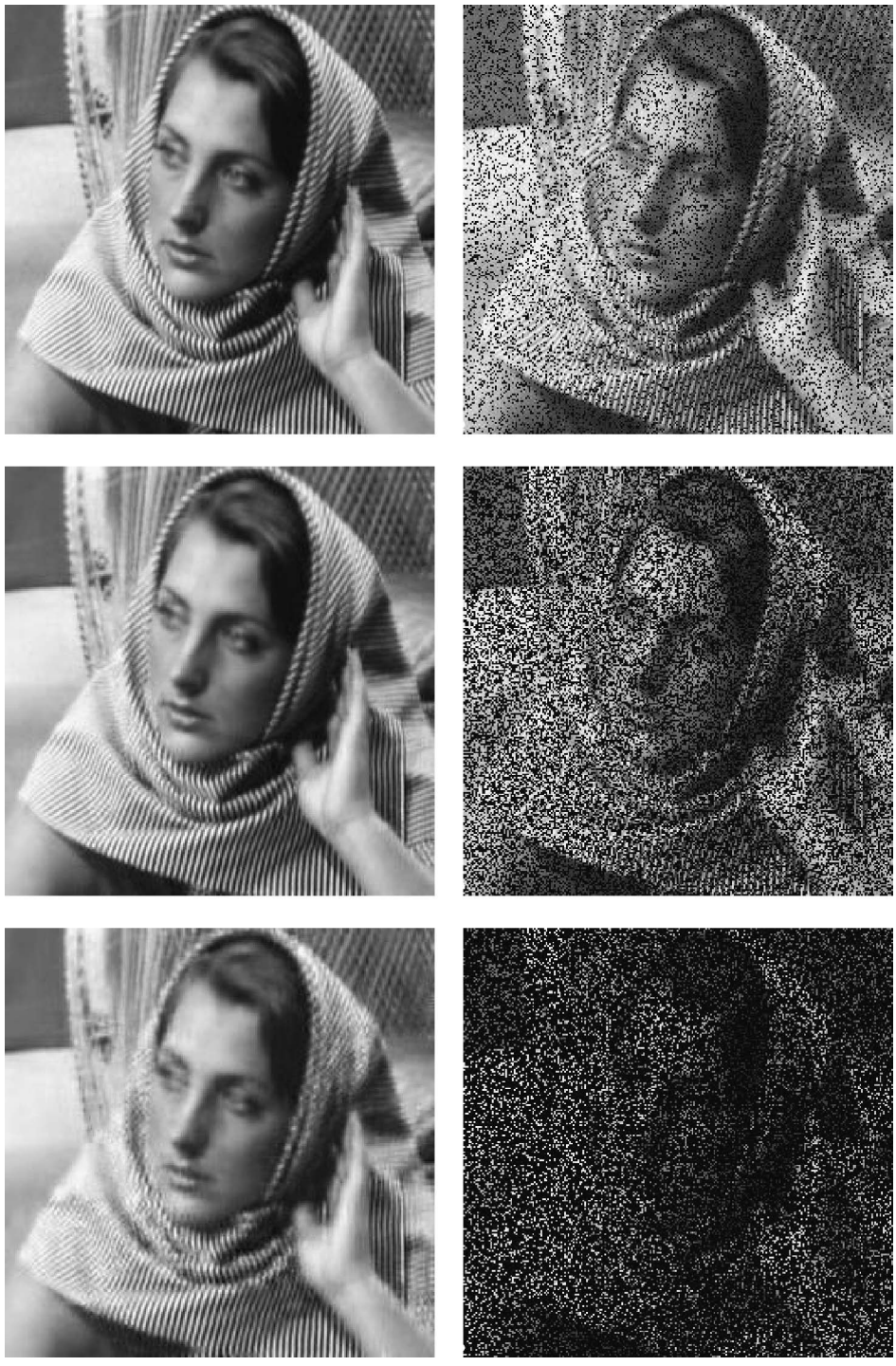

Fig. 6. Three `Barbara` images with 20%, 50%, and 80% missing pixels (right). The results of the MCA inpainting are given on the left.

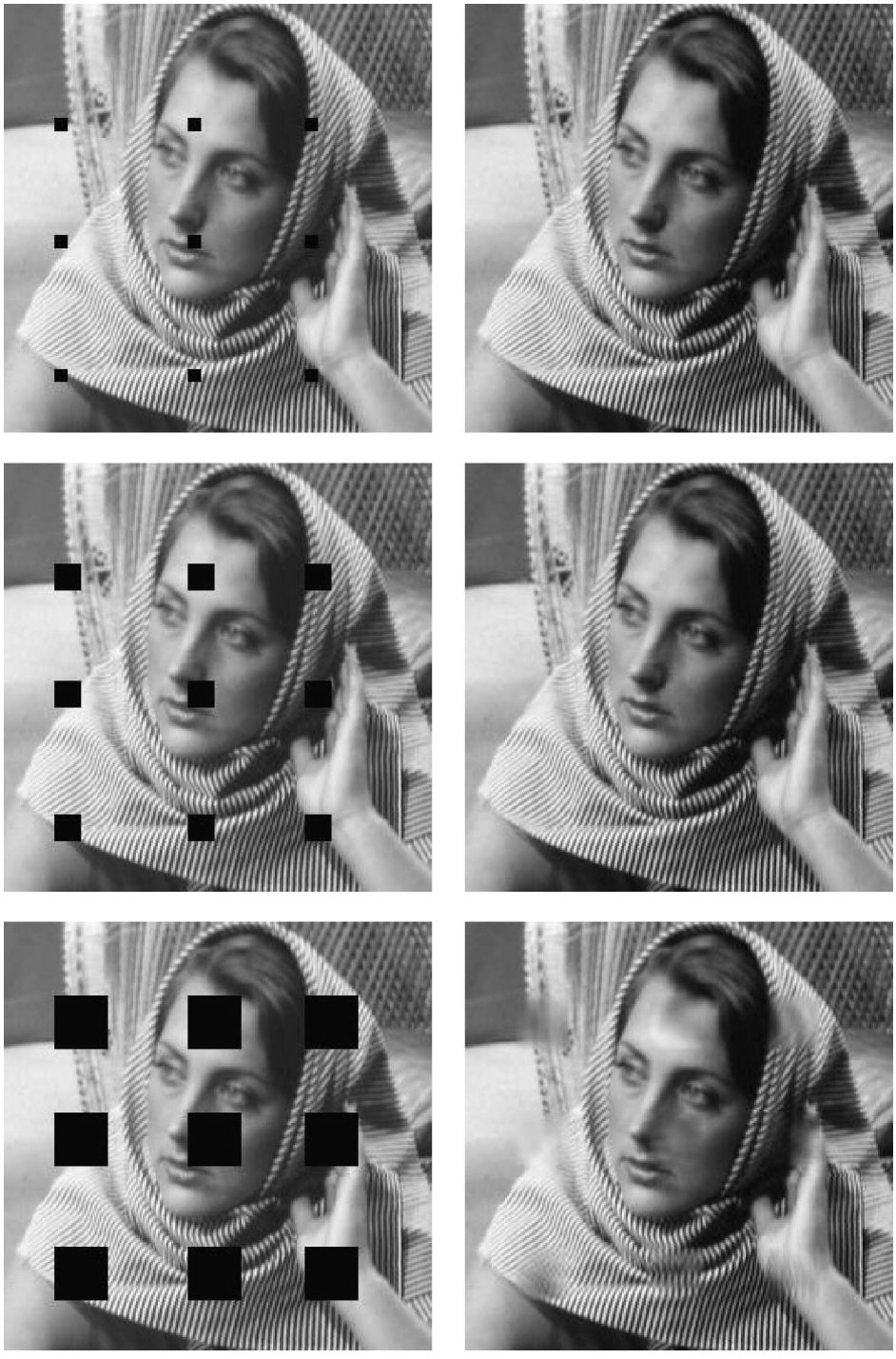

Fig. 7. Three `Barbara` images with three patterns of missing pixels—9 blocks of size 8×8, 16×16, and 32×32 pixels (right). The results of the MCA inpainting are given on the left.

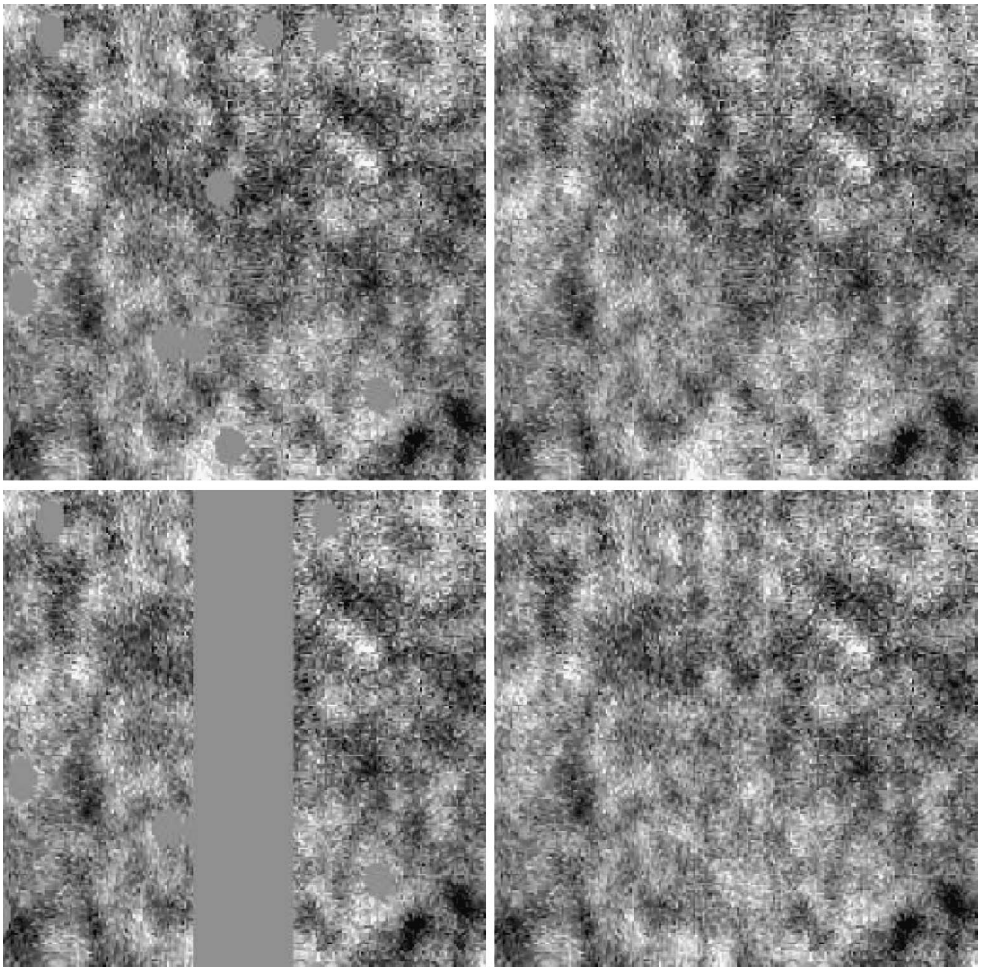

Fig. 8. WMAP cosmic microwave data and missing values. Upper left: Original image with missing data. Upper right: Result of the MCA inpainting. Bottom left: A large band of missing data have been imposed to the original image. Bottom right: The MCA inpainting result.

5. Discussion

In this paper we have presented a novel method for inpainting—filling holes in an image. Our method is based on the ability to represent texture and cartoon layers as sparse combinations of atoms of predetermined dictionaries. The proposed approach is a fusion of basis pursuit with the total-variation regularization scheme, allowing missing data and automatically filling in missing pixels.

Further theoretical work should attempt to document the performance of the method in filling in missing samples when the object truly has a sparse representation. It seems urgent to make a thorough study of the approximations used in proceeding from the original model to the numerical solution. Both topics are in our current research agenda.

References

[1] J.F. Aujol, G. Aubert, L. Blanc-Feraud, A. Chambolle, Image decomposition: Application to textured images and SAR images, Technical Report ISRN I3S/RR-2003-01-FR, INRIA—Project ARIANA, Sophia Antipolis, 2003.

[2] M. Bertalmio, L. Vese, G. Sapiro, S. Osher, Simultaneous structure and texture image inpainting, IEEE Trans. Image Process. 12 (2003) 882–889.

[3] J.S. De Bonet, Multiresolution sampling procedure for analysis and synthesis of texture images, in: Proceedings of SIGGRAPH, 1997.

[4] A.G. Bruce, S. Sardy, P. Tseng, Block coordinate relaxation methods for nonparametric signal de-noising, in: Proceedings of the SPIE—The International Society for Optical Engineering, vol. 3391, 1998, pp. 75–86.

[5] A.M. Bruckstein, M. Elad, A generalized uncertainty principle and sparse representation in pairs of $\mathbf{r}^n$ bases, IEEE Trans. Inform. Theory 48 (2002) 2558–2567.

[6] V. Caselles, G. Sapiro, C. Ballester, M. Bertalmio, J. Verdera, Filling-in by joint interpolation of vector fields and grey levels, IEEE Trans. Image Process. 10 (2001) 1200–1211.

[7] T. Chan, J. Shen, Local inpainting models and TV inpainting, SIAM J. Appl. Math. 62 (2001) 1019–1043.

[8] S.S. Chen, D.L. Donoho, M.A. Saunder, Atomic decomposition by basis pursuit, SIAM J. Sci. Comput. 20 (1998) 33–61.

[9] D. Donoho, X. Huo, Uncertainty principles and ideal atomic decomposition, IEEE Trans. Inform. Theory 47 (7) (2001) 2845–2862.

[10] D.L. Donoho, M. Elad, Optimally sparse representation in general (non-orthogonal) dictionaries via ℓ^1 minimization, Proc. Natl. Acad. Sci. 100 (2003) 2197–2202.

[11] D.L. Donoho, M. Elad, V. Temlyakov, Stable recovery of sparse overcomplete representations in the presence of noise, IEEE Trans. Inform. Theory (2004), in press.

[12] D.L. Donoho, X. Huo, Uncertainty principles and ideal atomic decomposition, IEEE Trans. Inform. Theory 47 (2001) 2845–2862.

[13] A.A. Efros, T.K. Leung, Texture synthesis by non-parametric sampling, in: IEEE International Conference on Computer Vision, Corfu, Greece, September 1999, pp. 1033–1038.

[14] M. Elad, A.M. Bruckstein, A generalized uncertainty principle and sparse representation in pairs of bases, IEEE Trans. Inform. Theory 48 (2002) 2558–2567.

[15] J.-J. Fuchs, On sparse representations in arbitrary redundant bases, IEEE Trans. Inform. Theory 50 (6) (2004) 1341–1344.

[16] R. Gribonval, M. Nielsen, Some remarks on nonlinear approximation with Schauder bases, East J. Approx. 7 (2) (2001) 267–285.

[17] R. Gribonval, M. Nielsen, Sparse representations in unions of bases, IEEE Trans. Inform. Theory 49 (12) (2003) 3320–3325.

[18] O.G. Guleryuz, Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising: Part I—Theory, IEEE Trans. Image Process. (2004), submitted for publication.

[19] O.G. Guleryuz, Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising: Part II—Adaptive algorithms, IEEE Trans. Image Process. (2004), submitted for publication.

[20] A.L. Bertozzi, M. Bertalmio, G. Sapiro, Navier–Stokes fluid dynamics and image and video inpainting, in: Proc. IEEE Computer Vision and Pattern Recognition (CVPR), 2001.

[21] V. Caselles, M. Bertalmio, G. Sapiro, C. Ballester, Image inpainting, in: Comput. Graph. (SIGGRAPH 2000), July 2000, pp. 417–424.

[22] Y. Meyer, Oscillating Patterns in Image Processing and Nonlinear Evolution Equations, University Lecture Series, vol. 22, Amer. Math. Soc., 2001.

[23] L.I. Rudin, S. Osher, E. Fatemi, Nonlinear total variation noise removal algorithm, Physica D 60 (1992) 259–268.

[24] J.-L. Starck, E. Candès, D.L. Donoho, The curvelet transform for image denoising, IEEE Trans. Image Process. 11 (6) (2002) 131–141.

[25] J.-L. Starck, M. Elad, D.L. Donoho, Image decomposition via the combination of sparse representations and a variational approach, IEEE Trans. Image Process. (2004), in press.

[26] J.-L. Starck, M. Elad, D.L. Donoho, Redundant multiscale transforms and their application for morphological component analysis, Adv. Imag. Electron Phys. (2004) 132.

[27] T.A. Tropp, Just relax: Convex programming methods for subset selection and sparse approximation, IEEE Trans. Inform. Theory (2004), in press.

[28] L.A. Vese, S. Osher, Modeling textures with total variation minimization and oscillating patterns in image processing, J. Sci. Comput. 19 (2003) 553–577.
[29] L.Y. Wei, M. Levoy, Fast texture synthesis using tree-structured vector quantization, in: Proceedings of SIGGRAPH, 2000.