Joint structure-texture low dimensional modeling for image decomposition with a plug and play framework

Antoine Guennec, Jean-François Aujol, Yann Traonmilin

Joint structure-texture low dimensional modeling for image decomposition with a plug and play framework

Antoine Guennec*, Jean-François Aujol , and Yann Traonmilin

Abstract. To address the problem of separating images into a structure and a texture component, we introduce a joint structure-texture model. Instead of considering two separate regularizations for each component, we consider a joint structure-texture model regularization function that takes both components as inputs. This allows for the regularization to take into account the shared information between the two components. We present evidence that shows a performance gain compared to separate regularization models. To implement the joint regularization, we adapt the plug and play framework to our setting, using deep neural networks. We train the corresponding deep prior on a randomly generated synthetic dataset of examples of this model. In the context of image decomposition, we show that while trained on synthetic datasets, our plug and play method generalizes well to natural images. Furthermore, we show that this framework permits to leverage the structure-texture decompositions to solve inverse imaging problems such as inpainting.

Key words. image decomposition, low dimensional models, regularization learning, plug-and-play prior

MSC codes. 68U10, 62H35, 90C26, 94A08

1. Introduction.

The inverse problem of decomposing an image into structure and texture components (also known as cartoon-texture decomposition) has been a longstanding area of research, with many applications such as image/video compression, material recognition, biomedical imaging and texture enhancement/removal. The problem is defined as follows: given an image $f \in E = \mathbb{R}^{n_1 \times n_2}$, find a decomposition

$$f = u + v \tag{1.1}$$

such that the image u is a piecewise constant (or piecewise smooth depending on the definition) approximation of f, containing the basic geometries present in the image. The image v contains the texture which is locally zero-mean and contains the oscillating and local patterns. As the system associated to the problem is underdetermined, prior information on the cartoon and texture components is needed to hope for a satisfactory decomposition.

The classical method achieve such a decomposition is to solve the optimization problem

$$\underset{u,v \in E}{\text{minimize}}\ R_s(u) + \lambda R_t(v) \quad \text{subject to } f = u + v \tag{1.2}$$

where $R_s(\cdot)$ and $R_t(\cdot)$ are regularization functions that enforce the characteristics of the structure and texture components respectively, and λ is a tuning parameter that balances the relative strengths of the structure and the texture respective priors. Many preceding works use the total variation [2, 30, 31] for the regularization function R_s in order to enforce some piecewise constant characteristics into the structure component. The texture regularization has been the center of attention of the different models, with various proposals such as L^2

*IMB, UMR 5251, Université de Bordeaux (antoine.guennec@math.u-bordeaux.com).

regularization [30, 40] or norms that emphasize sparsity [34, 41] or low-rank of the matrix of texture patches [31, 23, 16]. However, these approaches to image decomposition have two flaws:

1. The structure and texture priors are enforced separately. As we will argue more precisely in Section 2.2, while locally the two components are uncorrelated, this is not the case in the full image: the structural component often defines the frontiers of different structures present in the image. This often leads to uncertainty at the edges in the decomposition.
2. They introduce a necessary tuning parameter λ to balance the two regularization models. Current methods are relatively costly and it is often needed to perform multiple runs of the decomposition algorithm in order to set this parameter correctly. Without prior information on the underlying structure and texture components of an image, it is not possible to set the correct parameter. Furthermore, additional parameters are often introduced in the regularization functions. This leads to difficult and/or misleading comparison between proposed methods.

To the best of our knowledge, there are no methods considering a joint model on structure and texture. Moreover, the general problem of building good regularizations for complex combinations of low-dimensional models in inverse problems is in general an open question (see e.g. [24]).

For parameter tuning, there have been multiple attempts to mitigate this issue. In [3], it was proposed to use the correlation between the two components in order to tune the parameter for different total variation-based variational models. In [16] it was proposed to automatically tune the low patch rank model [31] by estimating the gradient sparsity of the structure and the patch-rank of the texture. However, setting a global parameter is still needed.

In this paper, to address these two problems, we explore the use of plug and play methods in order to construct a new regularization function for image decomposition.

1.1. The plug and play framework.

A recent advance in the field of inverse problems has been the introduction of the plug-and-play (PnP) framework [36]. Inverse problems are often solved via the minimization scheme

$$\underset{x\in E}{\text{minimize}}\ R(x) + F(x, y), \tag{1.3}$$

where R is the regularization term, F is the data fidelity term with respect to the observation y. For example, in the case where an image x_0 is corrupted by a linear operator $\mathcal{A}$ and a white Gaussian noise ϵ, i.e $y = \mathcal{A}x_0 + \epsilon$, we may set $F(x, y) = \|\mathcal{A}x - y\|_2^2$.

The PnP method leverages proximal splitting algorithms, established initially for convex problems, by substituting the traditional proximal operator $\text{Prox}_{R,\eta}(x)$ with a denoiser $D(x)$. In this context, associating a denoiser with a regularization function is not straightforward if we wish to obtain convergence properties. First initiated in [29], it was proposed to construct an explicit regularization function from a denoiser. However, given a differentiable denoiser $D : \mathbb{R}^N \to \mathbb{R}^N$, it was later proven in [28] that the desirable property

$$\nabla R = Id - D, \tag{1.4}$$

cannot hold without a Jacobian Symmetry property. Other models such as [17, 15] have been proposed, in order to bypass this constraint. In this paper, we focus on the gradient step denoiser [17], in which the regularization is set as $R(x) = \frac{1}{2}\|x - N(x)\|_2^2$, where $N : E \to E$ is parametrized by a neural network and the denoiser is defined from the constraint (1.4). As R is differentiable, (1.3) can be solved using descent iterative schemes such as the forward-backward algorithm (FB)

$$\begin{cases} z_{k+1} = x_k - \tau \nabla R(x_k) \\ x_{k+1} = \mathrm{Prox}_{F(\cdot,y),\eta}(z_{k+1}) \end{cases}; \tag{1.5}$$

where the proximal operator of a function $G : \mathbb{R}^N \to \mathbb{R}$ is defined by

$$\mathrm{Prox}_{G,\eta}(x) := \arg\min_z G(z) + \frac{1}{2\eta}\|z - x\|_2^2 . \tag{1.6}$$

1.2. Contributions. In this work, we introduce the joint structure-texture model for image decomposition and its implementation using an adapted PnP framework

- In Section 2, we present a low-dimensional model of image where structure and texture are considered to share support information. To enforce this model, we deviate from the classical paradigm (1.2) by considering the minimization of a single function that acts on both the structure and texture at the same time, i.e the structure-texture decomposition is the result of the optimization problem

$$\underset{\mathrm{x}=(u,v)\in E\times E}{\text{minimize}} \; R(\mathrm{x}) \quad \text{subject to } f = u + v. \tag{1.7}$$

- In Section 3, we construct a regularization for the joint structure-texture model, by adapting the PnP framework: it suffices to train a joint structure-texture denoiser. This framework removes the necessity of a tuning parameter for the structure-texture decomposition. In place, we provide an optional parameter that balances the projection direction onto the constraint $f = u + v$.
- In Section 4, we construct a prior of the decomposition model, using a database of randomly generated synthetic decompositions to train the denoiser in our PnP algorithm. The resulting regularization function is able to take into account information shared between the structure and texture. We demonstrate that our adapted PnP framework is able to define regularizations adapted to complex combinations of two low-dimensional models, which was shown to be generally impossible with just the sum of individual regularizations. Furthermore, we present evidence that the joint structure-texture modelization outperforms the usual separated models (Section 4.3).
- In Section 5, we perform experiments on synthetic and real natural images in order to illustrate the performance of our method. In particular, our constructed regularization allows to solve difficult inverse problems such as inpainting, working simultaneously on both the structure and texture component (Section 5.2). We also show that this model, while trained on synthetic data, is able to generalize well to natural images (Section 5.3) leading to interesting perspectives for the construction of deep priors for image processing.

1.3. Related Work. The first structure-texture decomposition models relied on variational methods, using the total variation to characterize the structural component and a function space norm to constrain the texture component, such as the L^2-norm [30], G-norm [22, 37] or $\mathcal{H}$-norm [3, 2]. While theoretically well-founded and able to capture the oscillating nature of texture, these norms are either difficult to implement or cannot capture textures with a small magnitude. To counteract the staircase effect given by the total variation [8], other regularization such as the total generalized variation [7] and the relative total variation [38] were proposed.

A more modern approach has been to consider the structure-texture decomposition in the context of sparse/low-rank priors. One of the earliest approach was to consider that texture can be sparsely represented in a suitable given transformation (e.g discrete cosine transform (DCT), Gabor transform) [34, 10]. While very successful in some applications, the issue with this approach is that many textures that arise in practical applications cannot be modeled by DCT or other related dictionaries. More recently, this approach was extended to use convolutional sparse coding instead [41], where convolutional filters are learned beforehand. Another approach was to consider that the matrix of texture patches is of low patch rank (LPR) [31]. However, this approach can fail if too many different textures are present in the image since the resulting sets of textures no longer live in a small patch-space. [23] proposed the blockwise low-rank texture model to counteract against this issue with LPR. Similarly to the low patch-rank prior, in [39] the cartoon and texture were separated based upon local patch recurrence with a given orientation. All of the aforementioned models above provide more or less an appropriate decomposition. However, they are relatively slow and require a tuning parameter to balance the resulting structure and texture. To address this matter, [16] took advantage of the underlying low dimensionality of the structure and texture spaces in order to provide a near tuning parameter-free and highly parallelized localized version of the LPR model.

Recently, learning based approaches have been proposed to solve the image decomposition problem. In [45] the authors proposed a self-example and unsupervised learning approach where the structure-texture decomposition associated regularization is optimized through the back propagation of a neural network. Similarly, in [32] it was proposed to recover the structural component from a random input z from a convolutional generative neural network f_θ, and to model the texture as low-rank. In [11], the authors showed that the iterative steps in the minimization of TV-ℓ_1 are similar to the architecture of an LSTM neural network and they proposed to use an LSTM in order to unfold the iterative hard-thresholding algorithm of TV-ℓ_1. Similarly, in [18], the authors proposed to use a CNN network in order to learn the structure prior. In [33, 44], other methods based upon unfolding the TV proximal operator have been proposed. One of the closest approach to our work can be found in [21], where the authors proposed to learn an image decomposition neural network training upon a handmade structure-texture dataset consisting of cartoon images onto which a homogeneous texture was added. However, this approach lacks two core details: texture locality (see Figure 1) and an associated regularization function to the decomposition that can thereafter be used to solve inverse problems.

2. Structure-Texture decomposition as a low dimensional recovery problem. In this section, we describe the image decomposition problem as a low-dimensional recovery problem. We highlight the fact that an optimal regularization for this problem cannot be the sum of a structural regularization and a textural regularization of the form (1.2), thus justifying the introduction of our framework for a joint regularization (1.7).

A way to describe image decomposition is to consider it as a low-dimensional recovery problem. In this setting, the underlying assumption is that the image we wish to decompose belongs to the sum of two low dimensional models, i.e. $f = u_0 + v_0$ with u_0 and v_0 each belonging to a low-dimensional model, denoted by Σ_s for the structure model and Σ_t for the texture model respectively. Then, the decomposition problem becomes: *recover* $(u_0, v_0) \in \Sigma_s \times \Sigma_t$ *from* $f = u_0 + v_0$.
For each data model Σ_s and Σ_t, we typically set corresponding regularization functions R_s and R_t whose minimization should enforce Σ_s and Σ_t respectively. We aim to recover (u_0, v_0) (or at least an approximation) via the optimization problem

$$\underset{(u,v)\in E\times E}{\text{minimize}}\ R_s(u) + R_t(v) \quad \text{subject to } f = u + v. \tag{2.1}$$

Optimally in this setting [6, 35], the regularization functions should be set as

$$R_s(u) = \text{dist}(u, \Sigma_s)^2 \quad \text{and} \quad R_t(v) = \text{dist}(v, \Sigma_t)^2. \tag{2.2}$$

Since this approach generally leads to NP-hard problems (e.g ℓ_0, rank minimization), a convex relaxation is often considered instead (e.g ℓ_1 norm used instead of ℓ_0 for sparsity). This setting can also be viewed in the context of compressive sensing. By setting the linear operator $\mathcal{A}=(Id\ Id)$, we aim to recover $\mathrm{x}_0=(u_0, v_0) \in \Sigma_s \times \Sigma_t$ from measurements $f=\mathcal{A}\mathrm{x}_0$, with $\dim(f) = n_1 n_2 < 2n_1 n_2 = \dim(\mathrm{x})$.

The choice of Σ_s and Σ_t is also of utmost importance to tune the texture scaling dilemma (which is tightly linked to the image resolution): repetitive patterns may be part of the structure if enlarged (zoom in) or be part of the texture component when shrunk (zoom out). In between these two states, it is ambiguous to distinguish between structure and texture with confidence. This is a choice that should be set in accordance to the specific application we wish to perform.

2.1. Previous work on low dimensional recovery for image decomposition. For the structure component, the total variation

$$\|u\|_{TV} = \sum_{i\in\Omega} \|(\nabla u)_i\|_2 = \|\nabla u\|_1, \quad \text{with } \Omega = [\![1, n_1 n_2]\!], \tag{2.3}$$

has been widely used to enforce gradient-sparsity and its associated low dimensional model is given by

$$\Sigma_{\text{GS}} = \{u \in \mathbb{R}^{n_1\times n_2} |\, \|\nabla u\|_0 \le k\}, \tag{2.4}$$

the set of vectors that are k-gradient-sparse. On the other hand, for the texture component a variety of models have been proposed. We present a (non-exhaustive) list of previous methods:

1. The earliest example of image decomposition by exploiting sparsity is given by [34], where we assume that both the structure and texture are sparse in an appropriate overcomplete dictionary. In essence, we assume that

$$u_0 \in \Sigma_{D_s} = \{\mathcal{D}_s x \mid \|x\|_0 \leq k_1\} \quad \text{and} \quad v_0 \in \Sigma_{D_t} = \{\mathcal{D}_t y \mid \|y\|_0 \leq k_2\}, \tag{2.5}$$

where $\mathcal{D}_s$ and $\mathcal{D}_t$ are the chosen overcomplete dictionaries. For example, $\mathcal{D}_s$ may correspond to a curvelet dictionary and $\mathcal{D}_t$ may correspond to a DCT or Gabor dictionary. We recover the decomposition via the minimization of an ℓ_1 optimization problem

$$(x_0, y_0) = \arg\min_{x,y} \|x\|_1 + \|y\|_1 \quad \text{subject to } f = \mathcal{D}_s x + \mathcal{D}_t y, \tag{2.6}$$

and the resulting decomposition is given by $(u, v) = (\mathcal{D}_s x_0, \mathcal{D}_t y_0)$. In fact, with the appropriate constraints upon the dictionaries and underlying sparsity of u_0 and v_0, (2.6) is able to exactly recover (u_0, v_0).
2. In the Low Patch rank interpretation of texture (LPR) model [31], the texture is considered to be of low patch-rank, i.e

$$v_0 \in \Sigma_{\text{LPR}} = \{v \in \mathbb{R}^{n_1 \times n_2} | \text{ rank}(\mathcal{P}(v)) \leq l\}, \tag{2.7}$$

where $\mathcal{P}$ is a patch operator. Moreover, since the nuclear norm

$$\|X\|_* = \sum_{i=1}^{\min(n_1,n_2)} \sigma_i(X) \tag{2.8}$$

is a convex relaxation of the rank, (2.7) is able to recover the low patch-rank textures (under some conditions) . The decomposition is pursued via the optimization problem:

$$\underset{(u,v)}{\text{minimize}}\ \mu \|u\|_{TV} + \gamma \|\mathcal{P}(v)\|_* \quad \text{subject to } f = u + v. \tag{2.9}$$

3. Similarly, in the Blockwise Low-Rank Texture Characterization (BNN) model [23] the texture is considered to be of low-rank 'blockwise', with $v_0 = v_0^1 + ... + v_0^m$ and for each $i \in \{1, \ldots, m\}$

$$v_0^i \in \Sigma_{\text{BNN}}^i = \{v \in \mathbb{R}^{n_1 \times n_2} | \text{ rank}(P_{k_i,\delta_i} \circ S_{\theta_i}(v)) \leq l\}, \tag{2.10}$$

where P_{k_i,δ_i} is a periodically-expanding operator with parameters (k_i, δ_i) and $S_{\theta_i}(v)$ is a shearing operator with parameter θ_i (see [23] for more information). Then, the BNN model of the texture component is given by

$$\Sigma_{\text{BNN}} = \Sigma_{\text{BNN}}^1 + \cdots + \Sigma_{\text{BNN}}^m \tag{2.11}$$

and structure and texture are recovered by the optimization problem

$$\underset{u,v \in E}{\text{minimize}}\ \mu \|u\|_{TV} + \sum_{i=1}^{m} \gamma \|v\|_{*,\text{BNN}}^i \quad \text{subject to } f = u + v. \tag{2.12}$$

4. In the convolutional sparse and low rank coding-based image decomposition model [41], convolutional filters $\{d_{s,i}\}_{i=1}^{K_s}$, $\{d_{t,i}\}_{i=1}^{K_t}$ that sparsely represent the structure and texture components are learned. The associated low dimensional models are given by (2.13)

$$\Sigma_s^{CS} = \left\{ \sum_{i=1}^{K_s} d_{s,i} * x_i \mid \sum_{i=1}^{K_s} \|x_i\|_0 \leq k_1 \right\} \quad \text{and} \quad \Sigma_t^{CS} = \left\{ \sum_{i=1}^{K_t} d_{t,i} * x_i \mid \sum_{i=1}^{K_s} \operatorname{rank}(x_i) \leq k_2 \right\}.$$

The decomposition model can be further restricted by considering that the structure component $\sum_{i=1}^{K_s} d_{s,i} * x_i$ is also gradient sparse.

Note that while the ℓ^1-norm (respectively the nuclear norm) has been shown to be optimal for sparse recovery (respectively low-rank recovery) [35], all these methods consider a sum of regularizations for decomposition. This "sum" approach is adapted for product models $\Sigma_s \times \Sigma_r$. We argue in the following that structure and texture are not best approximated by such product models.

2.2. The joint structure-texture with shared support model. For natural images, the structure and texture components should not be considered disjointedly because they share some common information: the support. While locally the structure and texture components can be considered uncorrelated, it is not so the case when taking the whole image into account. Usually, the structure and texture present in an image share a common border (e.g Figure 1), i.e. *the texture is expected to end when the structure also ends.*

Figure 1: An example of decomposition of the Barbara image. From left to right: original image f, structure component u, texture component v. We observe that structure and texture share a common border.

Consider Σ_s and Σ_t two low-dimensional models which contain all the structure and texture components separately, for example we may choose gradient sparsity $\Sigma_s = \Sigma_{\text{GS}}$ and low patch rank $\Sigma_t = \Sigma_{\text{LPR}}$. We define the notion of structure and texture with a given support.

Definition 2.1. *Consider a set of disjoint supports $\mathcal{I} = (I_r)_{r=1}^{|\mathcal{I}|}$ ($I_r \subset [\![1, n_1]\!] \times [\![1, n_2]\!]$) and u_I the restriction of u to the support I. We define the support-wise structure and texture low*

dimensional models as

$$\begin{aligned}\Sigma_{s,\mathcal{I}} &= \left\{ u \in \Sigma_s : |\nabla u_{I_r}| = 0, \forall I_r \in \mathcal{I} \right\};\\ \Sigma_{t,\mathcal{I}} &= \left\{ \sum_r \mathbb{1}_{I_r} \cdot v_r \mid v_r \in \Sigma_t \right\}.\end{aligned} \tag{2.14}$$

By abuse of notation, we suppose that ∇u_{I_r} only contains the gradients inside the support I_r (we exclude the gradients on the boundary of I_r).

Fundamentally, this definition stems from the fact that textures can be expanded (infinitely) on a canvas and the observed textures in a local section of an image are delimited by the structure. Hence the consideration that a local texture should be $\mathbb{1}_{I_r} \cdot v_r$ in the definition of the support-wise texture model.

We set $\mathcal{Q}(n_1, n_2)$ as the set of partitions of $[\![1, n_1]\!] \times [\![1, n_2]\!]$[1]with connected sets. We can now define the joint low-dimensional structure model.

Definition 2.2. *We define the joint structure-texture with a shared support model as*

$$\Sigma_{s\otimes t} = \bigcup_{\Omega \in \mathcal{Q}(n_1,n_2)} \Sigma_{s,\Omega} \times \Sigma_{t,\Omega} \tag{2.15}$$

We immediately remark that $\Sigma_{s\otimes t}$ is a union of product models that cannot be written as a cartesian product of structure and texture.

2.3. On optimal regularization for low dimensional models ?. In the case of separated models, where we consider that the structure and texture components are uncorrelated, the optimization problem (2.1) is natural to consider. Indeed, if we set the regularization functions R_s, R_t as in (2.2) and $R_{s,t}(u, v) = \operatorname{dist}((u, v), \Sigma_s \times \Sigma_t)^2$, since $\operatorname{dist}((u, v), \Sigma_s \times \Sigma_t)^2 = \operatorname{dist}(u, \Sigma_s)^2 + \operatorname{dist}(v, \Sigma_t)^2$, we have

$$\begin{aligned}\min_{\substack{u,v\in E\\ u+v=f}} R_{s,t}(u, v) &= \min_{\substack{u,v\in E\\ u+v=f}} \operatorname{dist}(u, \Sigma_s)^2 + \operatorname{dist}(v, \Sigma_t)^2\\ &= \min_{\substack{u,v\in E\\ u+v=f}} R_s(u) + R_t(v).\end{aligned} \tag{2.16}$$

Hence, the optimal strategy in this case is to minimize $R_s + R_t$. However, in the case of the joint structure-texture model, this property is no longer satisfied and shared borders between the two components imposes an additional constraint on the optimization problem. Since the model $\Sigma_{s\otimes t}$ is more constrained than $\Sigma_s \times \Sigma_t$, a dedicated joint regularization can thus potentially perform better.

Note that a similar problem has been studied in [24], where the recovery of matrices that are both sparse and low-rank is studied (intersection of models). Oymak et al. show that a sum of dedicated regularizations cannot perform better than individual regularizations. Later work studies theoretically heuristics to solve such problems [12]. This shows that designing

[1] $\Omega = \{\Omega_1, \ldots, \Omega_m\} \in \mathcal{Q}(n_1, n_2) \iff \bigcup_{i=1}^m \Omega_i = [\![1, n_1]\!] \times [\![1, n_2]\!]$ and $\Omega_i \cap \Omega_j = \emptyset,\ \forall i \neq j$.

joint regularization functions for such complex combinations of models directly is not an easy task. In the next Section, we introduce a PnP method to design such adapted regularizations. This framework permits to stay within the global theory of regularization of low-dimensional models.

3. PnP for Image decomposition. Instead of considering two regularization functions in order to decompose an image (one for each component), we propose to use a single regularization function that takes both the structure and texture components as input. By doing so, we solve the problem of joint regularization and we remove the necessity of a structure/texture balance tuning parameter. We aim to recover $\mathrm{x}_0 = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} \in \Sigma_{s \otimes t}$ from the original image $f = \mathcal{A}\mathrm{x}_0$, with $\mathcal{A} = (Id\ Id)$, via an optimization of the form

$$\underset{\mathrm{x}=(u,v)}{\text{minimize}}\ R(\mathrm{x}) \quad \text{subject to } f = \mathcal{A}\mathrm{x}. \tag{3.1}$$

However, setting an explicit regularization that achieves this goal is clearly inconceivable as minimizing over the set of partitions $\mathcal{Q}(n_1, n_2)$ introduces an exploding complexity.
We propose to use a gradient-step denoiser in order to obtain a regularization function R that accurately captures the joint structure-texture with a shared support model. Experiments validating this approach are given in Section 4 and Section 5.

3.1. The gradient step denoiser applied to image decomposition. In [17], the authors proposed the *gradient step denoiser*, a plug-and-play scheme in which the denoiser is connected to an explicit regularization functional. The gradient step denoiser takes the form

$$D(x) = (Id - \nabla R)(x), \tag{3.2}$$

where R is the associated regularization function

$$R(x) = \frac{1}{2} \|x - N(x)\|^2 \tag{3.3}$$

and $N : \mathbb{R}^n \to \mathbb{R}^n$ is parametrized by a neural network. In the context of plug and play, the authors used the gradient step denoiser with a forward-backward algorithm to solve an optimization problem of the form

$$\underset{x}{\text{minimize}}\ R(x) + F(x) \tag{3.4}$$

where $F : \mathbb{R}^n \to R$ is the data fidelity term. For example, in the case of image restoration from a linear observation (deblurring, inpainting, etc...), we may set $F(x) = \|y - \mathcal{A}x\|_2^2$ where y is our degraded image and $\mathcal{A}$ the degradation operation.

If we set $\mathcal{C}_f := \{\mathrm{x} = (u, v) \in E \times E \mid \begin{pmatrix} Id & Id \end{pmatrix} \mathrm{x} = f\}$, the convex set of couples (u, v) that decompose f, then (3.1) is equivalent to

$$\underset{\mathrm{x} \in E \times E}{\text{minimize}}\ R(\mathrm{x}) + \chi_{\mathcal{C}_f}(\mathrm{x}) \tag{3.5}$$

where χ is the indicator function, i.e for a convex set $\mathcal{C}$, $\chi_{\mathcal{C}}(x) = \begin{cases} 0 & \text{if } x \in \mathcal{C} \\ +\infty & \text{otherwise} \end{cases}$. Then, the decomposition (3.5) fits nicely in the context of image restoration (3.4) with $F = \chi_{\mathcal{C}_f}$ which can be solved using a projected gradient descent [5]. The following Lemma gives explicitly the proximal operator of

Lemma 3.1. *The proximal operator of $\chi_{\mathcal{C}_f}$ (the orthogonal projection onto $\mathcal{C}_f$) for* $\mathrm{x} = \begin{pmatrix} u \\ v \end{pmatrix}$ *is given by*

$$\mathcal{P}_{\mathcal{C}_f}(\mathrm{x}) := \mathrm{Prox}_{\chi_{\mathcal{C}_f},\lambda}(\mathrm{x}) = \begin{pmatrix} u \\ v \end{pmatrix} - \frac{1}{2}\begin{pmatrix} u+v-f \\ u+v-f \end{pmatrix} \tag{3.6}$$

Proof. This is an immediate consequence of the more general Lemma 3.2. For $b = f = u + v$, we have (with $L = \begin{pmatrix} Id & Id \end{pmatrix}$ and L^+ is the pseudo-inverse of L),

$$L^+Lx = \begin{pmatrix} Id & Id \end{pmatrix}^T \left(\begin{pmatrix} Id & Id \end{pmatrix}^T \begin{pmatrix} Id & Id \end{pmatrix}\right)^{-1} \begin{pmatrix} Id & Id \end{pmatrix} x = \frac{1}{2}\begin{pmatrix} u+v \\ u+v \end{pmatrix}$$

and $L^+b = \frac{1}{2}\begin{pmatrix} f \\ f \end{pmatrix}$ and

$$\begin{aligned} \mathrm{Prox}_{\chi_{\mathcal{C}_f},\lambda}(\mathrm{x}) &= (I - L^+L)x + L^+b \\ &= \begin{pmatrix} u \\ v \end{pmatrix} - \frac{1}{2}\begin{pmatrix} u+v-f \\ u+v-f \end{pmatrix}. \end{aligned} \tag{3.7}$$

■

In full, the projected gradient step (equivalent to the Forward-Backward algorithm (1.5)) iterations for image decomposition to minimize (3.1) with R satisfying (3.2), is by

$$\begin{cases} \mathrm{y}_{k+1} = (1-\tau)\mathrm{x}_k + \tau D(\mathrm{x}_k) \\ \mathrm{x}_{k+1} = \mathcal{P}_{\mathcal{C}_f}(\mathrm{y}_k) \end{cases} \tag{3.8}$$

where τ is the gradient step parameter. Notice that in the convex case, the Forward-Backward algorithm (1.5) converges as soon as $\tau \leq \frac{2}{L}$, where L is the Lipschitz constant of the regularization function R.

We train the gradient step denoiser with Gaussian noise (3.2) by minimizing the mean square error loss function

$$\mathcal{L}(D) = \mathbb{E}_{\mathrm{x}\in\Sigma_{s\otimes t},\epsilon\sim\mathcal{N}(0,\sigma_2)} \|D(\mathrm{x}+\epsilon) - \mathrm{x}\|_2^2 . \tag{3.9}$$

Essentially, the loss guarantees that the denoiser 'projects' well onto $\Sigma_{s\otimes t}$, since

$$\begin{aligned} \mathrm{dist}(D(\mathrm{x}+\epsilon), \Sigma_{s\otimes t})^2 &= \inf_{\mathrm{y}\in\Sigma_{s\otimes t}} \|D(\mathrm{x}+\epsilon) - \mathrm{y}\|_2^2 \\ &\leq \|D(\mathrm{x}+\epsilon) - \mathrm{x}\|_2^2 , \end{aligned} \tag{3.10}$$

Algorithm 3.1 Joint structure-texture gradient descent

Param.: $\tau > 0$
Input f
Output: The output structure and texture $\hat{\mathrm{x}} = (\hat{u}, \hat{v})$
$\mathrm{x}_0 = (f, 0)$
while not converged **do**
  $\mathrm{y}_{k+1} = (1-\tau)\mathrm{x}_k + \tau D(\mathrm{x}_k)$
  $\mathrm{x}_{k+1} = \mathcal{P}_{\mathcal{C}_f}(\mathrm{y}_{k+1})$
end while

for any $\mathrm{x} \in \Sigma_{s\otimes t}$ and a perturbation ϵ such that $\mathrm{x} + \epsilon \notin \Sigma_{s\otimes t}$. In our approach, we deviate from the original implementation as we do not add the noise level σ as input of the model (blind denoising). The training is performed on multiple noise levels without prior knowledge of σ. Furthermore, 30% of the training was performed without noise. Similarly to [26], we observed that prioritizing the training of the denoiser on low noise levels greatly improved the overall performance of the denoising.

By using differentiable layers in N (e.g ELU layer instead of RELU), we ensure that the projected gradient descent converges. Indeed, $\chi_{\mathcal{C}_f}$ is lower semi-continuous and thus we are in the convergence conditions provided by Theorem 1 of [17]. In what follows, we parametrized the neural network N using a DRUNet architecture (Fig. [42]), with ELU layers instead of RELU.

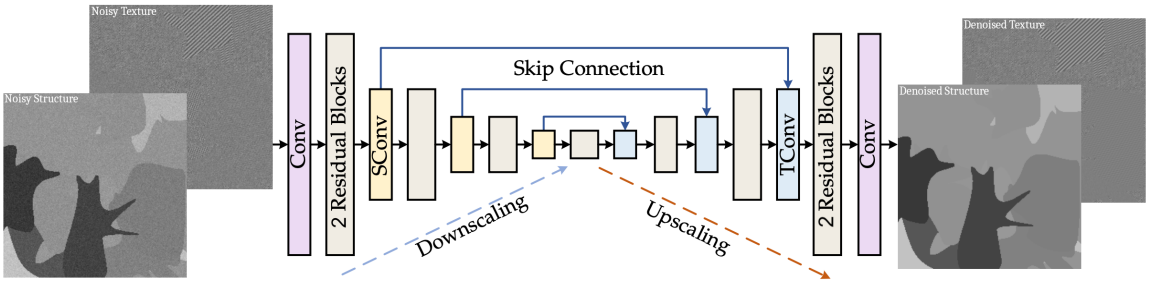


Figure 2: Architecture of the DRUNet denoiser [42] used to parametrize N. Contrarily to the initial implementation of the gradient step PnP, we do not use a noise level map and the structure/texture components are both set in an invidual channel.

3.2. Application to inverse problems. The structure and texture each provide an important perceptual information of the content in an image. With prior knowledge on the structure and texture components in the original image, we may use the regularization $R(u, v)$ in the applications to solve inverse problems of the form

$$b = \mathcal{M}x \quad \text{or} \quad b = \mathcal{M}x + \varepsilon, \tag{3.11}$$

where $\mathcal{M}$ is a linear operator, ε is a Gaussian white noise and b is the observation.

In the noiseless setting, given $\mathcal{M}$ (e.g a mask) and a corrupted observation $b = \mathcal{M}f$, we aim to recover f through solving the optimization problem

$$\min_{\mathrm{x}=(u,v)} R(\mathrm{x}) \quad \text{subject to } \mathcal{M}(u+v) = b \tag{3.12}$$

To solve this problem, we consider the convex set $\mathcal{C}_b(\mathcal{M}) = \{\mathrm{x} = (u,v) \mid \mathcal{M}(u+v) = b\}$ and we set

$$\mathcal{P}_{\mathcal{C}_b(\mathcal{M})}(\mathrm{x}) = \underset{\mathcal{M}(Id\ Id)\mathrm{y}=b}{\arg\min} \frac{1}{2}\|\mathrm{y}-\mathrm{x}\|_2^2. \tag{3.13}$$

Then, we solve the problem via a projected gradient descent

$$\begin{cases} \mathrm{z}_n = \mathrm{x}_n - \tau \nabla R(\mathrm{x}_n) \\ \mathrm{x}_{n+1} = \mathcal{P}_{\mathcal{C}_b(\mathcal{M})}(\mathrm{z}_{n+1}) \end{cases}. \tag{3.14}$$

The projection is given by the following Lemma.

Lemma 3.2. *Let $\mathcal{M}$ be a linear operator. The proximal operator of $\chi_{\mathcal{C}_b(\mathcal{M})}$ (the orthogonal projection onto $\mathcal{C}_b(\mathcal{M})$) for $\mathrm{x} = \begin{pmatrix} u \\ v \end{pmatrix}$ is given by*

$$\mathcal{P}_{\mathcal{C}_b(\mathcal{M})}(\mathrm{x}) := \mathrm{Prox}_{\chi_{\mathcal{C}_b(\mathcal{M})},\lambda}(\mathrm{x}) = (I - L^+L)x + L^+b \tag{3.15}$$

where $L = \mathcal{M}\begin{pmatrix} Id & Id \end{pmatrix}$ and L^+ is the pseudo-inverse of L.

Proof. Let $L = \mathcal{M}\begin{pmatrix} Id & Id \end{pmatrix}$, $\lambda > 0$ and $\mathrm{x} = (u,v) \in E^2$, we have

$$\begin{aligned} \mathrm{Prox}_{\chi_{\mathcal{C}_{f(b)}},\lambda}(\mathrm{x}) &= \underset{\mathrm{y}\in E^2}{\arg\min} \lambda\chi_{\mathcal{C}_f(b)}(\mathrm{y}) + \frac{1}{2}\|\mathrm{y}-\mathrm{x}\|_2^2 \\ &= \underset{\substack{\mathrm{y}\in E^2 \\ L\mathrm{y}=b}}{\arg\min} \frac{1}{2}\|\mathrm{y}-\mathrm{x}\|_2^2. \end{aligned} \tag{3.16}$$

We have that $L\mathrm{y} = b$ is equivalent to $y = L^+b + w$ where $w = \ker(L)$. Hence we minimize $\min_{w\in\ker L} \|w - \mathrm{x} + L^+b\|_2^2$. The solution of this least squares problem is the definition of the orthogonal projection of $\mathrm{x} - L^+b$ on $\ker L$. The orthogonal projection on $\ker L$ is given by $I - L^+L$ and, as $L^+LL^+ = L^+$, we have

$$\mathrm{Prox}_{\chi_{\mathcal{C}_f},\lambda}(\mathrm{x}) = (I - L^+L)x - (I - L^+L)L^+b + L^+b = (I - L^+L)x + L^+b. \tag{3.17}$$ ∎

For example, when $\mathcal{M}$ is a mask operator, i.e the inpainting task (see Section 5.2), we find that its associated projection operator is given by

$$\mathcal{P}_{\mathcal{C}_b(\mathcal{M})}(u,v) = \begin{pmatrix} u \\ v \end{pmatrix} + \frac{1}{2}\begin{pmatrix} \mathcal{M}(b-u-v) \\ \mathcal{M}(b-u-v) \end{pmatrix}. \tag{3.18}$$

Similarly, if we consider the inverse problem with noise, we aim to recover f through the optimization problem

$$\min_{\mathrm{x}=(u,v)} R(\mathrm{x}) + \frac{\mu}{2} \|\mathcal{M}(u+v) - b\|_2^2. \tag{3.19}$$

As $(u,v) \mapsto \|\mathcal{M}(u+v)-b\|_2^2$ is differentiable, this can be solved using a gradient descent scheme

$$\mathrm{x}_{n+1} = \mathrm{x}_n - \tau \nabla R(\mathrm{x}_n) - \tau\mu \begin{pmatrix} Id \\ Id \end{pmatrix} \mathcal{M}^T \left(\mathcal{M} \begin{pmatrix} Id & Id \end{pmatrix} \mathrm{x}_n - b \right) \tag{3.20}$$

3.3. Projection Tuning parameter for the joint structure-texture model. One of the constraints given by considering a single regularization for both the structure and texture is that we lose any type of control on the given result. Because we are in the setting of exact decomposition, we do not have any tuning parameter. While it is often advantageous to have little to no tuning in a decomposition method, we introduce a method to balance the structure/texture output through the projection operation $\mathcal{P}_{\mathcal{C}_f}$.
Essentially, the projection $\mathcal{P}_{\mathcal{C}_f}$ equally adds the residual of the output of the denoiser into both the structure and texture components in order for the result to fit the equation $f = u+v$. However, depending on the residual one may wish to add more or less of the residual to either the structure or texture component. Hence, we may consider the non-orthogonal projection instead

$$\widetilde{\mathcal{P}}_{\mathcal{C}_f}((u,v)^T, \mu) = \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} \mu(f-u-v) \\ (1-\mu)(f-u-v) \end{pmatrix}, \tag{3.21}$$

where $\mu \in (0,1)$ is a tuning parameter. Setting μ low will import less of the remaining texture from the residual into the structure and a high μ will import less of the remaining structure contained in the residual into the texture (see Figure 3).

3.4. Adaptive step selection. The projected gradient step descent denoiser has the downside to be non-convex and usual convex techniques may fail. To handle this, there are multiple ways we may approach to stabilize the gradient descent:

1. Initialization near the true solution, e.g using another decomposition scheme to initialize x_0 or a trained neural network that directly does a first decomposition.
2. Backtracking methods as it was originally implemented in [17],
3. Regularization search: at each iteration, we perform a line search in order to set an optimal gradient step τ and projection tuning parameter μ that minimizes the most the regularization function R_{x}.

We found that this last approach with the simple initialization $\mathrm{x}_0 = (f, 0)$ leads to the best recovery result for synthetic images. In a second step, we may also decrease the gradient step τ_n in order to enforce $\|\mathrm{x}_n - \mathrm{y}_n\| \to 0$.
The parameter search does not impact much the speed of the projected gradient step algorithm as the computational cost of $R(\mathrm{x})$ is low when compared to $\nabla R(\mathrm{x})$. Moreover, the rate of convergence is greatly increased, so less iterations are required to reach an optimal output.

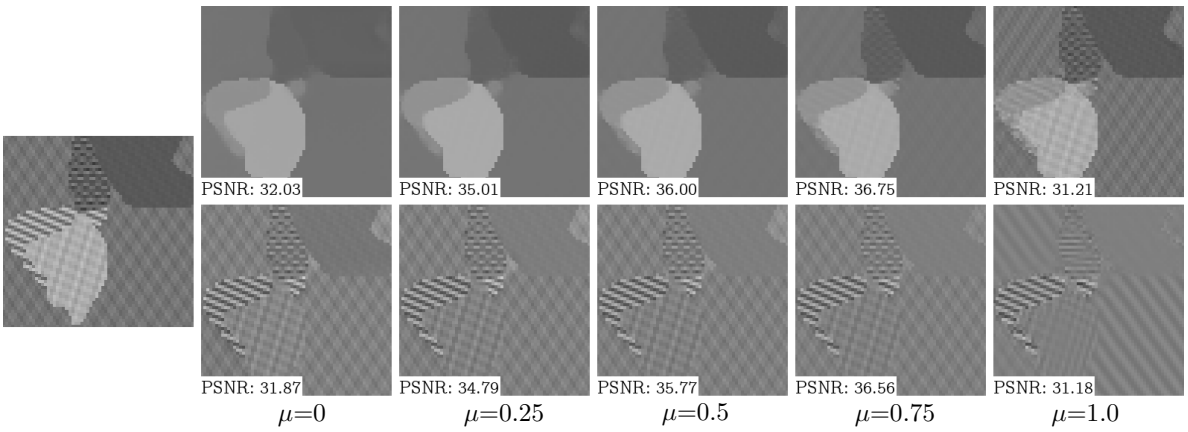


Figure 3: Illustration of the projection direction μ on a synthetic image. At the far left: original image f, top: structure component u_μ, bottom: texture component v_μ. For each decomposition, the first two iterations of the projected gradient descent were obtained using μ=0.5 in order to obtain an initial decomposition and the following iterations were obtained using the indicated projection direction μ. Setting μ low will reinforce the structure model and setting μ high will reinforce the texture model. The PSNR with respect to the ground truth decomposition is given at the bottom left of each image.

Algorithm 3.2 Joint structure-texture projected gradient descent with optimal regularization line search

Init.: $\mathrm{x}_0 = (f, 0)$,
Input f
Output: The output structure and texture $\hat{\mathrm{x}} = (\hat{u}, \hat{v})$
while not converged **do**
$\tau_k = \underset{\tau \in \mathbb{R}_+}{\arg\min}\ R((1-\tau)\mathrm{x}_k + \tau D(\mathrm{x}_k))$
$\mathrm{y}_{k+1} = (1-\tau_k)\mathrm{x}_k + \tau_k D(\mathrm{x}_k)$
$\mu_k = \underset{\mu \in \mathbb{R}}{\arg\min}\ R(\widetilde{\mathcal{P}}_{\mathcal{C}_f}(\mathrm{y}_{k+1}, \mu))$
$\mathrm{x}_{k+1} = \mathcal{P}_{\mathcal{C}_f}(\mathrm{y}_{k+1}, \mu_k)$
end while

4. Synthetic images as a regularity prior.

4.1. Background. Inherently, we wish that our neural network learns the low-dimensional joint structure-texture model $\Sigma_{s \otimes t}$. One of the core dilemmas with associating machine learning methods to the image decomposition problem is the absence of ground truth (especially with natural images). In order to train the denoiser, we designed a procedure that generates random piecewise constant images (with a connected support) and an associated texture component, generated from a texture model. This enabled us to train the denoiser with an endless supply of training examples, without any ambiguity on the ground truth. In [1], the

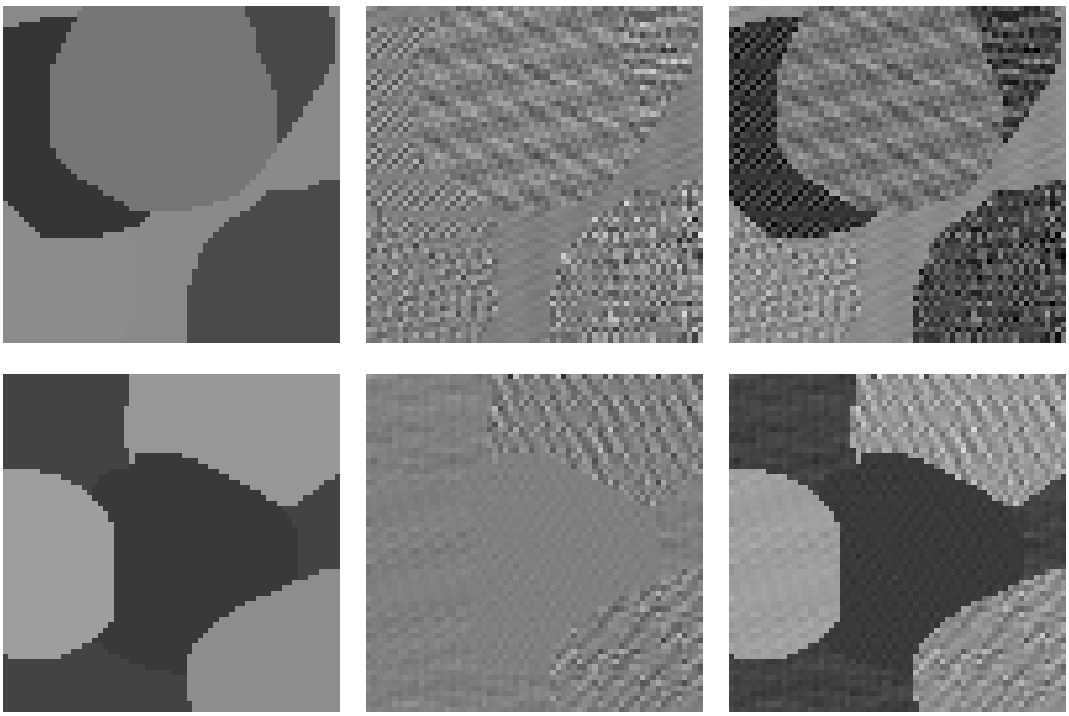


Figure 4: Examples of generated structure (left) and texture (center) images used in the numerical experiments and to train the different neural networks. On the right, we show their sum.

authors used a similar approach where they trained a denoiser on the dead leaves synthetic image model and demonstrated that it could reach near-optimal results by training only upon synthetic images. The synthetic joint structure-texture image model that we propose follows the same construction: we generate a synthetic image by superposing random shapes with an additional texture. However, contrarily to the dead leaves generation, we heavily limit the number of superposed shapes as the associated textures should remain small locally.

4.2. Database design. In order to create a connected support, we proceeded in two steps. First we produce a connected support via the Lane-Riesenfeld algorithm [20], where we randomly scatter initial points around an origin ((pos_x, pos_y) in Algoritm 4.1) and we recursively apply a subdivision process (split + average) to those points until we obtain a smooth curve (Figure 5).

Given the ordered points $\mathrm{P} = \{p_1, ..., p_k\} \subset \mathbb{R}^2$, we define the splitting and averaging procedures as

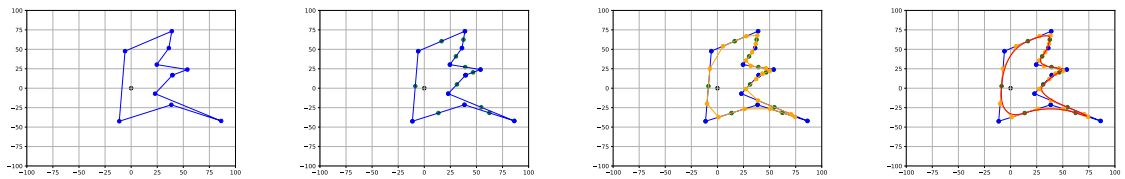
- $\mathbf{split}(\mathrm{P}) = \{p_1, \frac{p_1+p_2}{2}, p_2, \frac{p_2+p_3}{2}, ..., p_k, \frac{p_k+p_1}{2}\}$,
- $\mathbf{average}(\mathrm{P}) = \{\frac{p_1+p_2}{2}, ..., \frac{p_{k-1}+p_k}{2}, \frac{p_k+p_1}{2}\}$.

Note that in the averaging step, other weights may be used. Using any weights taken from a line in Pascal's triangle will lead the points to converge to a smooth curve. Once the contour of the shape is generated, we may fill it using a flood fill algorithm. In full summary, single

connected support is generated as follows:

1. Randomly select k (ordered) points around a central point $c \in \mathbb{N}^2$, $\mathrm{P}_0 = \{p_1, \dots p_k\}$.
2. Subdivide the points, $\mathrm{P}_{n+1} = \textbf{average}(\textbf{split})(\mathrm{P}_n)$, until a smooth enough set of points is achieved.
3. Project the resulting points onto a canvas and use a flood fill algorithm to obtain the image support.

To generate the final structural component, we randomly scatter the aforementioned generated shapes onto a canvas with varying level of intensity.



(a) (b) (c) (d)

Figure 5: Illustration of the subdivison process (a) Initial point scatter, (b) Splitting step, (c) averaging step, (d) Final shape (in red) after ten subdivisions.

Algorithm 4.1 Synthetic image generation

Param.: K, t_{min}, t_{max}, s_{min}, s_{max}
Output: S, T (the synthetic structure and texture components)
$S = ones(n, m)$
$T = \text{generate\_texture}(n, m)$
for i in $[0, ..., K-1]$ **do**
  $pox_x, pox_y = \text{randint}(0, n), \text{randint}(0, m)$
  $\Omega = \text{generate\_support}(pos_x, pos_y)$
  $\alpha_s, \alpha_t = \text{uniform}(s_{min}, s_{max}), \text{uniform}(t_{min}, t_{max})$
  $S_{|\Omega} = \alpha_s \cdot \mathbb{1}_\Omega$
  $T_{|\Omega} = \alpha_t \cdot \mathbb{1}_\Omega \cdot \text{generate\_random\_texture}()$
end for

The textural component is much more straightforward to generate. In the literature there have been multiple texture models that have been proposed, e.g low-patch rank [31], low-rank [41, 43], sparse dictionary [25], etc... Using random distributions, we generate textures from these models, which are then cropped to fit its corresponding support. We provide an example of the sparse Fourier texture generation in Algorithm 4.2.

4.3. A tool to build an optimal regularization. Up to our knowledge, every image decomposition model has relied upon a regularization of the form $\lambda R_s(u) + R_t(v)$. As discussed in Section 2.3, while this scheme is optimal when we consider the two components to be uncorrelated, it is not the case otherwise. We show evidence that this is in fact suboptimal

Algorithm 4.2 Random sparse Fourier texture generation

Param.: $freq^x_{min}$, $freq^y_{min}$, s_{max},
Output: T
$s = \text{randint}(1, s_{max})$
$T_{freq} = zeros(n, m)$
for i in $[0, ..., s_{max}]$ **do**
  $x_{freq} = \text{randint}(freq^x_{min}, n - freq^x_{min})$
  $y_{freq} = \text{randint}(freq^y_{min}, m - freq^y_{min})$
  $\hat{T}[x_{freq}, y_{freq}] = 1$
  $\hat{T}[-x_{freq}, -y_{freq}] = 1$
end for
$T = ifft(T)$

in the case of the joint structure-texture model, and that a regularization which takes both structure and texture as inputs leads to a better result. This further supports our main hypothesis that the interaction between the structure and the texture components provides an invaluable information to perform an efficient separation.
In our experiments, we selected the DRUNet architecture (see Figure 2) in order to parametrize the neural network N in (3.3), and we set the texture model Σ_t to a sparse model in the high frequencies (superposition of cosines/sines). We trained three separate denoisers:

- $D_{\mathrm{x}} = Id - \nabla R_{\mathrm{x}}$ which is trained on denoising structure-texture couples $\mathrm{x} = (u, v)$
- $D_s = Id - \nabla R_s$ which is trained on denoising only the structure.
- $D_t = Id - \nabla R_t$ which is trained on denoising only the texture.

In terms of denoising performance, we observe that D_{x} slightly outperforms D_s and D_t in both structure and texture performance (Table 1). Unsurprisingly, there is a large performance gap between the structural and textural components for the task of Gaussian noise removal. Piecewise constant images are possibly the easiest image category to denoise, whereas textures are oppositely the most difficult ones. Diverging from the rest, we observed that D_s has an exceptionally high fixed point PSNR ($\sigma = 0$), indicating that the underlying structure space should lie near the minimizer of R_s.

While D_{x} is able to achieve similar denoising performance to D_s and D_t for both structure and texture components respectively, our experiments show that D_{x} is superior in the application of image decomposition. Given a dataset of 1000 synthetic images, we chose for each image a tuning parameter λ for the minimization of $\lambda R_s + R_t$ that maximizes the PSNR with respect to the ground truth. Even with this harsh condition in favor of the separated models, the joint structure-texture model algorithm has ability to better recover the decomposition into structure and texture (Table 2), with no user input for parameter tuning.

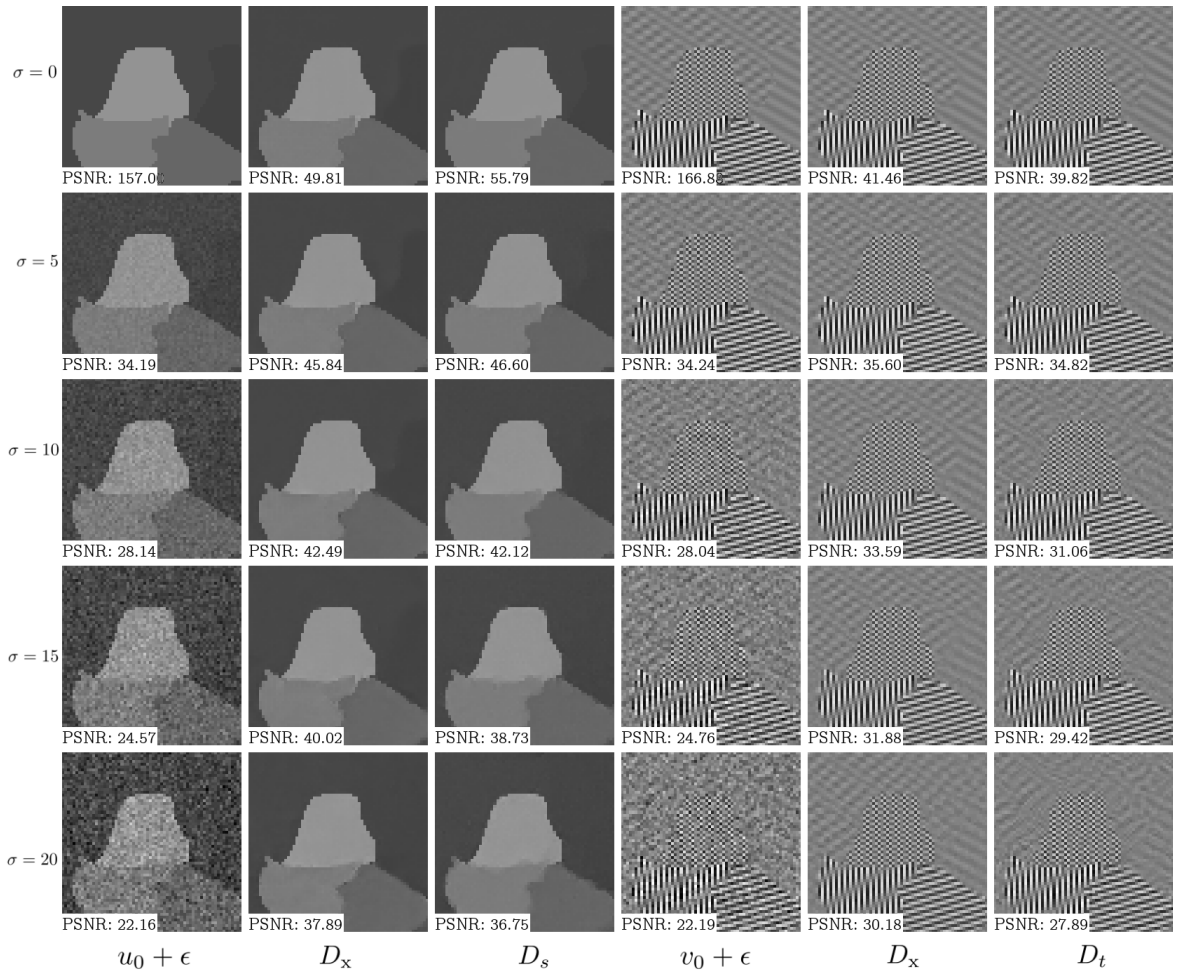


Figure 6: Denoising of a synthetic structure and texture with the different methods and different noise levels using a denoiser D_{x} that takes both structure and texture as input and D_s, D_t that takes only one component (structure and texture respectively). The results are close for low-level noise, however for high level noise D_{x} performs much better, especially on the texture recovery. The PSNR with respect to the ground truth is shown at the bottom left of the images.

	mean PSNR (best iteration)	mean PSNR (20 iterations)
R_{x}	**42.69**	**40.85**
$\lambda R_s + R_t$	40.12	38.83

Table 2: Comparison between joint and separated (R_{x} and $\lambda R_s + R_t$) regularization minimization for image decomposition, on a test set of 100 images. We used the line search method for R_{x}, and with an initialization with the LPR algorithm for $\lambda R_s + R_t$ and an optimal choice of λ. We find that the joint structure-texture modeling performs better than the separated one. We present the best PSNR out of 100 iterations and the PSNR at 20 iterations (around when the algorithm should achieve the best result).

	$\sigma(./255)$	0	5	10	15	20
(Structure)	D_{x}	49.50	**47.03**	**43.88**	**40.99**	**38.42**
	D_s	**55.07**	46.08	43.27	40.46	38.24
(Texture)	D_{x}	**44.96**	**36.22**	**32.24**	**30.04**	**28.52**
	D_t	39.51	35.18	31.39	29.31	27.90

Table 1: Mean PSNR denoising performance comparison between the joint and separated structure-texture denoisers, on a test set of 1000 generated 64×64 synthetic structure-texture images. While the denoising performance is similar for noise with a small standard deviation, denoising both components at the same time provides better denoising capability for both structure and texture.

5. Experiments.

5.1. Synthetic image decomposition. As discussed in Section 4.3, we compared the decompositions between the two regularization schemes $R(u, v)$ and $R(u) + R(v)$. As we can observe in Figure 7, even for images where the PSNR was close between the two decompositions, the joint structure-texture approach was able to better separate the two models. For example, in the second image, while the structure components for each approach have similar PSNR with respect to the ground truth, there is less texture present in the structure with the joint structure-texture method. Finally, the decomposition using the joint model converges very quickly to an appropriate point, needing less than 10 iterations to reach an optimal value (Fig. 8).

5.2. Inpainting. The task of inpainting large holes is very ill-posed and thus necessitates a prior knowledge in order to achieve a satisfactory recovery. As presented in [4], image decomposition modeling can be used to inpaint simultaneously both structure and texture. In the case of missing pixels in an image, we found the initialization of the projected gradient algorithm to be of utmost importance in order to recover correctly both the structure and the texture. If initialization is incorrectly set, the masked areas may be considered as providing structure. We found that filling the missing regions with an average onion-peel filling (iteratively filling the holes one layer at a time by taking the average of the surrounding pixels) provided an adequate initialization. In our experiments (Figure 9) on synthetic images we observe a perfect recovery of the textures present in the image and with an appropriate structure recovery (note that there is no way to recover the correct boundary in the masked areas). This indicates that the denoising task was able to successfully learn the texture model it was trained on.

5.3. Natural image decomposition. Using a denoiser D_{x} trained on 64×64 synthetic structure-texture image we decomposed natural images patchwise using an overlap of 16 (and a patchsize of 64×64). Moreover, we used a line search (as presented in Section (3.4)) at every iteration in order to select an optimal gradient descent parameter. We set the structure model Σ_s as piecewise constant images and the texture model Σ_t as the combination of sparse Fourier textures and low-patch rank. We stress that each decomposition reached in each case

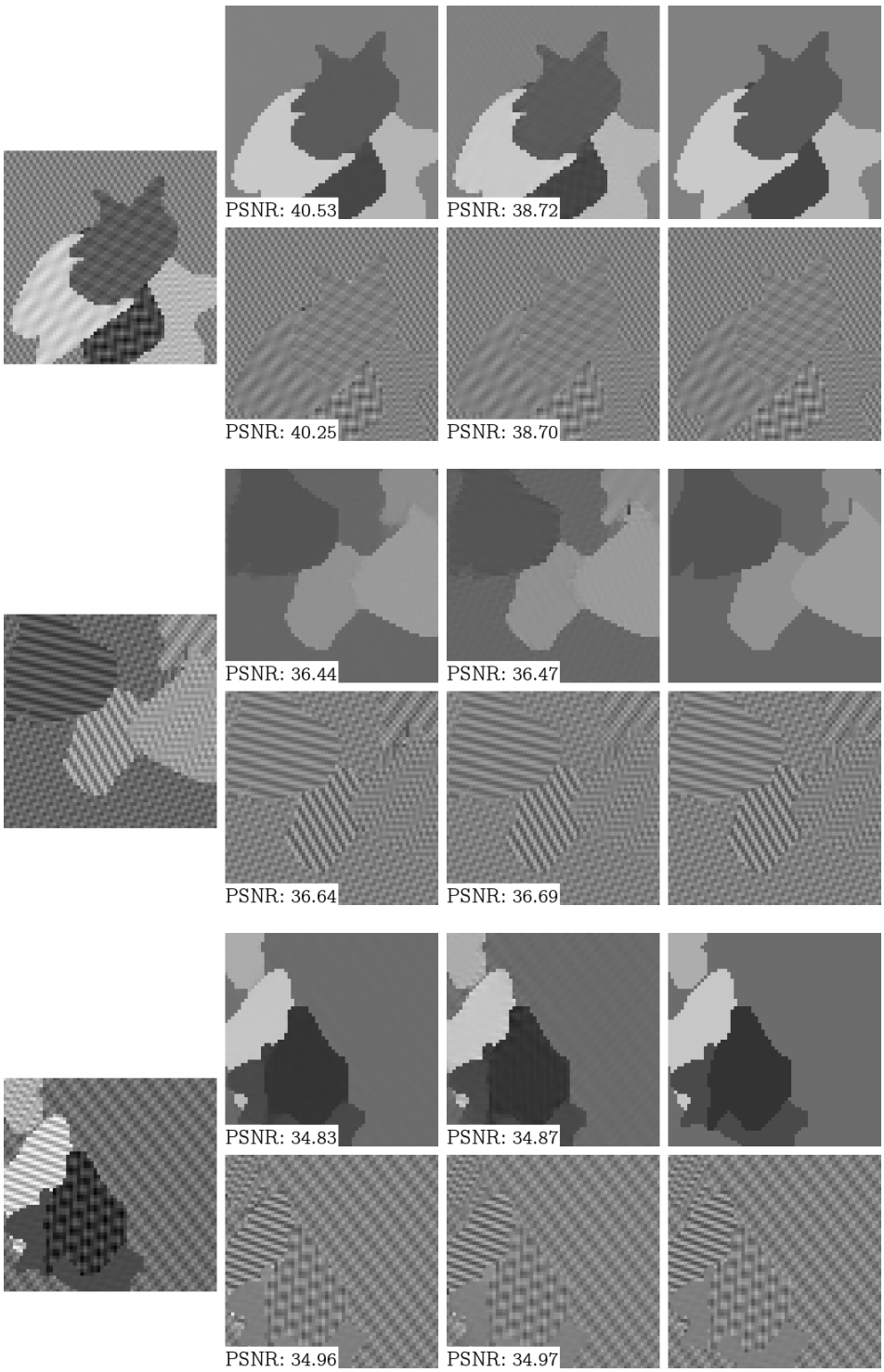


Figure 7: Comparison between the decompositions given by $R(u, v)$ and $R_s(u) + R_t(v)$ minimization. From left to right: original image, output from $R(u, v)$, output from $\lambda R(u) + R(v)$ and the target decomposition (u_0, v_0). In order to avoid cherry picking bias, the decompositions were selected with a small PSNR difference between each other. We observe that the regularization $R(u, v)$, trained on both component simultaneously is able to better fit the low dimensional models it was trained on. This demonstrates that the shared information between the two component is useful for the regularization in separating the two components. The PSNR with respect to the ground truth is shown at the bottom left of the images.

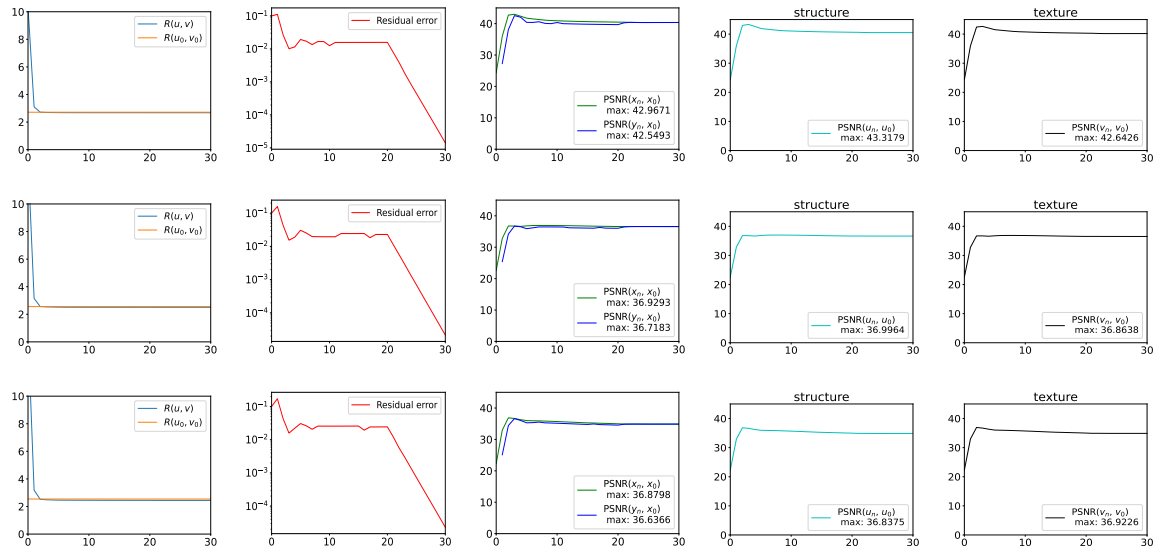


Figure 8: Regularization plots associated with the image decompositions of Fig 7. a) Regularization function R_{x}, b) Residual error $\frac{\left\|\mathrm{y}_n^1+\mathrm{y}_n^2-f\right\|_2}{\|f\|_2}$ in log scale, c) PSNR error with respect to the ground truth x_0 for y_n (blue curve) and x_n (green curve), d-e) PSNR error with respect to the cartoon/texture components respectively. In less than 10 iterations the algorithm converges to its optimal value, with only a slight dip in the PSNR plot. The residual error (The normalized error of y_n from $\mathcal{C}_f$) tends to zero in the last iterations as we half τ between each iterations.

was performed using no tuning parameter or manual input. We evaluated our algorithm on real images (Figure 10) and observed that the model, while trained only on synthetic images was able to generalize well to natural images.

We performed some decomposition on satellite images taken from the MLSRNet dataset [27]. As the images are noisy, we performed decomposition with a residual, i.e we do not use the projection $\mathcal{P}_{\mathcal{C}_f}$ in the last iteration. As the original measured image is noisy, this removes some of the noise present in the original image from the decomposition as it belong to neither the structure or texture models. However, we observed that this also extracts some features in the image such as the central road lines for the same reasons.

5.4. Towards natural image inpainting. In the context of natural image inpainting, we found that if the texture is close to the learned low dimensional model, we are able to appropriately inpaint the masked regions in the image (Figure 12). The mask shape is not visible in the reconstructed image. These preliminary results are encouraging for the design of inpainting methods (and more generally methods to solve inverse imaging problems) based on deep neural network architectures with a fully controlled low dimensional prior using a synthetic

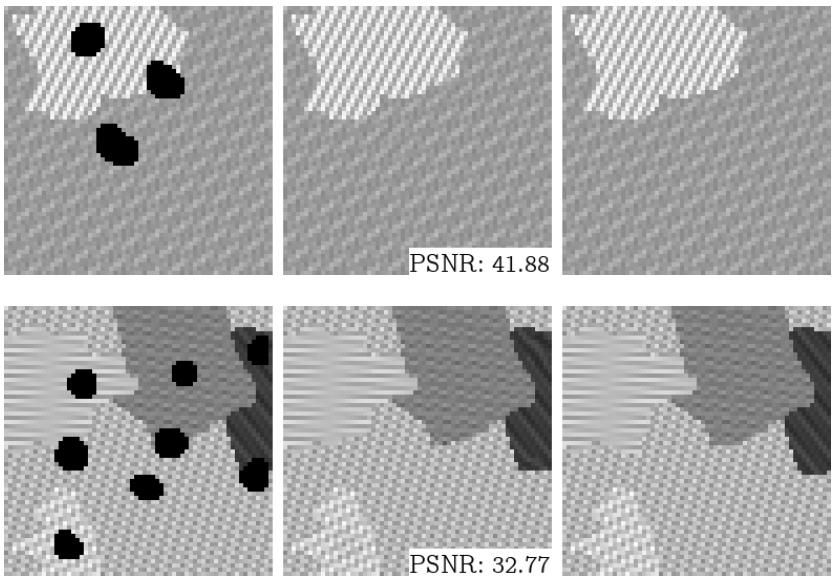


Figure 9: Inpainting recovery on synthetic images. From left to right: input masked image, reconstruction, original image. While the holes are relatively large, the regularization is able to recover well the different textures in the images. The PSNR with respect to the ground truth is shown at the bottom right of the images.

database.

6. Discussion. The joint structure-texture model and plug and play scheme trained using a synthetic dataset we have introduced is general and highly adaptable. Essentially, as long as we can generate data that fit the low dimensional models, we may learn a regularization function that can perform the decomposition. Furthermore, our research indicates that the learned regularization through denoising random synthetic data is able to learn effectively different low-dimensional models based on sparsity and low-rank. These last two decades, theoretical results were obtained that guaranteed (or not) recovery under certain conditions for different regularization functions associated with low dimensional models [13]. Learned regularization of low dimensional models as we introduced in this paper could be explored further in this context to solve various inverse problems.

Here, we have limited our area of study to piecewise constant structures and sparse Fourier and low patch rank textures. Other structure/texture models such as piecewise continuous structures and dictionary sparse textures could be investigated. Moreover, the texture can be learned on a mixture of different models. Even more broadly, our scheme allows a more abstract definition of texture such as learning the regularization using a dataset of textures [19]. Extensions of the two-component decomposition such as the jump-oscillation-trend [9] or cartoon-smooth-texture [14] could also be investigated in the future using the same process we have introduced here.

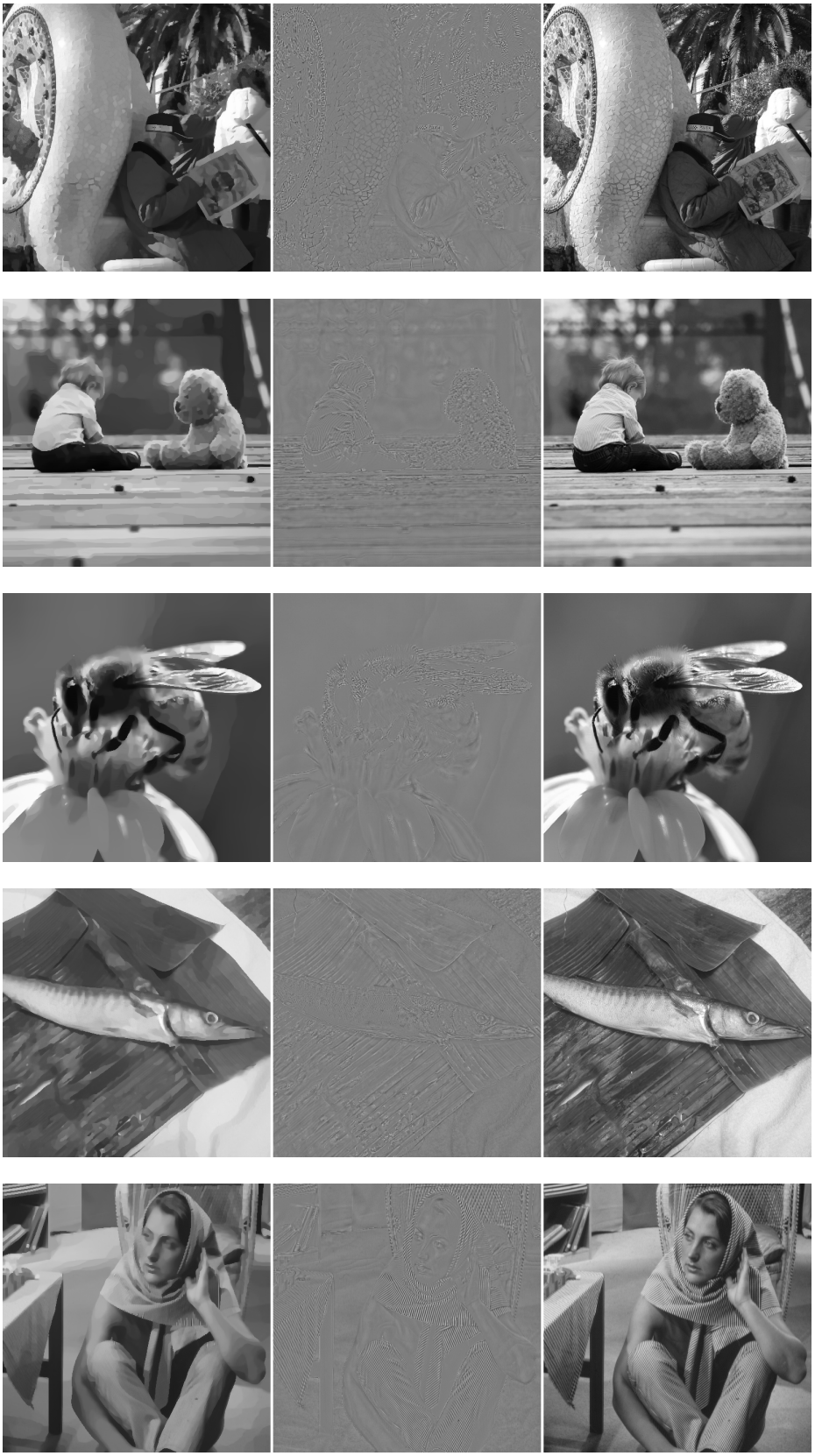


Figure 10: Natural image decomposition using the joint structure-texture model, using a projected gradient descent with line search. From left to right: structure, texture, original image.

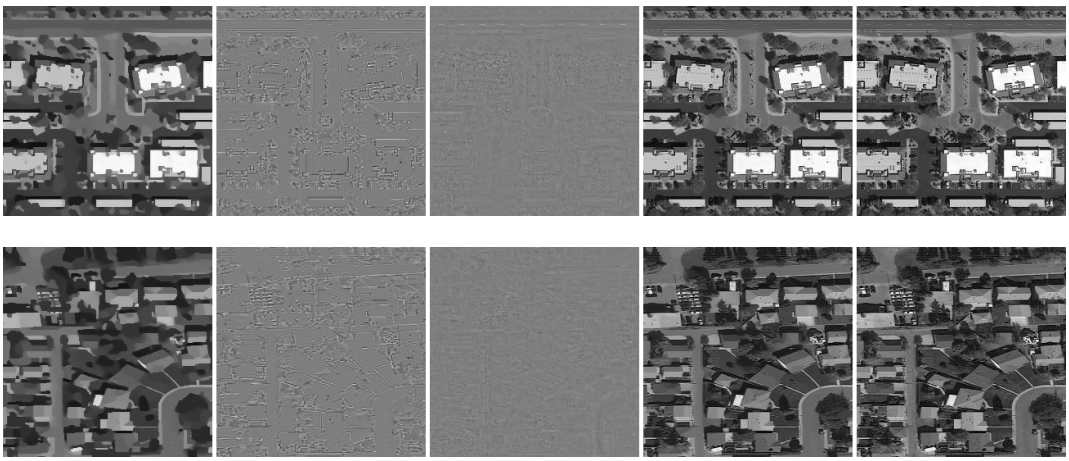


Figure 11: Satellite image decomposition with a residual. From left to right: structure, texture, residual $f - u - v$, denoised image $u + v$, original image.

Figure 12: Inpainting experiment on the Torsilyo image. From left to right: masked image, recovered image, original image. We observe that the masked regions on the scales of the fish are well recovered as the textures are close to the learned texture low dimensional model (sparse fourier texture/low patch rank).

Alternative PnP methods to the gradient step denoiser [17] should also be considered to accelerate the training and iterations in the optimization algorithm. While the gradient step denoiser is robust and performs well, the computation of $\nabla R(x)$ via autograd has a high computation and GPU memory cost for both training and inference.

7. Acknowledgments. Experiments presented in this paper were carried out using the PlaFRIM experimental testbed, supported by Inria, CNRS (LABRI and IMB), Université de Bordeaux, Bordeaux INP and Conseil Régional d'Aquitaine (see https://www.plafrim.fr). This work was supported by the French National Research Agency (ANR) under reference ANR-20-CE40-0001 (EFFIREG project), and by PEPR PDE_AI.

REFERENCES

[1] R. Achddou, Y. Gousseau, and S. Ladjal, *Synthetic images as a regularity prior for image restoration neural networks*, in International Conference on Scale Space and Variational Methods in Computer Vision, Springer, 2021, pp. 333–345.

[2] J.-F. Aujol, G. Aubert, L. Blanc-Féraud, and A. Chambolle, *Image decomposition into a bounded variation component and an oscillating component*, Journal of Mathematical Imaging and Vision, 22 (2005), pp. 71–88.

[3] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher, *Structure-texture image decomposition—modeling, algorithms, and parameter selection*, International journal of computer vision, 67 (2006), pp. 111–136.

[4] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher, *Simultaneous structure and texture image inpainting*, IEEE transactions on image processing, 12 (2003), pp. 882–889.

[5] D. Bertsekas, A. Nedic, and A. Ozdaglar, *Convex analysis and optimization*, vol. 1, Athena Scientific, 2003.

[6] A. Bourrier, M. E. Davies, T. Peleg, P. Pérez, and R. Gribonval, *Fundamental performance limits for ideal decoders in high-dimensional linear inverse problems*, IEEE Transactions on Information Theory, 60 (2014), pp. 7928–7946.

[7] K. Bredies, K. Kunisch, and T. Pock, *Total generalized variation*, SIAM Journal on Imaging Sciences, 3 (2010), pp. 492–526.

[8] V. Caselles, A. Chambolle, and M. Novaga, *The discontinuity set of solutions of the tv denoising problem and some extensions*, Multiscale modeling & simulation, 6 (2007), pp. 879–894.

[9] A. Cicone, M. Huska, S.-H. Kang, and S. Morigi, *Jot: a variational signal decomposition into jump, oscillation and trend*, IEEE Transactions on Signal Processing, 70 (2022), pp. 772–784.

[10] M. J. Fadili, J.-L. Starck, J. Bobin, and Y. Moudden, *Image decomposition and separation using sparse representations: An overview*, Proceedings of the IEEE, 98 (2009), pp. 983–994.

[11] Y. Fang, H. Fan, L. Sun, Y. Guo, and Z. Ma, *From tv-l 1 to gated recurrent nets*, in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2019, pp. 2212–2216.

[12] S. Foucart, R. Gribonval, L. Jacques, and H. Rauhut, *Jointly low-rank and bisparse recovery: Questions and partial answers*, Analysis and Applications, 18 (2020), pp. 25–48.

[13] S. Foucart and H. Rauhut, *An Invitation to Compressive Sensing*, Springer New York, New York, NY, 2013, pp. 1–39.

[14] L. Girometti, M. Huska, A. Lanza, and S. Morigi, *Quaternary image decomposition with cross-correlation-based multi-parameter selection*, in International Conference on Scale Space and Variational Methods in Computer Vision, Springer, 2023, pp. 120–133.

[15] A. Goujon, S. Neumayer, and M. Unser, *Learning weakly convex regularizers for convergent image-reconstruction algorithms*, SIAM Journal on Imaging Sciences, 17 (2024), pp. 91–115.

[16] A. Guennec, J. Aujol, and Y. Traonmilin, *Adaptive parameter selection for gradient-sparse + low patch-rank recovery: application to image decomposition*, (2023).

[17] S. Hurault, A. Leclaire, and N. Papadakis, *Gradient step denoiser for convergent plug-and-play*, in International Conference on Learning Representations (ICLR'22), 2022.

[18] Y. Kim, B. Ham, M. N. Do, and K. Sohn, *Structure-texture image decomposition using deep variational priors*, IEEE Transactions on Image Processing, 28 (2018), pp. 2692–2704.

[19] G. Kylberg, *Kylberg texture dataset v. 1.0*, Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala university, 2011.

[20] J. M. Lane and R. F. Riesenfeld, *A theoretical development for the computer generation and display of piecewise polynomial surfaces*, IEEE Transactions on Pattern Analysis and Machine Intelligence, (1980), pp. 35–46.

[21] K. Lu, S. You, and N. Barnes, *Deep texture and structure aware filtering network for image smoothing*, in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 217–233.

[22] Y. Meyer, *Oscillating patterns in image processing and nonlinear evolution equations: the fifteenth Dean Jacqueline B. Lewis memorial lectures*, vol. 22, American Mathematical Soc., 2001.

[23] S. Ono, T. Miyata, and I. Yamada, *Cartoon-texture image decomposition using blockwise low-rank texture characterization*, IEEE Transactions on Image Processing, 23 (2014), pp. 1128–1142.

[24] S. Oymak, A. Jalali, M. Fazel, Y. C. Eldar, and B. Hassibi, *Simultaneously structured models with application to sparse and low-rank matrices*, IEEE Transactions on Information Theory, 61 (2015), pp. 2886–2908.
[25] G. Peyré, *Sparse modeling of textures*, Journal of mathematical imaging and vision, 34 (2009), pp. 17–31.
[26] J. Prost, A. Houdard, A. Almansa, and N. Papadakis, *Learning local regularization for variational image restoration*, in International Conference on Scale Space and Variational Methods in Computer Vision, Springer, 2021, pp. 358–370.
[27] X. Qi, P. Zhu, Y. Wang, L. Zhang, J. Peng, M. Wu, J. Chen, X. Zhao, N. Zang, and P. T. Mathiopoulos, *Mlrsnet: A multi-label high spatial resolution remote sensing dataset for semantic scene understanding*, ISPRS Journal of Photogrammetry and Remote Sensing, 169 (2020), pp. 337–350.
[28] E. T. Reehorst and P. Schniter, *Regularization by denoising: Clarifications and new interpretations*, IEEE transactions on computational imaging, 5 (2019), p. 52.
[29] Y. Romano, M. Elad, and P. Milanfar, *The little engine that could: Regularization by denoising (red)*, SIAM Journal on Imaging Sciences, 10 (2017), pp. 1804–1844.
[30] L. I. Rudin, S. Osher, and E. Fatemi, *Nonlinear total variation based noise removal algorithms*, Physica D: nonlinear phenomena, 60 (1992), pp. 259–268.
[31] H. Schaeffer and S. Osher, *A low patch-rank interpretation of texture*, SIAM Journal on Imaging Sciences, 6 (2013), pp. 226–262.
[32] W. Shang, G. Liu, Y. Wang, J. Wang, and Y. Ma, *A non-convex low-rank image decomposition model via unsupervised network*, Signal Processing, 223 (2024), p. 109572.
[33] B. Shi, W. Xu, and X. Yang, *Ctdnet: cartoon-texture decomposition-based gray image super-resolution network with multiple degradations*, JOSA B, 40 (2023), pp. 3284–3290.
[34] J.-L. Starck, M. Elad, and D. L. Donoho, *Image decomposition via the combination of sparse representations and a variational approach*, IEEE transactions on image processing, 14 (2005), pp. 1570–1582.
[35] Y. Traonmilin, R. Gribonval, and S. Vaiter, *A theory of optimal convex regularization for low-dimensional recovery*, Information and Inference, A journal of the IMA, (2024).
[36] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, *Plug-and-play priors for model based reconstruction*, in 2013 IEEE global conference on signal and information processing, IEEE, 2013, pp. 945–948.
[37] L. A. Vese and S. J. Osher, *Modeling textures with total variation minimization and oscillating patterns in image processing*, Journal of scientific computing, 19 (2003), pp. 553–572.
[38] L. Xu, Q. Yan, Y. Xia, and J. Jia, *Structure extraction from texture via relative total variation*, ACM transactions on graphics (TOG), 31 (2012), pp. 1–10.
[39] R. Xu, Y. Xu, Y. Quan, and H. Ji, *Cartoon-texture image decomposition using orientation characteristics in patch recurrence*, SIAM Journal on Imaging Sciences, 13 (2020), pp. 1179–1210.
[40] W. Xu, C. Tang, Y. Su, B. Li, and Z. Lei, *Image decomposition model shearlet–hilbert–l 2 with better performance for denoising in espi fringe patterns*, Applied Optics, 57 (2018), pp. 861–871.
[41] H. Zhang and V. M. Patel, *Convolutional sparse and low-rank coding-based image decomposition*, IEEE Transactions on Image Processing, 27 (2017), pp. 2121–2133.
[42] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, *Plug-and-play image restoration with deep denoiser prior*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 44 (2021), pp. 6360–6376.
[43] Z. Zhang and H. He, *A customized low-rank prior model for structured cartoon–texture image decomposition*, Signal Processing: Image Communication, 96 (2021), p. 116308.
[44] C. Zheng, D. Shi, and W. Shi, *Adaptive unfolding total variation network for low-light image enhancement*, in Proceedings of the IEEE/CVF international conference on computer vision, 2021, pp. 4439–4448.
[45] F. Zhou, Q. Chen, B. Liu, and G. Qiu, *Structure and texture-aware image decomposition via training a neural network*, IEEE Transactions on Image Processing, 29 (2019), pp. 3458–3473.