



HAL
open science

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

Antoine Guennec, Jean-François Aujol, Yann Traonmilin

► **To cite this version:**

Antoine Guennec, Jean-François Aujol, Yann Traonmilin. Joint structure-texture low dimensional modeling for image decomposition with a plug and play framework. 2024. hal-04648963v2

HAL Id: hal-04648963

<https://hal.science/hal-04648963v2>

Preprint submitted on 4 Nov 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

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

$$(1.1) \quad f = u + v$$

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

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

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.fr).

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

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

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

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

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

64 **1.1. The plug and play framework.** A recent advance in the field of inverse problems has
 65 been the introduction of the plug-and-play (PnP) framework [36]. Inverse problems are often
 66 solved via the minimization scheme

$$67 \quad (1.3) \quad \underset{x \in E}{\text{minimize}} \quad R(x) + F(x, y),$$

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

71 The PnP method leverages proximal splitting algorithms, established initially for convex
 72 problems, by substituting the traditional proximal operator $\text{Prox}_{R,\eta}(x)$ with a denoiser $D(x)$.
 73 In this context, associating a denoiser with a regularization function is not straightforward if
 74 we wish to obtain convergence properties. First initiated in [29], it was proposed to construct
 75 an explicit regularization function from a denoiser. However, given a differentiable denoiser
 76 $D : \mathbb{R}^N \rightarrow \mathbb{R}^N$, it was later proven in [28] that the desirable property

$$77 \quad (1.4) \quad \nabla R = Id - D,$$

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 $N : E \rightarrow E$ 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)

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

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

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

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

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

- 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 modeling 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.

115 **1.3. Related Work.** The first structure-texture decomposition models relied on varia-
 116 tional methods, using the total variation to characterize the structural component and a
 117 function space norm to constrain the texture component, such as the L^2 -norm [30], G -norm
 118 [22, 37] or \mathcal{H} -norm [3, 2]. While theoretically well-founded and able to capture the oscillating
 119 nature of texture, these norms are either difficult to implement or cannot capture textures
 120 with a small magnitude. To counteract the staircase effect given by the total variation [8],
 121 other regularization such as the total generalized variation [7] and the relative total variation
 122 [38] were proposed.

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

141 Recently, learning based approaches have been proposed to solve the image decomposi-
 142 tion problem. In [45] the authors proposed a self-example and unsupervised learning approach
 143 where the structure-texture decomposition associated regularization is optimized through the
 144 back propagation of a neural network. Similarly, in [32] it was proposed to recover the struc-
 145 tural component from a random input z from a convolutional generative neural network f_θ ,
 146 and to model the texture as low-rank. In [11], the authors showed that the iterative steps
 147 in the minimization of $\text{TV}-\ell_1$ are similar to the architecture of an LSTM neural network and
 148 they proposed to use an LSTM in order to unfold the iterative hard-thresholding algorithm
 149 of $\text{TV}-\ell_1$. Similarly, in [18], the authors proposed to use a CNN network in order to learn the
 150 structure prior. In [33, 44], other methods based upon unfolding the TV proximal operator
 151 have been proposed. One of the closest approach to our work can be found in [21], where the
 152 authors proposed to learn an image decomposition neural network training upon a handmade
 153 structure-texture dataset consisting of cartoon images onto which a homogeneous texture was
 154 added. However, this approach lacks two core details: texture locality (see Figure 1) and an
 155 associated regularization function to the decomposition that can thereafter be used to solve
 156 inverse problems.

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

162 A way to describe image decomposition is to consider it as a low-dimensional recovery
 163 problem. In this setting, the underlying assumption is that the image we wish to decompose
 164 belongs to the sum of two low dimensional models, i.e. $f = u_0 + v_0$ with u_0 and v_0 each
 165 belonging to a low-dimensional model, denoted by Σ_s for the structure model and Σ_t for the
 166 texture model respectively. Then, the decomposition problem becomes: *recover* $(u_0, v_0) \in$
 167 $\Sigma_s \times \Sigma_t$ *from* $f = u_0 + v_0$.
 168 For each data model Σ_s and Σ_t , we typically set corresponding regularization functions R_s
 169 and R_t whose minimization should enforce Σ_s and Σ_t respectively. We aim to recover (u_0, v_0)
 170 (or at least an approximation) via the optimization problem

$$171 \quad (2.1) \quad \underset{(u,v) \in E \times E}{\text{minimize}} \quad R_s(u) + R_t(v) \quad \text{subject to } f = u + v.$$

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

$$173 \quad (2.2) \quad R_s(u) = \text{dist}(u, \Sigma_s)^2 \quad \text{and} \quad R_t(v) = \text{dist}(v, \Sigma_t)^2.$$

174 Since this approach generally leads to NP-hard problems (e.g ℓ_0 , rank minimization), a con-
 175 vex relaxation is often considered instead (e.g ℓ_1 norm used instead of ℓ_0 for sparsity). This
 176 setting can also be viewed in the context of compressive sensing. By setting the linear oper-
 177 ator $\mathcal{A}=(Id \ Id)$, we aim to recover $\mathbf{x}_0=(u_0, v_0) \in \Sigma_s \times \Sigma_t$ from measurements $f=\mathcal{A}\mathbf{x}_0$, with
 178 $\dim(f) = n_1 n_2 < 2n_1 n_2 = \dim(\mathbf{x})$.

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

185 **2.1. Previous work on low dimensional recovery for image decomposition.** For the
 186 structure component, the total variation

$$187 \quad (2.3) \quad \|u\|_{TV} = \sum_{i \in \Omega} \|(\nabla u)_i\|_2 = \|\nabla u\|_1, \quad \text{with } \Omega = \llbracket 1, n_1 n_2 \rrbracket,$$

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

$$190 \quad (2.4) \quad \Sigma_{GS} = \{u \in \mathbb{R}^{n_1 \times n_2} \mid \|\nabla u\|_0 \leq k\},$$

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

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

$$196 \quad (2.5) \quad u_0 \in \Sigma_{D_s} = \{\mathcal{D}_s x \mid \|x\|_0 \leq k_1\} \quad \text{and} \quad v_0 \in \Sigma_{D_t} = \{\mathcal{D}_t y \mid \|y\|_0 \leq k_2\},$$

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

$$200 \quad (2.6) \quad (x_0, y_0) = \arg \min_{x,y} \|x\|_1 + \|y\|_1 \quad \text{subject to } f = \mathcal{D}_s x + \mathcal{D}_t y,$$

201 and the resulting decomposition is given by $(u, v) = (\mathcal{D}_s x_0, \mathcal{D}_t y_0)$. In fact, with the
 202 appropriate constraints upon the dictionaries and underlying sparsity of u_0 and v_0 ,
 203 (2.6) is able to exactly recover (u_0, v_0) .

- 204 2. In the Low Patch rank interpretation of texture (LPR) model [31], the texture is
 205 considered to be of low patch-rank, i.e

$$206 \quad (2.7) \quad v_0 \in \Sigma_{\text{LPR}} = \{v \in \mathbb{R}^{n_1 \times n_2} \mid \text{rank}(\mathcal{P}(v)) \leq l\},$$

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

$$208 \quad (2.8) \quad \|X\|_* = \sum_{i=1}^{\min(n_1, n_2)} \sigma_i(X)$$

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

$$211 \quad (2.9) \quad \underset{(u,v)}{\text{minimize}} \quad \mu \|u\|_{TV} + \gamma \|\mathcal{P}(v)\|_* \quad \text{subject to } f = u + v.$$

- 212 3. Similarly, in the Blockwise Low-Rank Texture Characterization (BNN) model [23] the
 213 texture is considered to be of low-rank ‘blockwise’, with $v_0 = v_0^1 + \dots + v_0^m$ and for each
 214 $i \in \{1, \dots, m\}$

$$215 \quad (2.10) \quad v_0^i \in \Sigma_{\text{BNN}}^i = \{v \in \mathbb{R}^{n_1 \times n_2} \mid \text{rank}(P_{k_i, \delta_i} \circ S_{\theta_i}(v)) \leq l\},$$

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

$$219 \quad (2.11) \quad \Sigma_{\text{BNN}} = \Sigma_{\text{BNN}}^1 + \dots + \Sigma_{\text{BNN}}^m$$

220 and structure and texture are recovered by the optimization problem

$$221 \quad (2.12) \quad \underset{u,v \in E}{\text{minimize}} \quad \mu \|u\|_{TV} + \sum_{i=1}^m \gamma \|v\|_{*, \text{BNN}}^i \quad \text{subject to } f = u + v.$$

222 4. In the convolutional sparse and low rank coding-based image decomposition model
 223 [41], convolutional filters $\{d_{s,i}\}_{i=1}^{K_s}$, $\{d_{t,i}\}_{i=1}^{K_t}$ that sparsely represent the structure and
 224 texture components are learned. The associated low dimensional models are given by
 (2.13)

$$225 \Sigma_s^{CS} = \left\{ \sum_{i=1}^{K_s} d_{s,i} * x_i \mid \sum_{i=1}^{K_s} \|x_i\|_0 \leq k_1 \right\} \quad \text{and} \quad \Sigma_t^{CS} = \left\{ \sum_{i=1}^{K_t} d_{t,i} * x_i \mid \sum_{i=1}^{K_t} \text{rank}(x_i) \leq k_2 \right\}.$$

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

228 Note that while the ℓ^1 -norm (respectively the nuclear norm) has been shown to be optimal
 229 for sparse recovery (respectively low-rank recovery) [35], all these methods consider a sum
 230 of regularizations for decomposition. This "sum" approach is adapted for product models
 231 $\Sigma_s \times \Sigma_r$. We argue in the following that structure and texture are not best approximated by
 232 such product models.

233 **2.2. The joint structure-texture with shared support model.** For natural images, the
 234 structure and texture components should not be considered disjointedly because they share
 235 some common information: the support. While locally the structure and texture components
 236 can be considered uncorrelated, it is not so the case when taking the whole image into account.
 237 Usually, the structure and texture present in an image share a common border (e.g Figure 1),
 238 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.

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

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

245 *dimensional models as*

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

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

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

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

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

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

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

259 **2.3. On optimal regularization for low dimensional models ?** In the case of separated
 260 models, where we consider that the structure and texture components are uncorrelated, the
 261 optimization problem (2.1) is natural to consider. Indeed, if we set the regularization func-
 262 tions R_s, R_t as in (2.2) and $R_{s,t}(u, v) = \text{dist}((u, v), \Sigma_s \times \Sigma_t)^2$, since $\text{dist}((u, v), \Sigma_s \times \Sigma_t)^2 =$
 263 $\text{dist}(u, \Sigma_s)^2 + \text{dist}(v, \Sigma_t)^2$, we have

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

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

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

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

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

278 **3. PnP for Image decomposition.** Instead of considering two regularization functions in
 279 order to decompose an image (one for each component), we propose to use a single regulariza-
 280 tion function that takes both the structure and texture components as input. By doing so, we
 281 solve the problem of joint regularization and we remove the necessity of a structure/texture
 282 balance tuning parameter. We aim to recover $\mathbf{x}_0 = \begin{pmatrix} u_0 \\ v_0 \end{pmatrix} \in \Sigma_{s \otimes t}$ from the original image
 283 $f = \mathcal{A}\mathbf{x}_0$, with $\mathcal{A} = (Id \ Id)$, via an optimization of the form

$$284 \quad (3.1) \quad \underset{\mathbf{x}=(u,v)}{\text{minimize}} \ R(\mathbf{x}) \quad \text{subject to } f = \mathcal{A}\mathbf{x}.$$

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

290 **3.1. The gradient step denoiser applied to image decomposition.** In [17], the authors
 291 proposed the *gradient step denoiser*, a plug-and-play scheme in which the denoiser is connected
 292 to an explicit regularization functional. The gradient step denoiser takes the form

$$293 \quad (3.2) \quad D(x) = (Id - \nabla R)(x),$$

294 where R is the associated regularization function

$$295 \quad (3.3) \quad R(x) = \frac{1}{2} \|x - N(x)\|^2$$

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

$$299 \quad (3.4) \quad \underset{x}{\text{minimize}} \ R(x) + F(x)$$

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

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

$$305 \quad (3.5) \quad \underset{\mathbf{x} \in E \times E}{\text{minimize}} \ R(\mathbf{x}) + \chi_{\mathcal{C}_f}(\mathbf{x})$$

306 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
 307 decomposition (3.5) fits nicely in the context of image restoration (3.4) with $F = \chi_{\mathcal{C}_f}$ which
 308 can be solved using a projected gradient descent [5]. The following Lemma gives explicitly
 309 the proximal operator of

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

$$312 \quad (3.6) \quad \mathcal{P}_{\mathcal{C}_f}(\mathbf{x}) := \text{Prox}_{\chi_{\mathcal{C}_f}, \lambda}(\mathbf{x}) = \begin{pmatrix} u \\ v \end{pmatrix} - \frac{1}{2} \begin{pmatrix} u + v - f \\ u + v - f \end{pmatrix}$$

313 *Proof.* This is an immediate consequence of the more general Lemma 3.2. For $b = f =$
 314 $u + v$, we have (with $L = \begin{pmatrix} Id & Id \end{pmatrix}$ and L^+ is the pseudo-inverse of L),

$$315 \quad L^+ Lx = \begin{pmatrix} Id & Id \end{pmatrix}^T \left(\begin{pmatrix} Id & Id \end{pmatrix}^T \begin{pmatrix} Id & Id \end{pmatrix} \right)^{-1} \begin{pmatrix} Id & Id \end{pmatrix} x = \frac{1}{2} \begin{pmatrix} u + v \\ u + v \end{pmatrix}$$

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

$$317 \quad (3.7) \quad \begin{aligned} \text{Prox}_{\chi_{\mathcal{C}_f}, \lambda}(\mathbf{x}) &= (I - L^+ L)x + L^+ b \\ &= \begin{pmatrix} u \\ v \end{pmatrix} - \frac{1}{2} \begin{pmatrix} u + v - f \\ u + v - f \end{pmatrix}. \end{aligned} \quad \blacksquare$$

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

$$320 \quad (3.8) \quad \begin{cases} y_{k+1} = (1 - \tau)x_k + \tau D(x_k) \\ x_{k+1} = \mathcal{P}_{\mathcal{C}_f}(y_k) \end{cases}$$

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

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

$$326 \quad (3.9) \quad \mathcal{L}(D) = \mathbb{E}_{\mathbf{x} \in \Sigma_{s \otimes t}, \epsilon \sim \mathcal{N}(0, \sigma_2)} \|D(\mathbf{x} + \epsilon) - \mathbf{x}\|_2^2.$$

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

$$328 \quad (3.10) \quad \begin{aligned} \text{dist}(D(\mathbf{x} + \epsilon), \Sigma_{s \otimes t})^2 &= \inf_{y \in \Sigma_{s \otimes t}} \|D(\mathbf{x} + \epsilon) - y\|_2^2 \\ &\leq \|D(\mathbf{x} + \epsilon) - \mathbf{x}\|_2^2, \end{aligned}$$

Algorithm 3.1 Joint structure-texture gradient descent

Param.: $\tau > 0$
Input f
Output: The output structure and texture $\hat{x} = (\hat{u}, \hat{v})$
 $x_0 = (f, 0)$
while not converged **do**
 $y_{k+1} = (1 - \tau)x_k + \tau D(x_k)$
 $x_{k+1} = \mathcal{P}_{\mathcal{C}_f}(y_{k+1})$
end while

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

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

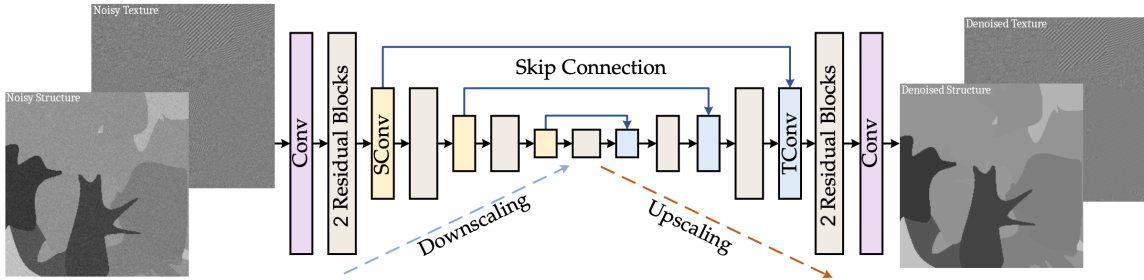


Figure 2: Architecture of the DRUNet denoiser [42] used to parametrize N . Contrarily to the initial implementation of the gradient step PnP, we do not use a noise level map and the structure/texture components are both set in an individual channel.

340 **3.2. Application to inverse problems.** The structure and texture each provide an impor-
 341 tant perceptual information of the content in an image. With prior knowledge on the structure
 342 and texture components in the original image, we may use the regularization $R(u, v)$ in the
 343 applications to solve inverse problems of the form

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

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

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

$$348 \quad (3.12) \quad \min_{x=(u,v)} R(x) \quad \text{subject to } \mathcal{M}(u+v) = b$$

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

$$351 \quad (3.13) \quad \mathcal{P}_{\mathcal{C}_b(\mathcal{M})}(x) = \arg \min_{\mathcal{M}(Id \ Id)y=b} \frac{1}{2} \|y - x\|_2^2.$$

352 Then, we solve the problem via a projected gradient descent

$$353 \quad (3.14) \quad \begin{cases} z_n = x_n - \tau \nabla R(x_n) \\ x_{n+1} = \mathcal{P}_{\mathcal{C}_b(\mathcal{M})}(z_{n+1}) \end{cases}.$$

354 The projection is given by the following Lemma.

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

$$357 \quad (3.15) \quad \mathcal{P}_{\mathcal{C}_b(\mathcal{M})}(x) := \text{Prox}_{\chi_{\mathcal{C}_b(\mathcal{M})}, \lambda}(x) = (I - L^+L)x + L^+b$$

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

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

$$360 \quad (3.16) \quad \begin{aligned} \text{Prox}_{\chi_{\mathcal{C}_f(b)}, \lambda}(x) &= \arg \min_{y \in E^2} \lambda \chi_{\mathcal{C}_f(b)}(y) + \frac{1}{2} \|y - x\|_2^2 \\ &= \arg \min_{\substack{y \in E^2 \\ Ly=b}} \frac{1}{2} \|y - x\|_2^2. \end{aligned}$$

361 We have that $Ly = b$ is equivalent to $y = L^+b + w$ where $w \in \ker(L)$. Hence we minimize
 362 $\min_{w \in \ker L} \|w - x + L^+b\|_2^2$. The solution of this least squares problem is the definition of the
 363 orthogonal projection of $x - L^+b$ on $\ker L$. The orthogonal projection on $\ker L$ is given by
 364 $I - L^+L$ and, as $L^+LL^+ = L^+$, we have

$$365 \quad (3.17) \quad \text{Prox}_{\chi_{\mathcal{C}_f}, \lambda}(x) = (I - L^+L)x - (I - L^+L)L^+b + L^+b = (I - L^+L)x + L^+b. \quad \blacksquare$$

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

$$368 \quad (3.18) \quad \mathcal{P}_{\mathcal{C}_b(\mathcal{M})}(u, v) = \begin{pmatrix} u \\ v \end{pmatrix} + \frac{1}{2} \begin{pmatrix} \mathcal{M}(b - u - v) \\ \mathcal{M}(b - u - v) \end{pmatrix}.$$

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

$$371 \quad (3.19) \quad \min_{\mathbf{x}=(u,v)} R(\mathbf{x}) + \frac{\mu}{2} \|\mathcal{M}(u+v) - b\|_2^2.$$

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

$$374 \quad (3.20) \quad \mathbf{x}_{n+1} = \mathbf{x}_n - \tau \nabla R(\mathbf{x}_n) - \tau \mu \begin{pmatrix} Id \\ Id \end{pmatrix} \mathcal{M}^T (\mathcal{M} (Id \quad Id) \mathbf{x}_n - b)$$

375 **3.3. Projection Tuning parameter for the joint structure-texture model.** One of the
 376 constraints given by considering a single regularization for both the structure and texture is
 377 that we lose any type of control on the given result. Because we are in the setting of exact
 378 decomposition, we do not have any tuning parameter. While it is often advantageous to
 379 have little to no tuning in a decomposition method, we introduce a method to balance the
 380 structure/texture output through the projection operation $\mathcal{P}_{\mathcal{C}_f}$.

381 Essentially, the projection $\mathcal{P}_{\mathcal{C}_f}$ equally adds the residual of the output of the denoiser into
 382 both the structure and texture components in order for the result to fit the equation $f = u + v$.
 383 However, depending on the residual one may wish to add more or less of the residual to either
 384 the structure or texture component. Hence, we may consider the non-orthogonal projection
 385 instead

$$386 \quad (3.21) \quad \tilde{\mathcal{P}}_{\mathcal{C}_f}((u, v)^T, \mu) = \begin{pmatrix} u \\ v \end{pmatrix} + \begin{pmatrix} \mu(f - u - v) \\ (1 - \mu)(f - u - v) \end{pmatrix},$$

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

390 **3.4. Adaptive step selection.** The projected gradient step descent denoiser has the down-
 391 side to be non-convex and usual convex techniques may fail. To handle this, there are multiple
 392 ways we may approach to stabilize the gradient descent:

- 393 1. Initialization near the true solution, e.g using another decomposition scheme to ini-
 394 tialize \mathbf{x}_0 or a trained neural network that directly does a first decomposition.
- 395 2. Backtracking methods as it was originally implemented in [17],
- 396 3. Regularization search: at each iteration, we perform a line search in order to set an
 397 optimal gradient step τ and projection tuning parameter μ that minimizes the most
 398 the regularization function $R_{\mathbf{x}}$.

399 We found that this last approach with the simple initialization $\mathbf{x}_0 = (f, 0)$ leads to the best
 400 recovery result for synthetic images. In a second step, we may also decrease the gradient step
 401 τ_n in order to enforce $\|\mathbf{x}_n - \mathbf{y}_n\| \rightarrow 0$.

402 The parameter search does not impact much the speed of the projected gradient step algorithm
 403 as the computational cost of $R(\mathbf{x})$ is low when compared to $\nabla R(\mathbf{x})$. Moreover, the rate of
 404 convergence is greatly increased, so less iterations are required to reach an optimal output.

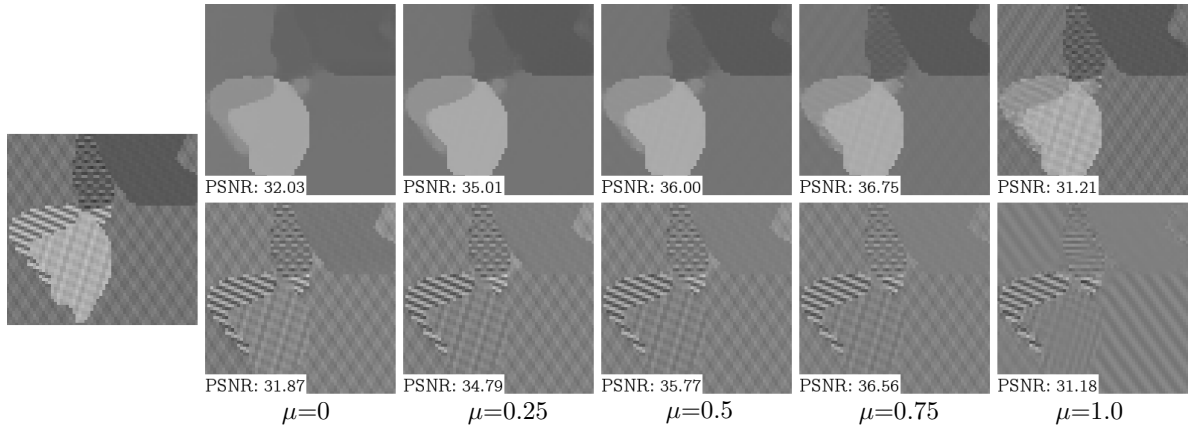


Figure 3: Illustration of the projection direction μ on a synthetic image. At the far left: original image f , top: structure component u_μ , bottom: texture component v_μ . For each decomposition, the first two iterations of the projected gradient descent were obtained using $\mu=0.5$ in order to obtain 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.: $x_0 = (f, 0)$,

Input f

Output: The output structure and texture $\hat{x} = (\hat{u}, \hat{v})$

while not converged **do**

$$\tau_k = \arg \min_{\tau \in \mathbb{R}_+} R((1 - \tau)x_k + \tau D(x_k))$$

$$y_{k+1} = (1 - \tau_k)x_k + \tau_k D(x_k)$$

$$\mu_k = \arg \min_{\mu \in \mathbb{R}} R(\tilde{\mathcal{P}}_{C_f}(y_{k+1}, \mu))$$

$$x_{k+1} = \mathcal{P}_{C_f}(y_{k+1}, \mu_k)$$

end while

405 4. Synthetic images as a regularity prior.

406 **4.1. Background.** Inherently, we wish that our neural network learns the low-dimensional
 407 joint structure-texture model $\Sigma_{s \otimes t}$. One of the core dilemmas with associating machine learn-
 408 ing methods to the image decomposition problem is the absence of ground truth (especially
 409 with natural images). In order to train the denoiser, we designed a procedure that gener-
 410 ates random piecewise constant images (with a connected support) and an associated texture
 411 component, generated from a texture model. This enabled us to train the denoiser with an
 412 endless supply of training examples, without any ambiguity on the ground truth. In [1], the

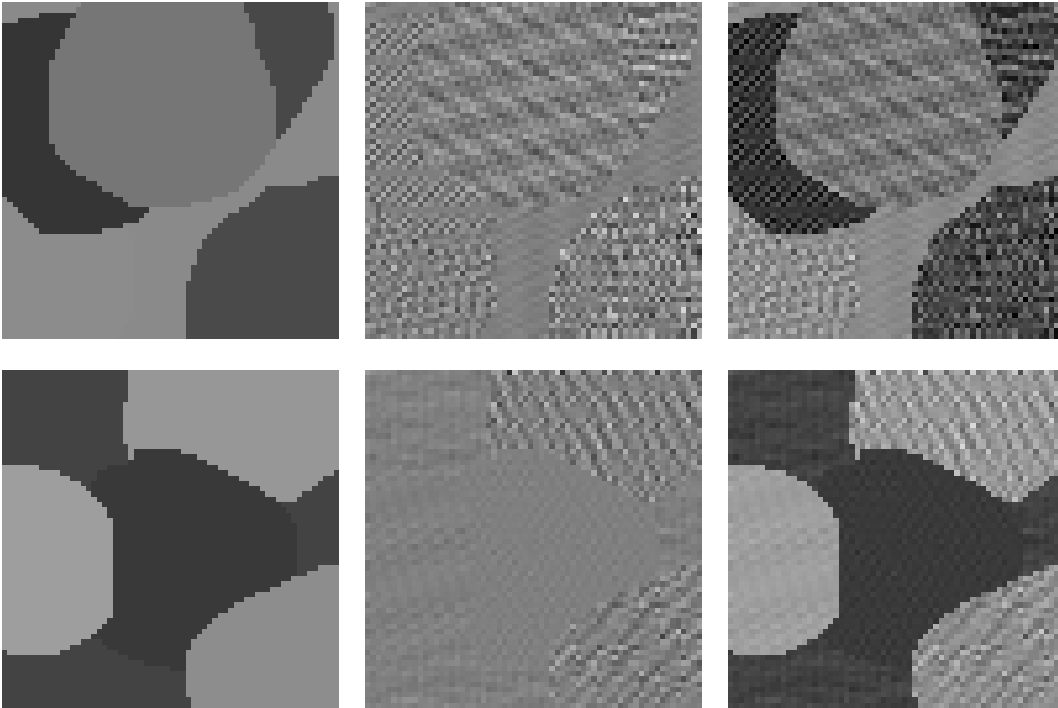


Figure 4: Examples of generated structure (left) and texture (center) images used in the numerical experiments and to train the different neural networks. On the right, we show their sum.

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

419 **4.2. Database design.** In order to create a connected support, we proceeded in two
 420 steps. First we produce a connected support via the Lane-Riesenfeld algorithm [20], where
 421 we randomly scatter initial points around an origin $((pos_x, pos_y)$ in Algorithm 4.1) and we
 422 recursively apply a subdivision process (split + average) to those points until we obtain a
 423 smooth curve (Figure 5).

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

- 426 • **split**(P) = $\{p_1, \frac{p_1+p_2}{2}, p_2, \frac{p_2+p_3}{2}, \dots, p_k, \frac{p_k+p_1}{2}\}$,
- 427 • **average**(P) = $\{\frac{p_1+p_2}{2}, \dots, \frac{p_{k-1}+p_k}{2}, \frac{p_k+p_1}{2}\}$.

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

431 connected support is generated as follows:

- 432 1. Randomly select k (ordered) points around a central point $c \in \mathbb{N}^2$, $P_0 = \{p_1, \dots, p_k\}$.
- 433 2. Subdivide the points, $P_{n+1} = \mathbf{average}(\mathbf{split})(P_n)$, until a smooth enough set of points is achieved.
- 434 3. Project the resulting points onto a canvas and use a flood fill algorithm to obtain the image support.

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

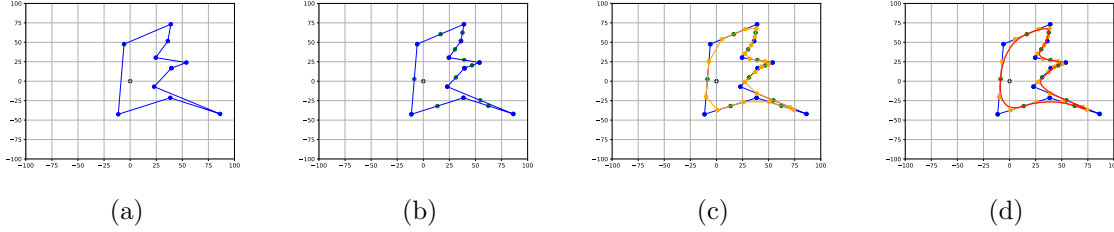


Figure 5: Illustration of the subdivision process (a) Initial point scatter, (b) Splitting step, (c) averaging step, (d) Final shape (in red) after ten subdivisions.

438

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)
 $S = \mathbf{ones}(n, m)$
 $T = \mathbf{generate_texture}(n, m)$
for i in $[0, \dots, K - 1]$ **do**
 $pos_x, pos_y = \mathbf{randint}(0, n), \mathbf{randint}(0, m)$
 $\Omega = \mathbf{generate_support}(pos_x, pos_y)$
 $\alpha_s, \alpha_t = \mathbf{uniform}(s_{min}, s_{max}), \mathbf{uniform}(t_{min}, t_{max})$
 $S|_{\Omega} = \alpha_s \cdot \mathbf{1}_{\Omega}$
 $T|_{\Omega} = \alpha_t \cdot \mathbf{1}_{\Omega} \cdot \mathbf{generate_random_texture}()$
end for

439 The textural component is much more straightforward to generate. In the literature there
440 have been multiple texture models that have been proposed, e.g low-patch rank [31], low-rank
441 [41, 43], sparse dictionary [25], etc... Using random distributions, we generate textures from
442 these models, which are then cropped to fit its corresponding support. We provide an example
443 of the sparse Fourier texture generation in Algorithm 4.2.

444 **4.3. A tool to build an optimal regularization.** Up to our knowledge, every image de-
445 composition model has relied upon a regularization of the form $\lambda R_s(u) + R_t(v)$. As discussed
446 in Section 2.3, while this scheme is optimal when we consider the two components to be un-
447 correlated, it is not the case otherwise. We show evidence that this is in fact suboptimal

Algorithm 4.2 Random sparse Fourier texture generation

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

448 in the case of the joint structure-texture model, and that a regularization which takes both
449 structure and texture as inputs leads to a better result. This further supports our main hy-
450 pothesis that the interaction between the structure and the texture components provides an
451 invaluable information to perform an efficient separation.

452 In our experiments, we selected the DRUNet architecture (see Figure 2) in order to parame-
453 trize the neural network N in (3.3), and we set the texture model Σ_t to a sparse model in the
454 high frequencies (superposition of cosines/sines). We trained three separate denoisers:

- 455 • $D_x = Id - \nabla R_x$ which is trained on denoising structure-texture couples $x = (u, v)$
- 456 • $D_s = Id - \nabla R_s$ which is trained on denoising only the structure.
- 457 • $D_t = Id - \nabla R_t$ which is trained on denoising only the texture.

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

465
466 While D_x is able to achieve similar denoising performance to D_s and D_t for both structure
467 and texture components respectively, our experiments show that D_x is superior in the appli-
468 cation of image decomposition. Given a dataset of 1000 synthetic images, we chose for each
469 image a tuning parameter λ for the minimization of $\lambda R_s + R_t$ that maximizes the PSNR with
470 respect to the ground truth. Even with this harsh condition in favor of the separated models,
471 the joint structure-texture model algorithm has ability to better recover the decomposition
472 into structure and texture (Table 2), with no user input for parameter tuning.

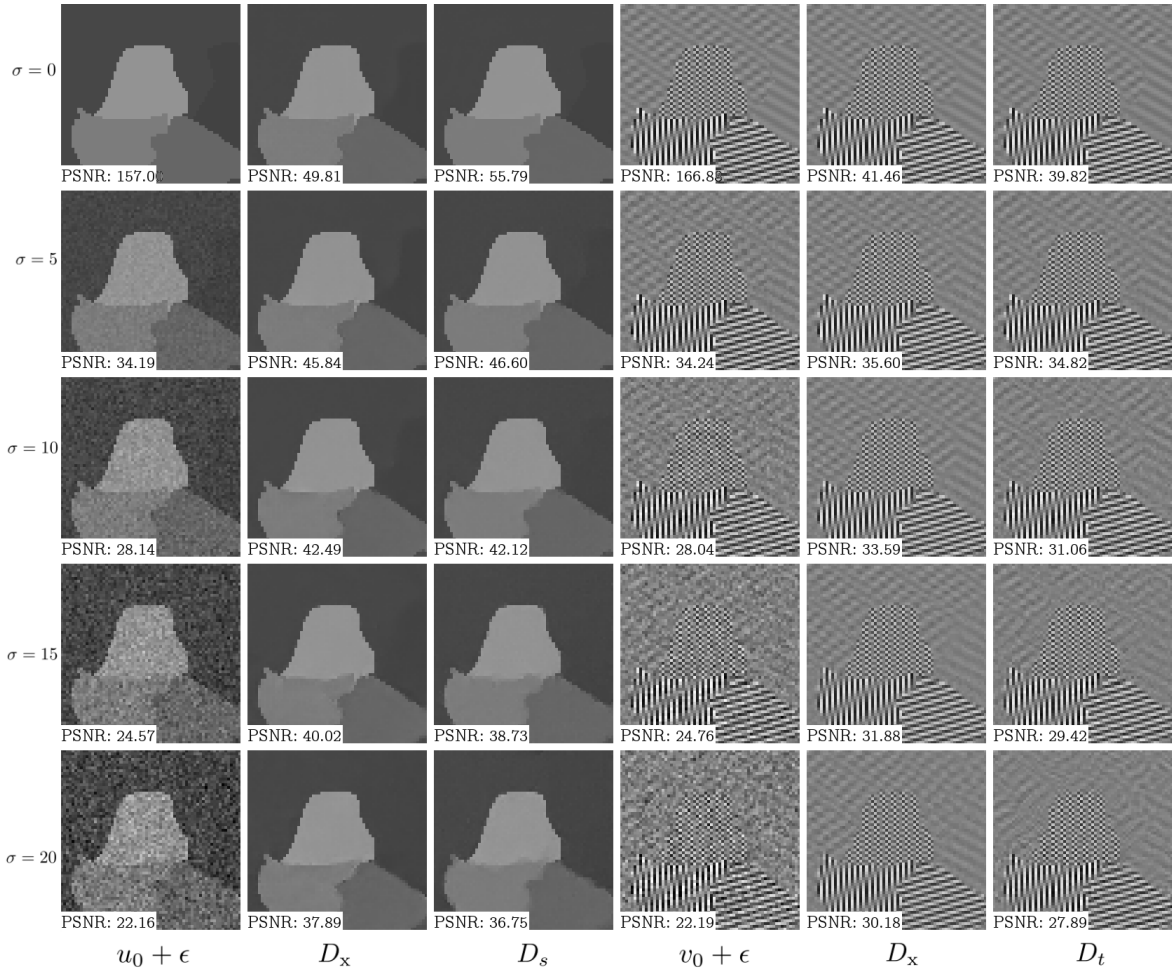


Figure 6: Denoising of a synthetic structure and texture with the different methods and different noise levels using a denoiser D_x that takes both structure and texture as input and D_s , D_t that takes only one component (structure and texture respectively). The results are close for low-level noise, 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.

473 5. Experiments.

474 **5.1. Synthetic image decomposition.** As discussed in Section 4.3, we compared the de-
475 compositions between the two regularization schemes $R(u, v)$ and $R(u) + R(v)$. As we can
476 observe in Figure 7, even for images where the PSNR was close between the two decomposi-
477 tions, the joint structure-texture approach was able to better separate the two models. For
478 example, in the second image, while the structure components for each approach have similar
479 PSNR with respect to the ground truth, there is less texture present in the structure with the
480 joint structure-texture method. Finally, the decomposition using the joint model converges
481 very quickly to an appropriate point, needing less than 10 iterations to reach an optimal value
482 (Fig. 8).

483 **5.2. Inpainting.** The task of inpainting large holes is very ill-posed and thus necessitates
484 a prior knowledge in order to achieve a satisfactory recovery. As presented in [4], image de-
485 composition modeling can be used to inpaint simultaneously both structure and texture. In
486 the case of missing pixels in an image, we found the initialization of the projected gradient
487 algorithm to be of utmost importance in order to recover correctly both the structure and
488 the texture. If initialization is incorrectly set, the masked areas may be considered as pro-
489 viding structure. We found that filling the missing regions with an average onion-peel filling
490 (iteratively filling the holes one layer at a time by taking the average of the surrounding pix-
491 els) provided an adequate initialization. In our experiments (Figure 9) on synthetic images
492 we observe a perfect recovery of the textures present in the image and with an appropriate
493 structure recovery (note that there is no way to recover the correct boundary in the masked
494 areas). This indicates that the denoising task was able to successfully learn the texture model
495 it was trained on.

496 **5.3. Natural image decomposition.** Using a denoiser D_x trained on 64×64 synthetic
497 structure-texture image we decomposed natural images patch-wise using an overlap of 16 (and
498 a patch-size of 64×64). Moreover, we used a line search (as presented in Section (3.4)) at
499 every iteration in order to select an optimal gradient descent parameter. We set the structure
500 model Σ_s as piecewise constant images and the texture model Σ_t as the combination of sparse
501 Fourier textures and low-patch rank. We stress that each decomposition reached in each case

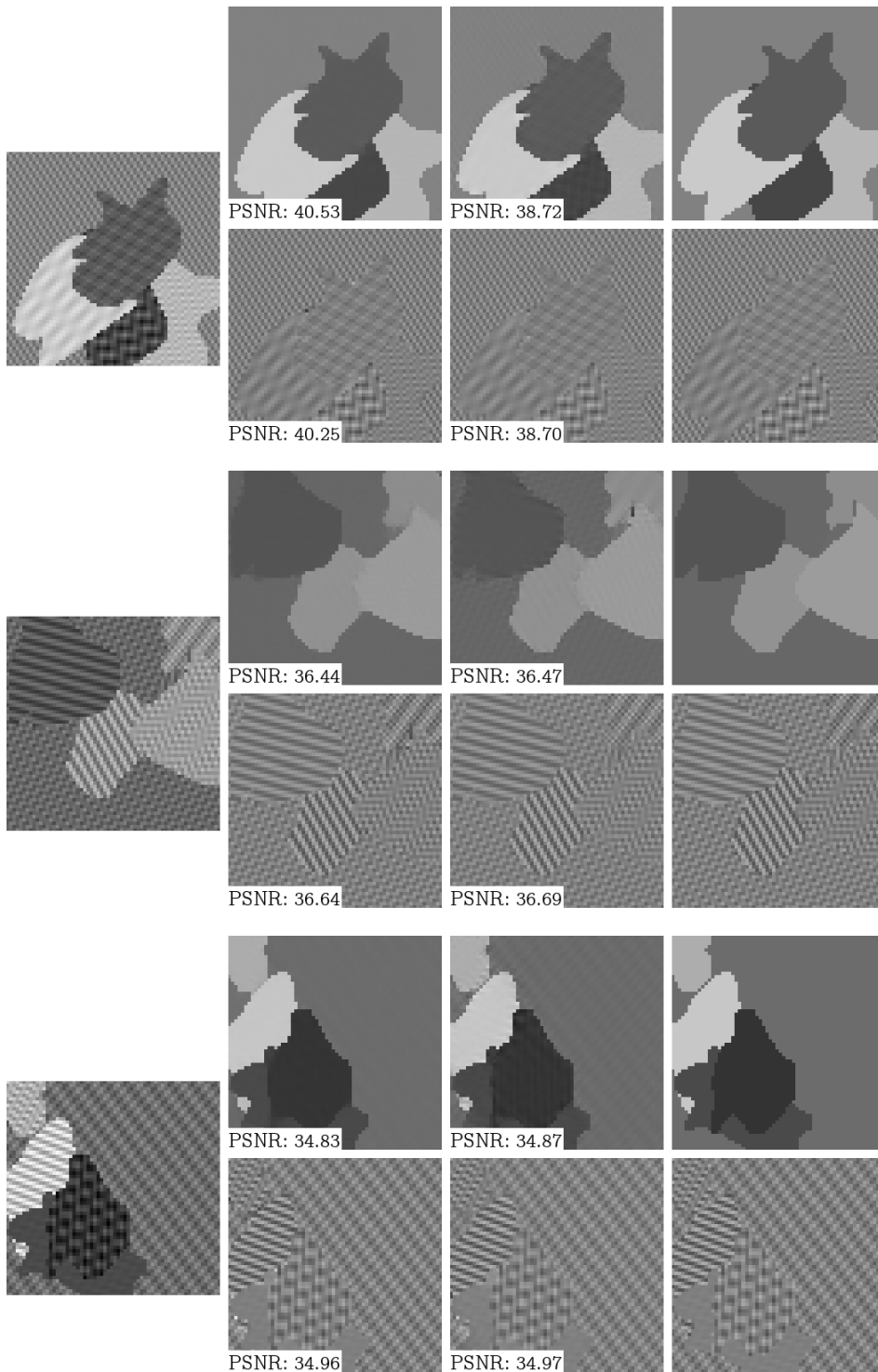


Figure 7: Comparison between the decompositions given by $R(u, v)$ and $R_s(u) + R_t(v)$ minimization. From left to right: original image, output from $R(u, v)$, output from $\lambda R(u) + R(v)$ and the target decomposition (u_0, v_0) . In order to avoid cherry picking bias, the decompositions were 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.

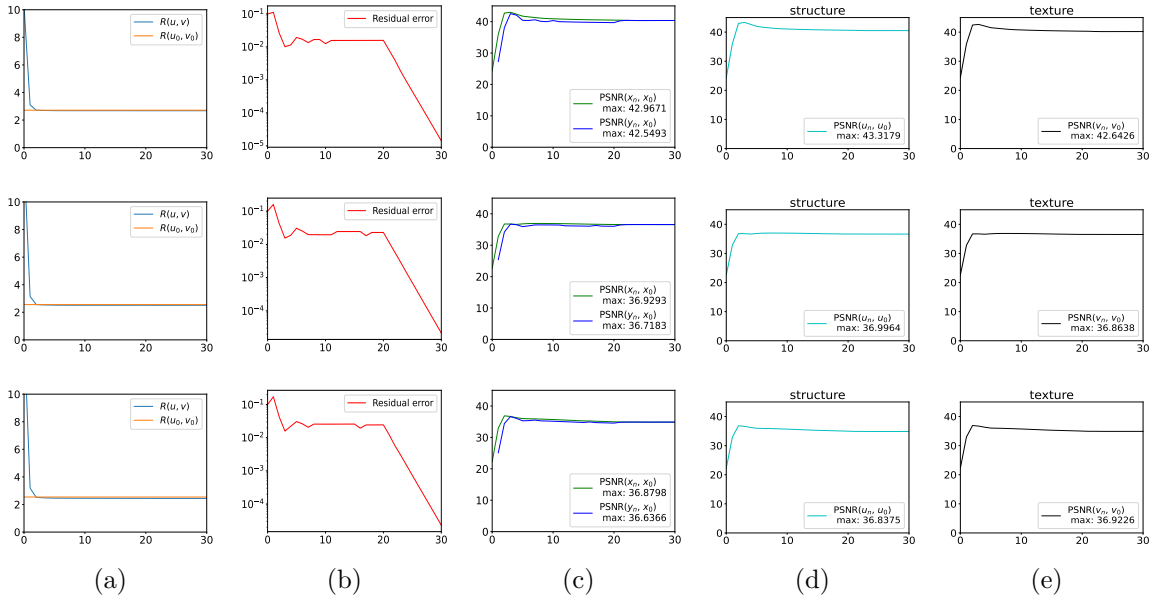


Figure 8: Regularization plots associated with the image decompositions of Fig 7. a) Regularization function R_x , b) Residual error $\frac{\|y_n^1 + y_n^2 - f\|_2}{\|f\|_2}$ in log scale, c) PSNR error with respect to the ground truth x_0 for y_n (blue curve) and x_n (green curve), d-e) PSNR error with 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.

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

505 We performed some decomposition on satellite images taken from the MLSRNet dataset
 506 [27]. As the images are noisy, we performed decomposition with a residual, i.e we do not use
 507 the projection $\mathcal{P}_{\mathcal{C}_f}$ in the last iteration. As the original measured image is noisy, this removes
 508 some of the noise present in the original image from the decomposition as it belong to neither
 509 the structure or texture models. However, we observed that this also extracts some features
 510 in the image such as the central road lines for the same reasons.

511 **5.4. Towards natural image inpainting.** In the context of natural image inpainting, we
 512 found that if the texture is close to the learned low dimensional model, we are able to appro-
 513 priately inpaint the masked regions in the image (Figure 12). The mask shape is not visible in
 514 the reconstructed image. These preliminary results are encouraging for the design of inpaint-
 515 ing methods (and more generally methods to solve inverse imaging problems) based on deep
 516 neural network architectures with a fully controlled low dimensional prior using a synthetic

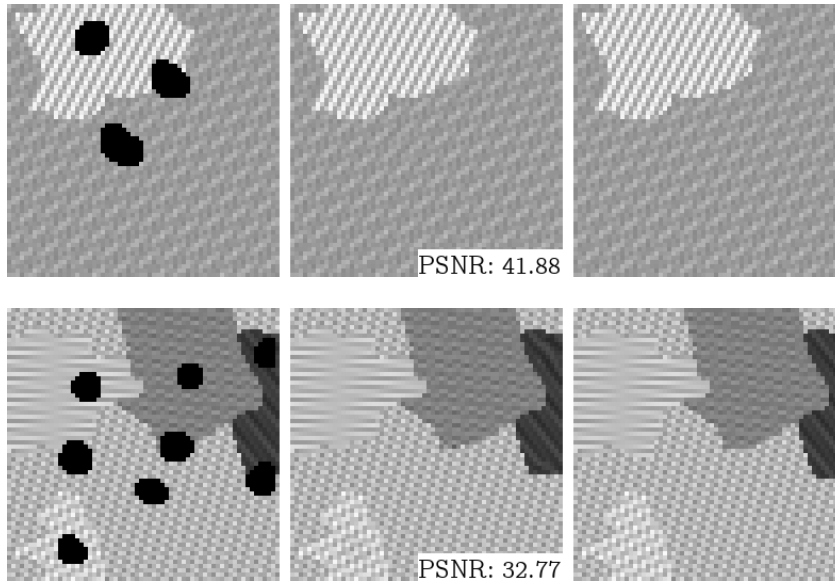


Figure 9: Inpainting recovery on synthetic images. From left to right: input masked image, reconstruction, original image. While the holes are relatively large, the regularization is able to recover well the different textures in the images. The PSNR with respect to the ground truth is shown at the bottom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

517 database.

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

528 Here, we have limited our area of study to piecewise constant structures and sparse Fourier
 529 and low patch rank textures. Other structure/texture models such as piecewise continuous
 530 structures and dictionary sparse textures could be investigated. Moreover, the texture can
 531 be learned on a mixture of different models. Even more broadly, our scheme allows a more
 532 abstract definition of texture such as learning the regularization using a dataset of textures
 533 [19]. Extensions of the two-component decomposition such as the jump-oscillation-trend [9]
 534 or cartoon-smooth-texture [14] could also be investigated in the future using the same process
 535 we have introduced here.

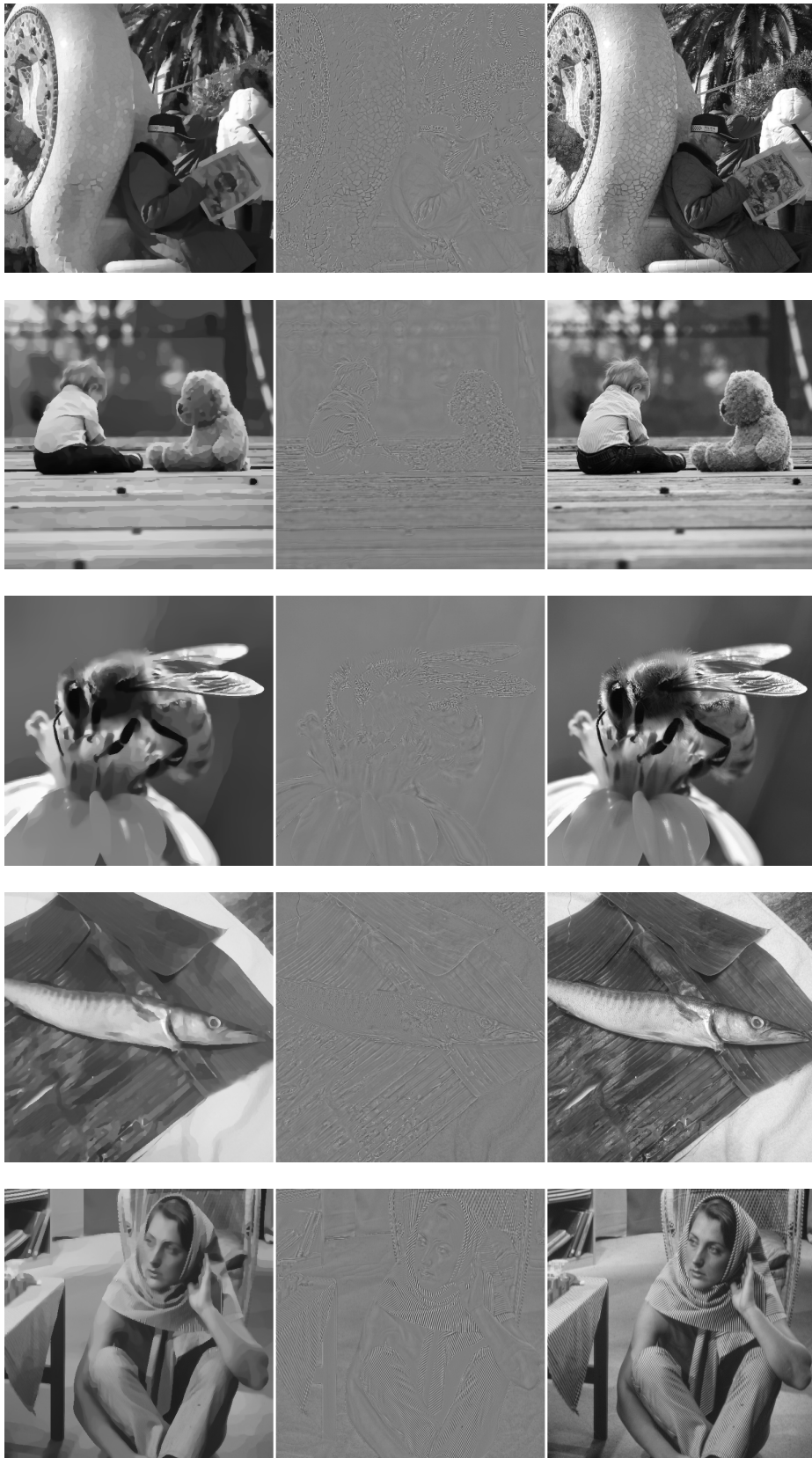


Figure 10: Natural image decomposition using the joint structure-texture model, using a projected gradient descent with line search. From left to right: structure, texture, original image.

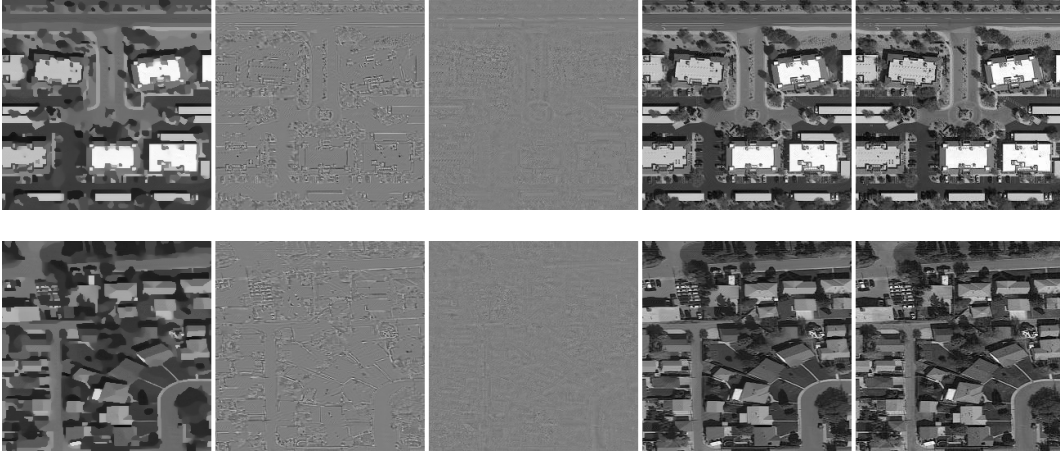


Figure 11: Satellite image decomposition with a residual. From left to right: structure, texture, residual $f - u - v$, denoised image $u + v$, original image.



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).

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

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

- [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. CICONI, 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. FOUCAUT, 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. FOUCAUT 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. HURAUULT, 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.

- 598 [24] S. OYMAK, A. JALALI, M. FAZEL, Y. C. ELДАР, AND B. HASSIBI, *Simultaneously structured models with*
599 *application to sparse and low-rank matrices*, IEEE Transactions on Information Theory, 61 (2015),
600 pp. 2886–2908.
- 601 [25] G. PEYRÉ, *Sparse modeling of textures*, Journal of mathematical imaging and vision, 34 (2009), pp. 17–31.
- 602 [26] J. PROST, A. HOUDARD, A. ALMANSA, AND N. PAPADAKIS, *Learning local regularization for variational*
603 *image restoration*, in International Conference on Scale Space and Variational Methods in Computer
604 Vision, Springer, 2021, pp. 358–370.
- 605 [27] X. QI, P. ZHU, Y. WANG, L. ZHANG, J. PENG, M. WU, J. CHEN, X. ZHAO, N. ZANG, AND P. T.
606 MATHIOPOULOS, *Mlrsnet: A multi-label high spatial resolution remote sensing dataset for semantic*
607 *scene understanding*, ISPRS Journal of Photogrammetry and Remote Sensing, 169 (2020), pp. 337–
608 350.
- 609 [28] E. T. REEHORST AND P. SCHNITER, *Regularization by denoising: Clarifications and new interpretations*,
610 IEEE transactions on computational imaging, 5 (2019), p. 52.
- 611 [29] Y. ROMANO, M. ELAD, AND P. MILANFAR, *The little engine that could: Regularization by denoising*
612 *(red)*, SIAM Journal on Imaging Sciences, 10 (2017), pp. 1804–1844.
- 613 [30] L. I. RUDIN, S. OSHER, AND E. FATEMI, *Nonlinear total variation based noise removal algorithms*, Physica
614 D: nonlinear phenomena, 60 (1992), pp. 259–268.
- 615 [31] H. SCHAEFFER AND S. OSHER, *A low patch-rank interpretation of texture*, SIAM Journal on Imaging
616 Sciences, 6 (2013), pp. 226–262.
- 617 [32] W. SHANG, G. LIU, Y. WANG, J. WANG, AND Y. MA, *A non-convex low-rank image decomposition*
618 *model via unsupervised network*, Signal Processing, 223 (2024), p. 109572.
- 619 [33] B. SHI, W. XU, AND X. YANG, *Ctdnet: cartoon-texture decomposition-based gray image super-resolution*
620 *network with multiple degradations*, JOSA B, 40 (2023), pp. 3284–3290.
- 621 [34] J.-L. STARCK, M. ELAD, AND D. L. DONOHO, *Image decomposition via the combination of sparse repre-*
622 *sentations and a variational approach*, IEEE transactions on image processing, 14 (2005), pp. 1570–
623 1582.
- 624 [35] Y. TRAONMILIN, R. GRIBONVAL, AND S. VAITER, *A theory of optimal convex regularization for low-*
625 *dimensional recovery*, Information and Inference, A journal of the IMA, (2024).
- 626 [36] S. V. VENKATAKRISHNAN, C. A. BOUMAN, AND B. WOHLBERG, *Plug-and-play priors for model based*
627 *reconstruction*, in 2013 IEEE global conference on signal and information processing, IEEE, 2013,
628 pp. 945–948.
- 629 [37] L. A. VESE AND S. J. OSHER, *Modeling textures with total variation minimization and oscillating patterns*
630 *in image processing*, Journal of scientific computing, 19 (2003), pp. 553–572.
- 631 [38] L. XU, Q. YAN, Y. XIA, AND J. JIA, *Structure extraction from texture via relative total variation*, ACM
632 transactions on graphics (TOG), 31 (2012), pp. 1–10.
- 633 [39] R. XU, Y. XU, Y. QUAN, AND H. JI, *Cartoon-texture image decomposition using orientation character-*
634 *istics in patch recurrence*, SIAM Journal on Imaging Sciences, 13 (2020), pp. 1179–1210.
- 635 [40] W. XU, C. TANG, Y. SU, B. LI, AND Z. LEI, *Image decomposition model shearlet-hilbert- l_2 with better*
636 *performance for denoising in espi fringe patterns*, Applied Optics, 57 (2018), pp. 861–871.
- 637 [41] H. ZHANG AND V. M. PATEL, *Convolutional sparse and low-rank coding-based image decomposition*, IEEE
638 Transactions on Image Processing, 27 (2017), pp. 2121–2133.
- 639 [42] K. ZHANG, Y. LI, W. ZUO, L. ZHANG, L. VAN GOOL, AND R. TIMOFTE, *Plug-and-play image restoration*
640 *with deep denoiser prior*, IEEE Transactions on Pattern Analysis and Machine Intelligence, 44 (2021),
641 pp. 6360–6376.
- 642 [43] Z. ZHANG AND H. HE, *A customized low-rank prior model for structured cartoon-texture image decom-*
643 *position*, Signal Processing: Image Communication, 96 (2021), p. 116308.
- 644 [44] C. ZHENG, D. SHI, AND W. SHI, *Adaptive unfolding total variation network for low-light image en-*
645 *hancement*, in Proceedings of the IEEE/CVF international conference on computer vision, 2021,
646 pp. 4439–4448.
- 647 [45] F. ZHOU, Q. CHEN, B. LIU, AND G. QIU, *Structure and texture-aware image decomposition via training*
648 *a neural network*, IEEE Transactions on Image Processing, 29 (2019), pp. 3458–3473.