

Wasserstein Loss for Image Synthesis and Restoration

Guillaume Tartavel, Gabriel Peyré, Yann Gousseau

▶ **To cite this version:**

Guillaume Tartavel, Gabriel Peyré, Yann Gousseau. Wasserstein Loss for Image Synthesis and Restoration. SIAM Journal on Imaging Sciences, 2016, 9 (4), pp.1726-1755. hal-01292843

HAL Id: hal-01292843
https://hal.science/hal-01292843

Submitted on 23 Mar 2016

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.

WASSERSTEIN LOSS FOR IMAGE SYNTHESIS AND RESTORATION

GUILLAUME TARTAVEL*,
GABRIEL PEYRÉ†, AND YANN GOUSSEAU*

Abstract. This paper presents a novel variational approach to impose statistical constraints to the output of both image generation (to perform typically texture synthesis) and image restoration (for instance to achieve denoising and super-resolution) methods. The empirical distributions of linear or non-linear descriptors are imposed to be close to some input distributions by minimizing a Wasserstein loss, i.e. the optimal transport distance between the distributions. We advocate the use of a Wasserstein distance because it is robust when using discrete distributions without the need to resort to kernel estimators. We showcase different estimators to tackle various image processing applications. These estimators include linear wavelet-based filtering to account for simple textures, non-linear sparse coding coefficients for more complicated patterns, and the image gradient to restore sharper contents. For applications to texture synthesis, the input distributions are the empirical distributions computed from an exemplar image. For image denoising and super-resolution, the estimation process is more difficult; we propose to make use of parametric models and we show results using Generalized Gaussian Distributions.

Key words. Optimal transport, Wasserstein loss, total variation, Generalized Gaussian Distributions, denoising, super-resolution.

AMS subject classifications. 90C25, 68U10

1. Introduction.

The statistical modeling of natural images is a long standing problem which is useful to tackle a wide range of image processing applications. In this article, we focus our attention to problems in both image synthesis and image restoration; we show how to address them with variational approaches using the statistical model as a data fidelity rather than a regularization. This allows us to obtain results that are more faithful to the targeted model (as opposed to traditional variational regularization methods) and to better restore and generate fine scale details.

1.1. Previous Works.

Texture synthesis: sampling vs. optimization. The texture synthesis problem is usually framed as a statistical estimation and re-sampling problem. Probably the most complete framework to achieve this goal is the work of Zhu et al. [74] that proposes a maximum-entropy method to estimate a generic class of Gibbs distributions from a given exemplar, leveraging only the stationarity of the thought after solution. The major drawback of this approach is that the resampling stage is slow since it necessitates the use of a Gibbs sampler. At the opposite part of the spectrum, the method of Galerne et al. [27] uses a simple statistical model (a stationary Gaussian distribution). It can only capture "micro-textures" without any geometrical pattern but can be learned and can re-synthesized very efficiently, hence enabling its use in real-time computer graphics pipelines [28]. Let us note however that the state of the art in computer graphics rather relies on heuristic methods to learn and re-sample: it typically considers re-copy of groups of pixels or patches from the input exemplar [19, 69, 38]. While these non-parametric approaches lead to high fidelity in the re-sampling, they suffer from a poor understanding of the underlying generated

*Telecom ParisTech, LTCI CNRS, 46 rue Barrault, F-75634 PARIS CEDEX 13, FRANCE
†CEREMADE, Université Paris-Dauphine, Place du Marechal De Lattre De Tassigny, 75775 PARIS CEDEX 16, FRANCE.

distribution and lead to synthesis results that copy verbatim large chunks of the input image without much randomization or "innovation"—see [1, 62] for an experimental exploration of this aspect.

To speed-up (and sometime even improve) the sampling quality, a increasingly large body of literature has considered replacing random sampling from a given distribution by an optimization of a non-convex energy, usually starting from a random initialization. While there is no strong mathematical analysis of the generated distribution, the rationale is that the random exploration of local minima associated to an energy which measures deviations from statistical constraints gives in practice results that are at least as good as sampling from the maximum entropy distribution inside these constraints, as originally advocated in [74]. The pioneer work [34] of Heeger and Bergen, that imposes the first order statistics of wavelet coefficients through iterative projections, may be seen as a precursor of such variational approaches. This idea was pushed much further by Portilla and Simoncelli [50] by imposing higher order statistics. Then, patch-based recopy methods have been explicitely reformulated as non-convex random optimization [35]. Tartavel et al [64] also integrates patch-based ideas into a variational texture synthesis algorithm by using sparse coding in a learned dictionary. Another recent line of research, also based on the statistical modeling of non-linear image features, makes use of deep neural networks, which are thus also trained on the input exemplar [29, 42].

Variational Restoration. While it is natural to think of texture synthesis as a statistical re-sampling problem rather than an optimization one, a large body of literature on image restoration (e.g. denoising or super-resolution) thinks the other way around. The idea of variational regularization, very often derived as a MAP (*Maximum A Posteriori* estimator), is to recover a clean and high resolution image by minimizing a functional accounting for both a fidelity term (which depends on the noise's properties) and a prior (which takes into account some knowledge about the structures of the image to recover). Two popular classes of priors are the Total Variation (TV) model of Rudin, Osher and Fatemi [57] and the sparse wavelet approximations of Donoho and Jonhstone [18]. The TV model favors cartoon images having a sparse gradient distribution, and shares similarity with wavelet models that capture discontinuous images using multiscale decompositions. Total variation has been used in almost all imaging applications, including for instance denoising [57], deblurring [6], interpolation [33], or image decomposition [3]. Wavelet approaches can give rise to different kinds of prior, using thresholding in orthogonal [18] or translation invariant [13] dictionaries, and used either as analysis or synthesis regularizers for inverse problems [22]. It is also possible to go beyond simple first-order sparse statistical models by accounting for spatial and multiscale dependency between wavelet coefficients [59, 55, 51].

Beside these historically popular methods, the state of the art in denoising and (to a lesser extend) more general restoration problems is non-local processing, that operates by comparing patches in the image. This is somehow similar to the recopy methods presented above for texture synthesis applications. The non-local mean method [9] introduces this idea for denoising as a non-linear filtering operator—see also [4]. It was then recast as a variational regularization [30, 23] that can be used for restoration purpose [2, 49]. The BM3D approach [16] extends one step further these non-local ideas by merging them with sparse approximation, giving rise to state of the art results. More recently, several works have achieved even better results by modeling patches through Gaussian distributions [72, 37].

Dictionary learning somehow bridges the gap between these patch-based comparison methods and sparse wavelet (or gradient) regularization. This approach was introduced by Olshausen and Fields [47] and then refined for denoising applications (see for instance [24, 21]) and more general imaging problems (see for instance [43]).

Statistical constraints for restoration. A well known issue with variational MAP estimators is that the output of the method is in general not distributed according to the though-after distribution, as explained and analyzed in details in the works of Nikolova [45]. A striking class of examples is given in [32] where it is argued that most thresholding operators are adapted to data which are significantly sparser that the initial prior. However, it makes sense for many applications to impose the distribution of the estimator, typically to reach a high fidelity in restoring fine scale details or geometric patterns such as edges. Variational methods usually do not come with such a guaranty on the output, which is problematic in some cases. A well known example is the output of TV regularization: it suffers from the "stair-casing" effect [45], resulting in many pixels having zero gradient values. This is usually not a desirable statistical feature and is in contradiction with a Laplacian prior on the gradient. More generally, the MAP tends to under-estimate the heavy tails of the distributions [71], hence removing some information and often resulting in the degradation of textured parts of the image.

The standard way to solve this issue is to use a conditional expectation (minimum mean square estimator) rather than a MAP. This has been proposed for instance in the case of sparse wavelet priors in [58] and for the total variation prior in [40]. The literature on inverse problems using bayesian sampling is huge, and we refer for instance to [60] for an overview. By construction, these approaches are perfectly faithful to the prior distribution, which alleviates some of the drawback of MAP estimators (e.g. the staircasing effect for TV). As already noted above for synthesis application, the main issue is the high computational complexity of the corresponding probabilistic sampling methods (Markov Chain Monte Carlo methods such as Gibbs samplers).

Statistical fidelity and optimal transport. Following several recent works, we propose to explore an alternative way to impose statistical prior constraints on the output of both synthesis and restoration algorithms. This approach is more heuristic but computationally less expensive than sampling-based methods; it integrates the statistical constraints directly into a variational MAP-like approach.

Our approach takes its roots in a series of conference papers where one of us introduced statistical losses over pixels values and wavelet coefficients to perform texture synthesis and restoration [53, 54]. The authors of [61] use a convexification of the same energy in order to be able to compute global optima at the expense of slower algorithms. An initial version of our framework was presented in the thesis manuscript [62]. A related work [20] has been done in parallel with an application to super-resolution. These previous works advocate the use of optimal transport distances (also known as Wasserstein distances) to account in a robust and simple way for discrepancies between discrete empirical distributions. We refer to the monograph of Villani for a detailed description of Wasserstein distance [67]. These distances have recently gained some interest in image processing, in particular to perform color and texture manipulation—see for instance [26] and the references therein.

Let us also note that optimal transport methods differ from more standard losses to measure statistical discrepancy. The most routinely used is the Kullback-Leibler (KL) divergence which is reminiscent of estimation using maximum likelihood meth-

ods. Of particular interest for our work are the recent works [11, 12] that use a KL loss to account for gradient statistics in the context of restoration problems and therefore pursue the same goal as we do in Sect. 4: preserving gradient statistics while restoring images. A similar approach is used by [73] to extend the Field of Experts approach of [56] with a Kullback-Leibler divergence to control the gradients' distribution. The idea of imposing the gradients' distribution is also at the core of [31] to tackle inverse problems in biomedical imaging. A chief advantage of the Wasserstein loss is that it can compare pairs of discrete empirical distributions without the need to use kernel density estimators, which is further complicated in practice by the difficult problem of selecting the bandwidth of the kernel.

1.2. Contributions. Sect. 2 introduces a novel framework to impose statistical constraints in imaging using a Wasserstein loss. The key idea is to use an optimal transport between discrete distributions of a set of features, that can easily be optimized using gradient descent methods. Sect. 3 presents a first application of this idea to perform texture synthesis, which is achieved by simply applying a memory-limited quasi-Newton (L-BFGS) optimization scheme to this statistical loss. Several examples on both linear (wavelet-based) and non-linear (sparse coding) features illustrate the ability of the method to generate random samples with prescribed features' distributions. Sect. 4 presents a second application to image restoration and in particular to denoising and super-resolution (inpainting of scattered missing pixels). This approach imposes the gradient distributions of the resulting image on top of a traditional variational estimator (we illustrate here the method with a TV regularizer). The target distributions are estimated from the single noisy observation by restricting the search to a parametric family of Generalized Gaussian Distributions. Numerical results show how this additional statistical fidelity is able to improve the restoration of fine scale details, edges, and textural features.

A Matlab implementation of the proposed framework is available online*.

2. Wasserstein Loss. This section formalizes the statistical fidelity term that is used in the remaining part of this paper both to perform variational texture synthesis (Sect. 3) and image restoration with statistical constraints (Sect. 4). The main novelty with respect to previous works is the use of an optimal transport distance between empirical measures (i.e. sum of Dirac's). While being non-convex, a chief advantage of this approach is that it is robust (because optimal transport is a well defined geometric distance between Dirac masses) and it is simple to compute (at least for 1-D distributions).

2.1. Wasserstein Distance. We consider discrete probability measures which are normalized sums of N Dirac's

$$\nu_z \overset{\text{def.}}{=} \frac{1}{N}\sum_{i=1}^{N} \delta_{z_i} \tag{2.1}$$

where $\delta_{\boldsymbol{t}}$ is the Dirac measure at a point $\boldsymbol{t} \in \mathbb{R}^J$. In practice, such a measure is an empirical distribution of a set of N features extracted from an image, as detailed in Sect. 2.2 below.

Optimal transport provides a blueprint to define classes of distances between measures defined on quite general spaces. We refer to the monograph of Villani [67] for

*https://bitbucket.org/gtartavel/autodiff_matlab/

an extended description of these distances and their properties. For the sake of simplicity and for its practical relevance to the targeted applications in imaging, we only consider optimal transport with a squared Euclidean cost and between two discrete measures $(\nu_z, \nu_{z'})$ having the same number N of Dirac masses. In this very specific setting, optimal transport is equivalent to optimal assignment and the corresponding distance, the so-called 2-Wasserstein distance, is defined as

$$\mathcal{W}(\nu_z, \nu_{z'})^2 \overset{\text{def.}}{=} \min_{\sigma} \frac{1}{N} \|z - z' \circ \sigma\|^2 = \frac{1}{N} \sum_{i=1}^{N} \left\| z_i - z'_{\sigma(i)} \right\|^2 \tag{2.2}$$

where σ runs among the set Σ_N of permutations of $\{1 \dots N\}$.

In the following, we denote $\sigma_{z,z'}$ an optimal σ in (2.2) and we assume an arbitrary choice if $\sigma_{z,z'}$ is not the unique solution. It corresponds to an optimal assignment between each point z_i and the point $z'_{\sigma_{z,z'}(i)}$.

The following proposition recalls several simple but important properties of optimal assignments. We refer to [7] for more details and proofs. It states that the optimal assignment to z' can be interpreted as an orthogonal projection on the highly non-convex and high dimensional set of all possible permutations of z'. Another crucial property is that the Wasserstein loss as a function of the Dirac's position (e.g. z or z') is smooth almost everywhere, namely as soon as the optimal permutation is unique (which is almost surely the case for points in arbitrary positions). Furthermore, the formula (2.4) for the gradient is straightforward since the Wasserstein loss is locally a simple ℓ^2 loss as long as the optimal permutation does not change.

PROPOSITION 2.1. *Let use denote*

$$\mathcal{C}_{z'} \overset{\text{def.}}{=} \{s \,;\, \nu_s = \nu_{z'}\} = z' \circ \Sigma_N = \{z' \circ \sigma \,;\quad \sigma \in \Sigma_N\}\,.$$

One has

$$\mathcal{W}(\nu_z, \nu_{z'})^2 = N^{-1} \operatorname{dist}(z, \mathcal{C}_{z'})^2 \quad \textit{where} \quad \operatorname{dist}(z, \mathcal{C})^2 \overset{\text{def.}}{=} \min_{s \in \mathcal{C}} \|z - s\|^2 \tag{2.3}$$

and

$$\operatorname{proj}_{\mathcal{C}_{z'}}(z) = z \circ \sigma_{z,z'} = \big(z_{\sigma_{z,z'}(i)}\big)_{i=1}^{N}.$$

If $\sigma_{z,z'}$ is the unique minimizer of (2.2), then the function $f_{z'} : z \mapsto \mathcal{W}(\nu_z, \nu_{z'})^2$ is C^1 at z and one has

$$\nabla f_{z'}(z) = 2N^{-1}\big(z - \operatorname{proj}_{\mathcal{C}_{z'}}(z)\big). \tag{2.4}$$

Computing $\sigma_{z,z'}$ and hence evaluating $\mathcal{W}(\nu_z, \nu_{z'})^2$ and the gradient (2.4) can be achieved using standard optimal assignment algorithms, which are combinatorial optimization methods with roughly cubic complexity. These methods do not scale to large N, and are thus not usable for imaging applications. The only exception is the case of 1-D distributions (i.e. each z_i is in $\mathbb{R}$) as detailed in the following proposition (see [7]).

PROPOSITION 2.2 (1D Wasserstein Distance). *For $J = 1$, one has*

$$\mathcal{W}(\nu_z, \nu_{z'})^2 = N^{-1} \|z \circ \sigma_z - z' \circ \sigma_{z'}\|^2 \quad \textit{and} \quad \sigma_{z,z'} = \sigma_{z'} \circ \sigma_z^{-1} \tag{2.5}$$

where σ_z sorts the values of z, that is, $\quad \ldots \leq z_{\sigma_z(i-1)} \leq z_{\sigma_z(i)} \leq \ldots$.

This shows that the Wasserstein loss and its corresponding gradient can be computed in $O(N\log(N))$ operations by simply sorting the values. This is the main reason why we consider 1-D distributions in the following. In the presence of higher dimensional distributions, we replace them by a set of 1-D distributions obtained by projections. This corresponds to the "sliced" Wasserstein approximation introduced in [7]. A typical illustration of this idea can be found in Sect. 4 where the 2-D distribution of the gradients of an image is replaced by K 1-D distributions of directional derivatives (4.3).

2.2. Feature-Space Statistical Loss. A feature extractor is a map H from an image $u \in \mathbb{R}^P$ having P pixels to a set of N real-value features $\big(H(u)_i\big)_{i=1}^N \in \mathbb{R}^N$. Very often one has $P = N$ but this needs not to be the case. The most simple feature extractors are linear maps such as:

- Pixel values using the identity map $H(u) = u$.
- Directional derivatives as later defined in (4.3).
- Filtering $H(u) = \psi \star u$ against a "wavelet"-like filter ψ (where $\star$ is the convolution).

It is of course possible to consider more complicated non-linear features such as SIFT vectors [41] or structure tensors [70] to account for more complicated features in an image such as edge curvature or corners. Sect. 3.3 studies a different way to extract and use non-linear features through sparse coding.

To perform texture synthesis (Sect. 3) or to improve the quality of restoration methods (Sect. 4), we make use of a set $\{H_k\}_{0\leq k\leq K}$ of feature extractors. A typical example (see Sect. 3.2 which uses it for texture synthesis) is to set $H_0 = \mathrm{Id}$ to be the pixel value extractor and to let the others extractors to account for a wavelet transform $H_k(u) = \psi_k \star u$ where ψ_k is a wavelet kernel at a given scale and orientation.

Given a set of input distributions $(\nu_{z_k})_{0\leq k\leq K}$, we now define our Wasserstein statistical loss as

$$E(u|(\nu_{z_k})_k) \stackrel{\text{def.}}{=} \sum_{k=0}^{K} \mathcal{W}(\nu_{H_k(u)}, \nu_{z_k})^2. \tag{2.6}$$

The input distributions $(\nu_{z_k})_k$ can be learned directly from an exemplar image v by setting $z_k = H_k(v)$ (as it is the case for texture synthesis in Sect. 3). In cases where no input image is available or if v is degraded (as it is the case for image restoration in Sect. 4), one needs to consider more advanced estimation procedures.

Note that we do not put any weights in front of each Wasserstein term in the sum (2.6) since they can be absorbed in the definition of the $H_k(u)$ as $\lambda^2\mathcal{W}(\nu_z, \nu_{z'})^2 = \mathcal{W}(\nu_{\lambda z}, \nu_{\lambda z'})^2$.

2.3. Gradient Computation. The non-convex Wasserstein loss $E(\cdot|(\nu_{z_k})_k)$ is to be used in variational methods for synthesis and restoration. We use gradient-type optimization schemes in the following sections to compute local minimizers of synthesis or restoration energies.

A nice feature of this approach is that the gradient of E (with respect to the image u) is simple to compute using the chain rule:

$$\nabla E(u|(\nu_{z_k})_k) = \sum_{k\leq K} \big[\partial H_k(u)\big]^*\big(\nabla f_{z_k}(H_k(u))\big) \tag{2.7}$$

where the gradient of the map f_z is computed as detailed in (2.4). One thus needs to compute the adjoint $[\partial H_k(u)]^*$ of the differential $\partial H_k(u)$ of the feature extractor H_k. Each time we introduce and use a new feature extractor, we thus explain how to compute this adjoint map.

3. Texture Synthesis using Linear and Sparse Features. This section applies the Wasserstein loss to the problem of texture synthesis, and explores the use of different classes of features extractors.

3.1. Variational Texture Synthesis Methods. Following several recent works (see Section 1.1) we formulate the problem of texture synthesis from an exemplar v as the one of computing a random stationary point u of a texture fidelity energy

$$\min_{u\in\mathbb{R}^P} E(u|(\nu_{z_k})_k) \quad \text{where} \quad \forall\, k=0,\dots,K, \quad \nu_{z_k} \overset{\text{def.}}{=} \nu_{H_k(v)}. \tag{3.1}$$

In order to compute a randomized stationary point of this energy, we use a gradient descent method. Starting from a realization $u^{(0)} \in \mathbb{R}^P$ of a white noise, we define the iterates as

$$u^{(\ell+1)} \overset{\text{def.}}{=} u^{(\ell)} - L^{(\ell)} \nabla E(u^{(\ell)}|(\nu_{z_k})_k) \tag{3.2}$$

where ∇E is computed using formula (2.7).

Here $L^{(\ell)}$ is a linear operator, which is intended to approximate the inverse of the Hessian of $E(\cdot|(\nu_{z_k})_k)$ at $u^{(\ell)}$. For the numerical applications, we use the L-BFGS algorithm [46], which is a limited-memory version of the BFGS algorithm [52]. This algorithm iteratively builds a low-rank approximation $L^{(\ell)}$ of the inverse of the Hessian. When dealing with high dimensional problems, this is a good compromise between the tractability of the gradient descent and the efficiency of Newton's algorithm. We use the Matlab implementation [48]; a convergence analysis of this algorithm in the case of a class of non-smooth functions is proposed in [39].

3.2. Linear Wavelet Features for Synthesis. To illustrate the usefulness of the proposed synthesis scheme, we first instantiate it in a simple setting, corresponding to a wavelet transform feature extractor. This allows us to revisit and improve the celebrated texture synthesis algorithm of Heeger and Bergen [34].

Heeger and Bergen's (HB) original algorithm. For the sake of simplicity, we consider a gray-level version of the algorithm. To impose the gray levels of the input image, the first feature is the image pixel values, i.e. we set $H_0(u) = u$. A steerable wavelet transform [34] computes wavelet coefficients with a linear transform

$$H_{1,\dots,K}(u) = (H_k(u))_{k=1}^K$$

where $H_k(u)$ is a sub-sampled convolution against a rotated and scaled wavelet atom. This feature extractor can be understood as multi-scale directional derivatives (i.e. derivatives of increasingly blurred version of the image u). The computation of the linear map $H_{1,\dots,K}$ is obtained in linear time with a fast filter bank algorithm [34].

Heeger and Bergen initial algorithm alternates between matching the spatial statistics and matching the wavelet statistics. From a Gaussian white noise realization $u^{(0)}$, it first modifies the gray-values statistics of the current image $u^{(\ell)}$ so that they match those of the exemplar v by projecting onto $\mathcal{C}_{\nu_0}$, i.e.

$$\tilde{u}^{(\ell)} \overset{\text{def.}}{=} \text{proj}_{\mathcal{C}_{H_0(v)}}(u^{(\ell)}). \tag{3.3}$$

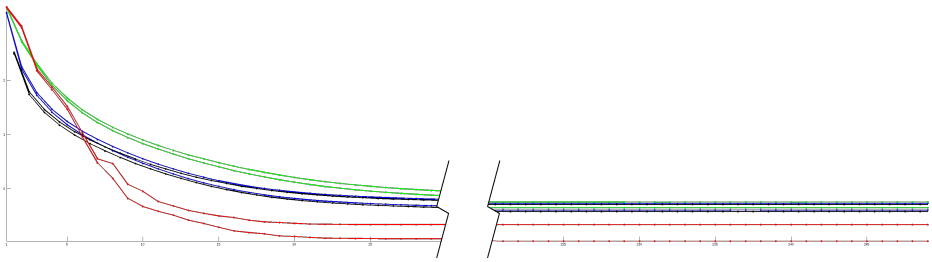


Fig. 3.1: Display of $\log(E(u^{(\ell)}|(\nu_{z_k})_k))$ as a function of ℓ for the three tested algorithms: (i) HB algorithm [34] (in blue, with a display of the half-iterations $\tilde{u}^{(\ell)}$ in black); (ii) gradient descent algorithm (in green); L-BFGS algorithm (in red). Each algorithm is represented with two curves, showing the minimum and maximum energy values obtained among 100 realization of the initialization $u^{(0)}$.

It then computes the wavelet coefficients of $\tilde{u}^{(\ell)}$ and matches them with those of v

$$\forall\, k = 1, \ldots, K, \quad z_k^{(\ell)} \overset{\text{def.}}{=} \mathrm{proj}_{\mathcal{C}_{H_k(v)}}(H_k(\tilde{u}^{(\ell)})). \tag{3.4}$$

The new iterates is then reconstructed from these coefficients $(z_k^{(\ell)})_{k=1}^K$

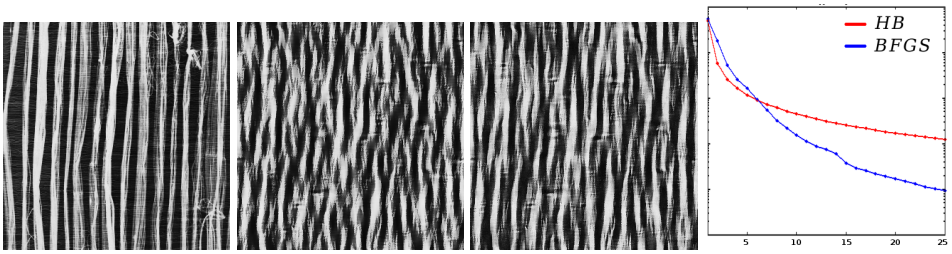
$$u^{(\ell+1)} \overset{\text{def.}}{=} H_{1,\ldots,K}^{+}\left(z_k^{(\ell)}\right)_{k=1}^{K} \tag{3.5}$$

where $H_{1,\ldots,K}^{+}$ is the pseudo-inverse of the wavelet transform $H_{1,\ldots,K}$ (i.e. the reconstruction which yields the image with the closest wavelet coefficients). Note that this pseudo inverse is also computed in linear time with a fast filter bank pyramid [34].

Variational re-formulation and improvement. The iterations of the three steps (3.3), (3.4), and (3.5) should not be mistaken for an alternated projections algorithm. Indeed, since the steerable pyramid is not orthogonal (it is a redundant transform), the two last steps do not correspond to an orthogonal projection. Using the fact that the transform $H_{1,\ldots,K}$ is a tight frame (i.e. it satisfies a conservation of energy), it can however be interpreted as computing alternated projections on the coefficients $(z_k^{(\ell)})_k$, see [62]. Another interpretation of this method is that it is in fact an alternated (i.e. alternatively on the pixel fidelity and then the wavelet fidelity) gradient descent of the energy (3.1) using a fixed step size, see [62].

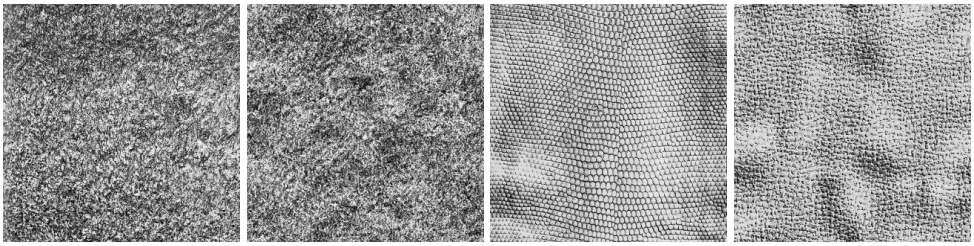
To obtain better results and a faster numerical scheme, it thus makes sense to replace this alternated gradient descent by the L-BGFS iterations (3.2) which integrates second order information about the statistical fidelity into the optimization scheme. The only missing ingredient to implement this descent is the computation of the adjoint ∂H_k^* of the derivative of the feature extractors. Since these are linear maps, one has $\partial H_k^* = H_k^*$, and these adjoints for $k = 1, \ldots, K$ are computed for all k with a fast filter bank, which, up to a diagonal re-scaling, is the same as the one computing $H_{1,\ldots,K}^{+}$.

Numerical illustrations. Figure 3.1 compares the decay of the energy (3.1) for HB algorithm [34] (iterates (3.3), (3.4), and (3.5)), a gradient descent with an manually tuned constant step size (corresponding to setting $L^{(\ell)}$ to be proportional to the identity matrix in (3.2)), and the L-BFGS algorithm. Since these algorithms are



Exemplar v HB algorithm L-BFGS $E(u^{(\ell)}|(\nu_{z_k})_k)$

Fig. 3.2: Synthesis results $u^{(\ell)}$ obtained after $\ell = 25$ iterations of HB algorithm and L-BFGS.



Exemplar v Synthesized Exemplar v Synthesized

Fig. 3.3: Two examples of synthesis output of the L-BFGS algorithm.

randomized by the choice of the initialization, we show the minimum and maximum energy values reached among 100 random white noise realizations for $u^{(0)}$, which allows one to get a clearer picture of the energy landscape explored by the methods. In these tests, the L-BFGS always yields lower values than the other algorithms; it has also a faster convergence speed. Note that HB algorithm (corresponding to an alternated gradient descent) converges faster than gradient descent, but the final state reached by H-B is only marginally better in term of energy than the one reached by gradient descent.

Figure 3.2 shows the resulting image $u^{(\ell)}$ computed by the HB [34] and the L-BFGS algorithms after $\ell = 25$ iterations. The vertical stripes in the exemplar image v give rise to long-range dependencies between the wavelet coefficients, which require many iterations to be set up properly. L-BFGS is able to generate them within 25 iterations, while the original algorithm of [34] requires more iterations to create them.

Let us finally note that while the L-BGFS approach brings some improvements with respect to the original HB algorithm, it suffers from the same drawback in term of texture modeling. Figure 3.3 shows two examples of syntheses obtained by the L-BFGS optimization method. This highlight the (well known) fact that methods based on first order statistics of multiscale decompositions are quite good at reproducing highly stochastic micro-textures (in the sense [27]) such as the exemplar on the left of the figure, but fail at generating geometric patterns, such as in the exemplar on the right. Synthesizing geometric features requires either to use higher order statistical constraints on the wavelet coefficients (see for instance [50]) or more complicated non-linear feature extractors, as exemplified in the next section.

3.3. Non-linear Sparse Features. To improve synthesis quality, we follow the path explored in [63] and propose to use non-linear feature extractors obtained through sparse decompositions. Sparse decompositions are able to better reproduce sparse features (such as edges) and also permit the use of dictionary learning methods to capture these features in an exemplar-driven manner, thus bridging the gap between the HB algorithm and copy-based methods (see Sect. 1.1 for an overview of these approaches).

Sparse coding feature extractor. Given a dictionary $D = (d_i)_{i=1}^N$ of N atoms d_i, a sparse approximation $\mathcal{Z}_D(h)$ of a signal h (which may be an image u or small patches extracted from it for instance) is commonly defined using a penalized ℓ^1 regularization

$$\mathcal{Z}_D(h) \in \arg\min_z \frac{1}{2}\|Dz - h\|^2 + \alpha\|z\|_1 \quad \text{where} \quad \|z\|_1 \stackrel{\text{def.}}{=} \sum_{i=1}^N |z_i|, \tag{3.6}$$

where the parameter α controls the sparsity of the coefficients. This corresponds to the celebrated Lasso problem [65], also known as basis pursuit [10]. Note that the solution of the optimization problem (3.6) is in general not unique, so we define $\mathcal{Z}_D(h)$ as being one of the minimizers.

It is important to realize that while the reconstruction operator $Dz = \sum_i z_i d_i$ is linear, the sparse coding operator $\mathcal{Z}_D$ is highly non-linear. More precisely, the non-smoothness of $\|\cdot\|_1$ creates a large number of zero coefficients in z, and this sparsity effect increases with α. It is precisely this non-linear effects that is important to drive the generation of sparse features (such as isolated dots or edges) in a synthesis output.

Computing the minimizer of (3.6) requires to solve a convex optimization problem. A popular way to approximate the solution of this optimization for large scale imaging problem is to use a so-called first order proximal scheme. We use in this section the Fast Iterative Soft Thresholding Algorithm (FISTA, [5]) which defines a sequence of iterates $\mathcal{Z}_D^{(\ell)}(h)$ converging to $\mathcal{Z}_D(h)$ as ℓ grows toward $+\infty$.

Synthesis with sparse features. We define a sparse synthesis algorithm using the L-BFGS iterations (3.2) to minimize (3.1) with a trivial pixel-domain extractor $H_0(u) = u$ and a (possibly decimated) set of K sparse features extractors. This is formalized as

$$\forall\, k = 1, \ldots, K, \quad H_k(u) \stackrel{\text{def.}}{=} S_k\Big(\mathcal{Z}_D^{(\ell)}(\Pi(u))\Big).$$

Here, Π and $(S_k)_k$ are fixed linear maps. The role of $S_k(z) = (z_i)_{i\in I_k}$ is simply to select a sub-class of coefficients indexed by I_k, associated to atoms $(d_i)_{i\in I_k}$ which are usually translations of the same template (e.g. for wavelet-type or translation invariant dictionaries). The role of Π is to act as a pre-processor. In the following, we explore two possibilities: Π being a patch-extractor to perform the sparse coding on small patches, and $\Pi = \mathrm{Id}$ so that one sparse codes the image u itself.

It is important to note that the feature is defined using the output $\mathcal{Z}_D^{(\ell)}$ of FISTA. It is an approximation of the original $\mathcal{Z}_D$, the latter being well defined mathematically but numerically unaccessible.

In order to implement iterations (3.2), one needs to compute the gradient ∇E using (2.7), which in turn necessitates to compute the adjoint of the derivative of the features extractors as follow

$$\partial H_k^*(z) = \Pi^*[\partial \mathcal{Z}_D^{(\ell)}]^*(S_k^*(z)).$$

The adjoint of the selector is simply

$$S_k^*(z)_i = \begin{cases} z_i & \text{if} \quad i \in I_k, \\ 0 & \text{otherwise,} \end{cases}$$

and Π^* is detailed in the two specific cases bellow.

It is important to realize that the adjoint of the differential of FISTA's output $[\partial \mathcal{Z}_D^{(\ell)}]^*$ is in general very different from the adjoint of the theoretical sparse coding operator $[\partial \mathcal{Z}_D]^*$. This is well documented in [17], where the authors argue that the derivative of sparse coding operators are unstable and computationally out of reach. In contrast, $\partial \mathcal{Z}_D^{(\ell)}$ and its adjoint are easy to compute using automatic differentiation technics, which is equivalent to formally differentiating the iterates of FISTA. We refer to [62] for more implementation details.

Dictionary learning methods. The resulting synthesis method is quite general because of the degree of freedom in the choice of the linear maps $(D, \Pi, (S_k)_k)$ and the regularization parameter α. It is beyond the scope of this paper to explore in full generality this method; we illustrate it in two specific settings bellow.

In these two examples, one first estimates D using standard dictionary learning technics (see for instance [21, 25, 43]) from the exemplar image v alone. One minimizes the sparse coding energy with respect to the dictionary D, i.e.

$$\min_{D \in \mathcal{D}} \min_z \left\{ \frac{1}{2} \|Dz - \Pi(v)\|^2 + \alpha \|z\|_1 \; ; \; \forall i, \|d_i\| \leq 1 \right\}. \tag{3.7}$$

Here $\mathcal{D}$ is a linear set of constraints (specified bellow in each numerical example) that imposes some structural property on the dictionary (such as translation invariance) and make the problem well-posed. The standard way [21, 25, 43] to find a stationary point of the (highly non-convex) energy (3.7) is to iteratively minimize with respect to D and to z since the energy is convex with respect to each of these parameters alone, even if it is not jointly convex.

Numerical illustration #1: patch-based dictionaries. A popular class of image processing methods (see Sect. 1.1 for a review) makes use of the redundancy between the patches of the image. This corresponds to the best known usage of dictionary-learning approaches that apply sparse coding to these so-called patches. It reduces the computational complexity as well as the number of degrees of freedoms to design the dictionary. This corresponds to defining Π as a patch extractor,

$$\Pi(u) = (\Pi_s(u))_{s \in \mathcal{S}} \quad \text{where} \quad \Pi_s(u) = (u_{s+t})_{t \in \{1,\dots,\tau\}^2}$$

where $s \in \mathcal{S}$ runs over a fixed set of extracting position (usually located on an uniform grid on the image domain) and $\tau \times \tau$ is the size of the patches. Note that the adjoint operator Π^* simply corresponds to reconstructing an image by tiling the patches and adding the overlapping parts:

$$\Pi^*((h_s)_{s \in \mathcal{S}})_i = \sum_{s+t=i} (h_s)_t.$$

In this setting, we impose the dictionary to handle each patch independently and in the same way. That is, we consider $Dz = (\bar{D} z_s)_{s \in \mathcal{S}}$ where $\bar{D} = (\bar{d}_k)_{k=1}^K$ is a "reduced" dictionary made of K atoms $d_k \in \mathbb{R}^{\tau^2}$. The coefficients are thus

$z = (z_s)_{s\in\mathcal{S}} \in \mathbb{R}^{|\mathcal{S}|\times K}$ where $z_s \in \mathbb{R}^K$ are the coefficients of the patch $\Pi_s(u)$. The set of coefficients $z = (z_s)_{s\in\mathcal{S}}$ is thus subdivided into K groups $\{1,\dots,N\} = I_1 \cup \dots \cup I_K$, each I_k corresponding to the $|\mathcal{S}|$ coefficients associated to a given atom d_k. This construction is usually the one considered in the literature [21, 25, 43]. The learning optimization (3.7) is carried directly over the reduced dictionary $\bar{D}$, hence making the overall problem tractable. It makes sense to design our statistical constraint model to impose the first order statistics of each of these group.

Numerical illustration #2: translation invariant dictionaries. In this part, we apply the sparse coding operator to the image itself, that is, we set $\Pi = \mathrm{Id}_P$, which is the identity map on the image space $\mathbb{R}^P$. It is not possible to train a generic unstructured dictionary D on the whole image $u \in \mathbb{R}^P$ because optimizing and applying the dictionary would be numerically intractable, and also because the learning problem from a single exemplar image v would not be well posed because of the too many degrees of freedom. The most natural constraint is to impose the dictionary to be translation-invariant, i.e. to constrain it to be obtained by translating a set of atoms $\{\varphi_k\}_{k=1}^K$. In this case, applying the dictionary to a set of weights $z = (z^1,\dots,z^K)$ is

$$Dz = \sum_{k=1}^{K} z^k \star \varphi_k \tag{3.8}$$

where $\star$ is the discrete 2-D convolution. Formula (3.8) defines a linear constraint $\mathcal{D}$ on the set of allowable dictionary. Note that similar shift-invariant constraints have been proposed in the dictionary learning literature, see for instance [8]. The energy (3.7) can be efficiently minimized directly with respect to the filters $(\varphi_k)_k$ using fast convolution computations as detailed in [8] for instance. This specific structure segments the set of coefficients in non-overlapping consecutive indexes $\{1,\dots,N\} = I_1 \cup \dots I_K$, so that the total number of coefficients is $N = KP$ where P is the number of pixels.

Numerical results. The synthesis results using the two setups presented above (Π being a patch extractor and $\Pi = \mathrm{Id}$) are shown in Fig. 3.4. In order to improve the synthesis results, following the previous work [64], we have added to the Wasserstein statistical loss defined in (2.6) a frequency matching term. More precisely, we minimize the function

$$F : u \mapsto E(u|(\nu_{z_k})_k) + \frac{\beta}{2}\||\hat{u}| - |\hat{v}|\|^2 \tag{3.9}$$

where $E(u|(\nu_{z_k})_k)$ is defined in (3.1) and the second term measures the fidelity to the modulus of the Fourier transform $\hat{v}$ of the exemplar image v.

Figure 3.4 shows several exemplar images, the syntheses obtained by minimizing (3.9) with the patch-based dictionary (described in the numerical illustration #1) and with the translation-invariant dictionary (described in the numerical illustration #2). In the patch-based case, the parameters are $\tau = 12$, $K = 192$ (redundancy around 1.5), $\alpha = 0.1$ and $\beta = 1$. In the translation-invariant case, we use $\tau = 12$ (size of each φ_k), $K = 5$, $\alpha = 2$, and $\beta = 1/16$. These results show that the switch from a fixed linear representation (as used in Section 3.2) to adaptive non-linear representations enables a more faithfully synthesis of geometrical textural patterns.

4. Gradient Statistics for Image Restoration. This section applies the Wasserstein loss defined in Sect. 2 to the problem of image restoration. As explained

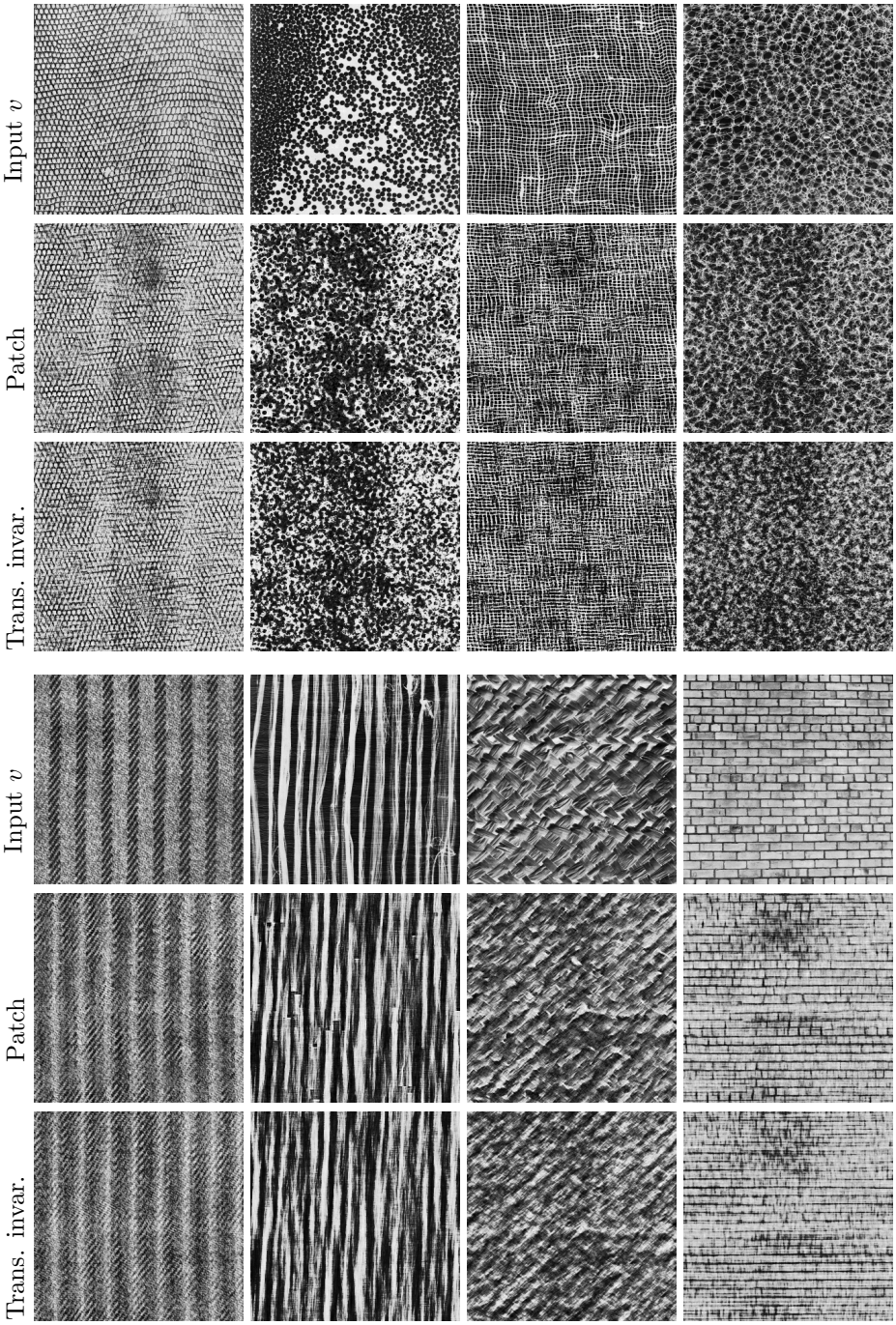


Fig. 3.4: Synthesized images using the sparse decomposition (3.9) from Sect. 3.3. The rows show respectively: the input image v, the synthesized image using the patch-based decomposition (case #1), the synthesized image using the translation-invariant decomposition (case #2).

in the introduction, the goal is to control the statistical properties of the restored image. As a case study, we consider the classical TV regularization. We choose to impose the statistics of the gradients of the image, since this controls both textured aspects and contrast [11]. In order to estimate the true distributions of gradients from the noisy observation alone, we assume that they follow some parametric model. The Wasserstein loss is then computed between the estimated distributions and the distributions of the denoised image. We will see that such a framework is efficient to prevent some defaults of the total variation. As a surprising by-product, we also observe that the Wasserstein statistical loss can actually replace the total variation term, therefore acting both as a fidelity and a regularization term.

4.1. Wasserstein Loss for Image Restoration. We consider the problem of recovering an approximation u of some unknown image u_0 from degraded noisy measurements

$$y = \Phi u_0 + \eta$$

where $\Phi : \mathbb{R}^P \mapsto \mathbb{R}^Q$ is a linear operator modeling the acquisition process and η is some additive noise. We assume that η is a realization of a white noise of mean 0 and variance $\sigma^2 \mathrm{Id}_Q$. In this section, we show applications to image denoising (corresponding to $\Phi = \mathrm{Id}$) and give an example of denoising (that is, Φ is a non-uniform down-sampling operator).

Variational problem. Following the general setting of variational estimators for inverse problems, we consider a recovery obtained by a minimization of the form

$$\min_u \ \{\lambda' R(u) + \lambda'' E(u|(\nu_{z_k})_k) \ ; \ u \in \mathcal{C}_\alpha\} \tag{4.1}$$

where $E(u|(\nu_{z_k})_k)$ is a statistical fidelity term (2.7) and $R(u)$ is a regularization term. The parameters $\lambda', \lambda'' \geq 0$ controls the trade-off between the statistical fidelity term and the regularization term. For the sake of normalization, we choose some $\lambda \in [0,1]$ to define $\lambda'' \overset{\text{def.}}{=} \frac{1-\lambda}{2K}$ and $\lambda' \overset{\text{def.}}{=} \frac{\lambda}{\sqrt{2}P}$ when using TV regularization (4.2). The constraint set

$$\mathcal{C}_\alpha \overset{\text{def.}}{=} \left\{ u \ ; \ \|\Phi u - y\|^2 \leq \alpha \sigma^2 Q \right\}$$

accounts for the presence of a Gaussian noise of variance σ^2, and the parameter α is typically chosen close to 1. In the following, we use the classical total variation (TV) regularization [57]

$$R(u) \overset{\text{def.}}{=} \|\nabla u\|_1 = \sum_i \sqrt{(\partial_{i,1}\ u)^2 + (\partial_{i,2}\ u)^2} \tag{4.2}$$

where $(\partial_{i,1} u, \partial_{i,2} u)$ is the usual forward finite difference approximation of the gradient of u at pixel i. While the TV penalty is not the state of the art in image denoising and restoration (see Sect. 1.1 for a review), it is a good benchmark to exemplify the main features of our approach.

As detailed in Sect. 1.1, traditional variational approaches and in particular TV regularization have difficulties to restore fine-scale textural details. To account for such high frequency content, it thus makes sense to study and to impose the statistical distribution of local derivatives of the image. To define the statistical fidelity

$E(u|(\nu_{z_k})_k)$, we thus use gradient-domain linear features extractors H_k that corresponds to K directional derivatives

$$H_k(u) \stackrel{\text{def.}}{=} \big(\cos(k\pi/K)\partial_{i,1}\ u + \sin(k\pi/K)\partial_{i,2}\ u \big)_{i=1}^{P} \quad \forall\, k = 1 \dots K. \tag{4.3}$$

In the numerical simulations, we use $K = 4$ directional derivatives.

A delicate issue to be able to apply the variational method (4.1) is to define the distribution $(\nu_{z_k})_{k=1}^K$ from the degraded observation y alone. We detail how we achieve this in the two considered scenarios (denoising and super-resolution) in Sect. 4.2.

Minimization algorithm. The optimization (4.1) is both non-convex (because of the Wasserstein loss $E(\cdot|(\nu_{z_k})_k)$) and non-smooth (because of the ℓ^1 norm in the definition of the TV regularization (4.2)). We apply the first order proximal algorithm of [14]. While this algorithm is originally designed to handle convex functional, it has been applied with success to non-convex functionals [15] although there is no proof of convergence in this case. In our setting, we also reached the conclusion that this algorithm was able to converge even in the presence of a smooth non-convex term in the energy to be minimized.

The algorithm of [14] minimizes a function of the form $F(u) + G(u) + H(L(u))$ where F is a smooth function, L is a linear operator, and (G, H) are possibly non-smooth functions for which one is able to compute the so-called proximal operator. We refer to [14] for the detailed exposition of the iterations of the algorithm.

In our case, we use the splitting $F(u) = E(u|(\nu_{z_k})_k)$, $G(u)$ is the indicator of the constraint $\mathcal{C}_\alpha$, $L = (\partial_1, \partial_2)$ is the discretized gradient operator, and $H(p_1, p_2) = \sum_i \sqrt{p_{1,i}^2 + p_{2,i}^2}$ is the ℓ^1 norm of 2-D vector fields. The proximal operators of G and H are respectively the orthogonal projection on $\mathcal{C}_\alpha$ and the soft thresholding operator (see [14] for more details). The gradient of F is computed using (2.7), and making use of the fact that

$$\partial H_k(u)^* = H_k^* = \cos(k\pi/K)\partial_1^* + \sin(k\pi/K)\partial_2^*$$

where the adjoints $(\partial_1^*, \partial_2^*)$ of the forward derivatives are minus the backward derivatives.

4.2. Estimation of the Prior Distributions. In contrast to the texture synthesis problem studied in Sect. 3, the estimation of the target distributions $(\nu_{z_k})_k$ is a delicate problem for restoration applications since one does not have direct access to a clean image. Ideally, each ν_{z_k} should be a good approximation of the (unknown) distribution $\nu_{H_k(u_0)}$ associated with the image to recover. The estimation of ν_{z_k} from the noisy observations y is in itself a difficult statistical problem: we do not aim at covering it in full generality.

To make this estimation well-posed, a common practice is to assume some parametric form for the distributions. To account for the sparsity of the derivatives of natural images, a popular parametric model [44] is the Generalized Gaussian Distribution (GGD). This model has been successfully used in particular for image restoration [59] or texture modeling [66]. We thus impose the discrete distribution ν_{z_k} to be the best match of a continuous GGD density g_{p_k,s_k} of parameter (p_k, s_k). The GGD density is

$$g_{p,s}(z) \stackrel{\text{def.}}{=} \frac{1}{C_{p,s}} e^{-\frac{|z|^p}{s^p}}$$

where $C_{p,s}$ is a normalizing constant, s controls the variance of the distribution, and p is the shape parameter (controlling the sparsity of the distribution, e.g. $p = 2$ corresponds to a Gaussian distribution).

For each k, the estimation of ν_{z_k} is thus achieved in two steps: (i) estimation of the parameters (p_k, s_k) from the input y ; (ii) estimation of the quantized positions $z_k \in \mathbb{R}^N$ from the estimated (p_k, s_k).

Parameters estimation. We first treat the case where $\Phi = \mathrm{Id}$ so that $y = u_0 + \eta$. One thus has $H_k(y) = H_k(u_0) + \eta_k$ where $\eta_k \overset{\text{def.}}{=} H_k(\eta)$ is the realization of a (colored) Gaussian noise for which each entry has a variance $2\sigma^2$ (by linearity of the differentiation operator).

The known distribution of the coefficients of $H_k(y)$ is thus equal to a convolution between the unknown distributions of $H_k(u_0)$ and a Gaussian distribution of variance $2\sigma^2$. The estimation problem is thus a well known deconvolution problem, which has been studied extensively in the case of a 1-D parametric GGD distribution. We use the method described in [62] which performs a matching between the cumulants of the GGD + Gaussian noise and the empirical cumulants computed from $H_k(y)$. The cumulants of order 2 and 4 (the two first non-zero cumulants) yield two equations whose unknown are (p_k, s_k), hence providing an estimation of these parameters.

In the case of inpainting, Φ is a diagonal masking operator, that is, some pixels are unknown. The parameters (p_k, s_k) of the GGD modeling $H_k(y)$ are obtained in the same way as before; the only difference is that the empirical cumulants of $H_k(y)$ are computed using only the subset of known values of the gradient $H_k(y)$.

Thanks to the homogeneity property of the cumulants, the deblurring case could also be handled in a similar way. However, this approach rapidly becomes unstable when the convolution kernel is getting bigger. This is because the gradients are highly damaged as the image is blurred, making their distributions difficult to estimate in the presence of noise.

Quantized positions estimation. The quantized positions z_k should be chosen to minimize the discrepancy between the continuous distributions G_{p_k,s_k} and the discrete distribution ν_{z_k}. For 1-D distributions, the solution to this optimal quantization problem is well-known and corresponds to the posterior expectation on each quantile of the distributions:

$$\forall\, i = 1 \dots N, \quad (z_k)_i \overset{\text{def.}}{=} N^{-1} \int_{t=t_{i-1}}^{t_i} t \, dg_{p_k,s_k}(t)$$

where $t_i \overset{\text{def.}}{=} G^{-1}_{p_k,s_k}(i/N)$ are the quantiles and G_{p_k,s_k} the cumulative distribution of the GGD. Then

$$\forall\, z \in \mathbb{R}^N, \quad \mathcal{W}(\nu_z, g_{p_k,s_k})^2 = \mathcal{W}(\nu_z, \nu_{z_k})^2 + \mathcal{W}(\nu_{z_k}, g_{p_k,s_k})^2$$

as proved in [62]. Hence the Wasserstein distance to the GGD is equal to the distance to the discretization z_k, up to an additive constant which has no impact on the minimization problem.

4.3. Numerical Results. This section presents several restoration results obtained using our approach in the case of denoising. We then compare such results to other approaches and in particular to NL-Bayes [37] and to the gradient histogram matching [11]. We also study the ability of our approach to preserve the gradient and

the color distributions, in contrast with other methods. More results are available in [62].

To be efficient on complex scenes, our approach would require a former segmentation step such as the one presented in [11]: this way, we would be able to estimate a different GGD model for each region of the image, and thus have a texture term which is adaptive with respect to each region of the image.

We first present the result of our algorithm on several uniform textures. In the following experiments, the fidelity constant of (4.1) is set to $\alpha = 0.7$: this value yields a visually good trade-off between denoising and over-smoothing in the standard TV case. The optimization algorithm is stopped after 200 iterations, which we found sufficient for the considered problem. When not specified, the size of the images is $P = 128 \times 128$ pixels and their values are in $[0, 1]$.

In the set of experiments given in Fig. 4.1, we present from left to right: the original noise-free image; the observed image, contaminated by a white Gaussian noise of standard deviation 15% of the image amplitude, i.e. $\sigma = 0.15$ for images whose dynamic is $[0, 1]$; the result of the TV denoising algorithm [57], which is a special case of our algorithm with $\lambda = 1$; the result of our algorithm with only the texture fidelity term (without TV), which corresponds to the case $\lambda = 0$; the result of our algorithm with a trade-off $0 < \lambda < 1$ which we called "hybrid" method, and in particular with $\lambda = 1‰ = 1/1000$ which is experimentally well suited on our set of images.

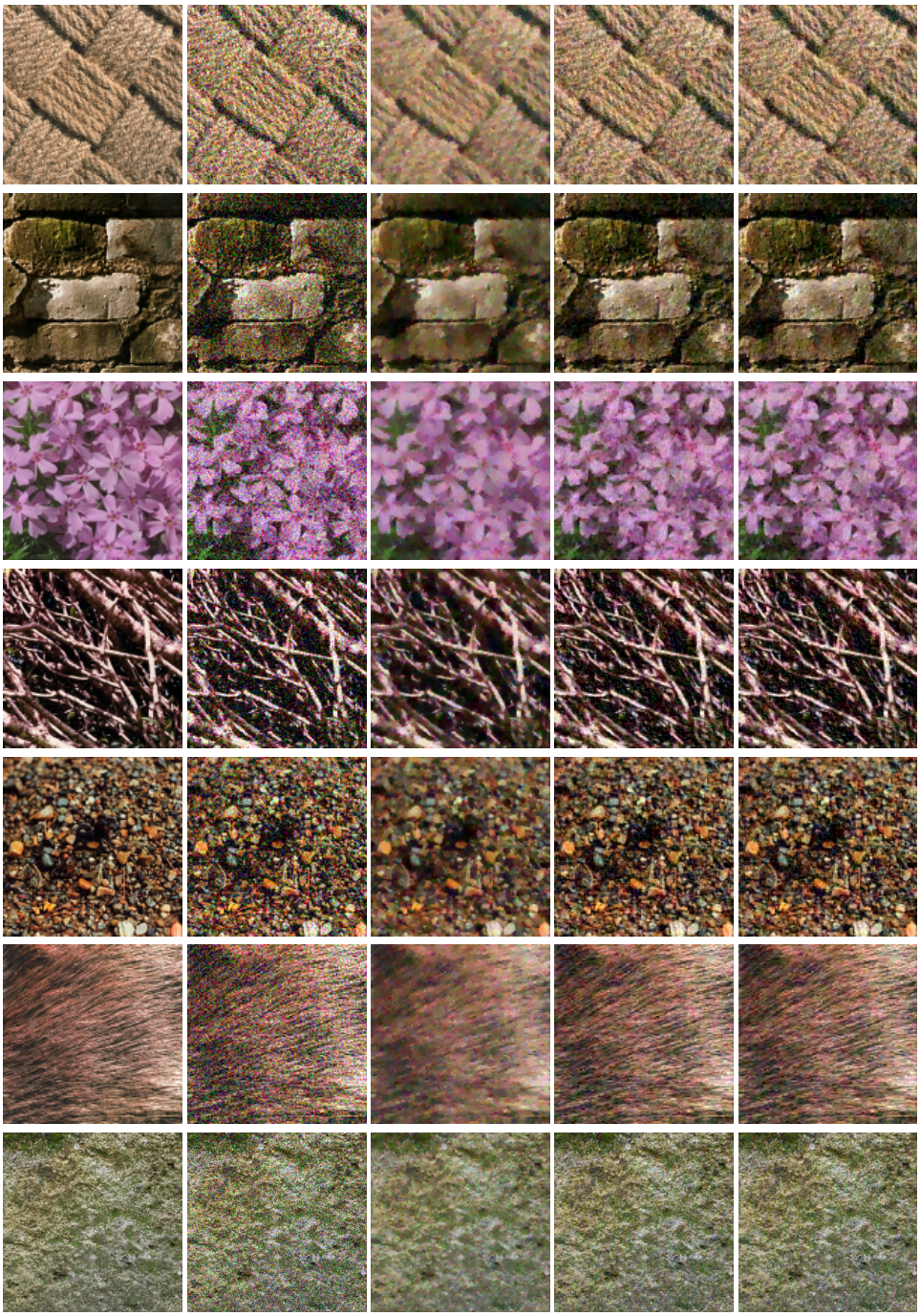
The TV approach (3rd column) removes most of the textures, resulting in smooth images, sometimes with a cartoon aspect. On the contrary, the approach corresponding to $\lambda = 0$ (4th column) does not destroy any textural content at the cost of some residual noise. It is interesting to note that in this experiment the texture term (Wasserstein statistical loss) is used without additional regularization term. The hybrid approach (5th column) make use of a TV regularization term to eliminate more noise. However, the pure Wasserstein loss ($\lambda = 0$) approach and the hybrid one produce very similar results on several images.

A few examples of inpainting are shown on Fig. 4.2: in addition to a white Gaussian noise of standard deviation $\sigma = 0.1$, 40% of the pixels has been lost (i.e. the degradation operator Φ is a non-uniform down-sampling). We do not focus more on this case here, instead we refer to the former results presented in [62].

Influence of the trade-off parameter. Figure 4.3 further illustrates the effect of the parameter λ on the denoised image. The parameter varies from $\lambda = 1$ (TV algorithm) to $\lambda = 0$ (Wasserstein term only, no TV term). The visually optimal result is usually obtained as a trade-off between these two terms, since the TV term over-smooths the image and destroys textures, while the GGD term only does not always provide a strong enough regularization. As stressed in the previous experiment, the TV is not always necessary and the Wasserstein term alone may perform well, especially in the case of textures with fine details.

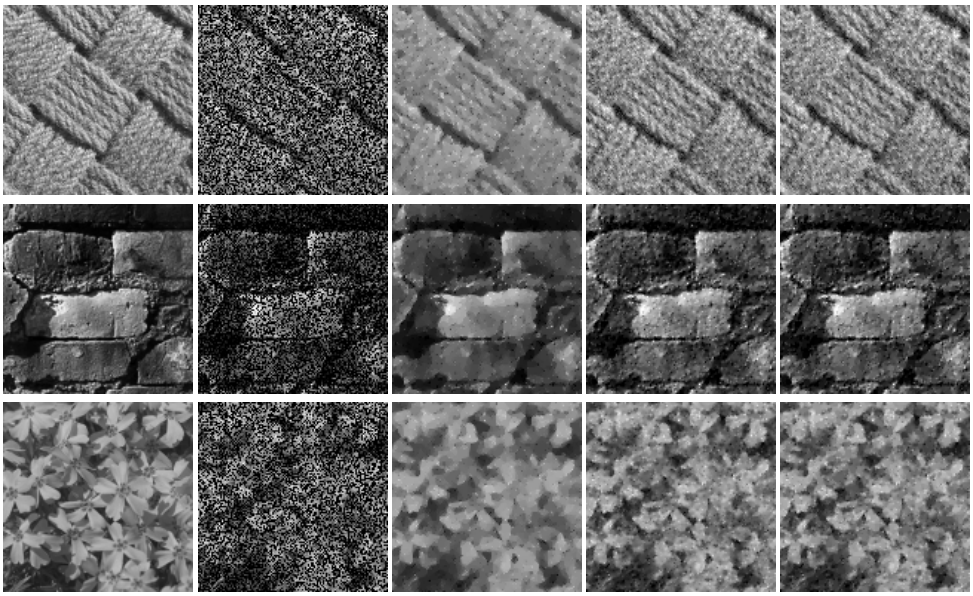
Comparison with state of the art approaches. We now compare our results to other denoising approaches: first the state-of-the-art denoising algorithm NL-Bayes [37] and the inverse problem algorithm PLE [72]; then we display some comparisons with the results from [11] which introduced the idea of preserving the gradient distribution to restore images.

Patch-based approaches. Figure 4.4 compares our results to both NL-Bayes [37] and PLE [72]. Online demos [36, 68] are available for these two algorithms. They



Original (noise-free)	Noisy, $\sigma = 0.15$	TV ($\lambda = 1$)	GGD ($\lambda = 0$)	Hybrid, $\lambda = 1‰$

Fig. 4.1: Denoising of images using the TV + gradient distribution fidelity term. From left to right: original noise-free image, noisy observation, result of TV denoising ($\lambda'' = 0$ in (4.1)), result of the Wasserstein term without TV ($\lambda' = 0$ in (4.1)), combination of TV and Wasserstein term.



Original (noise-free)	Degraded, 40% missing, $\sigma = 0.1$	TV ($\lambda = 1$)	GGD ($\lambda = 0$)	Hybrid, $\lambda = 1‰$

Fig. 4.2: Inpainting of noist images using the TV + gradient distribution fidelity term. From left to right: original image, noisy observation with missing pixels, result of TV inpainting ($\lambda' = 0$ in (4.1)), result of the Wasserstein term without TV ($\lambda' = 0$ in (4.1)), combination of TV and Wasserstein term.

both produce a final image that is highly denoised. Our approach does not remove as much noise, but it preserves more details and produces a sharper image.

A reason for this behavior is that NL-Bayes and PLE use the self-similarity of the image (i.e. its intrinsic redundancy): they assume that the patches of the images provide a redundant information. They perform denoising by enforcing the intra-patch similarities while removing the singular elements of the patches. Although these methods produce state-of-the-art results, some details are lost as a result of an increase of the self-similarity of the image. Our texture fidelity term enforces the gradient distribution to follow the estimated law: it prevents the image to be smoother than expected by preserving the amount of granularity of the image.

Comparison with the approach of [11]. We compare on Fig. 4.5 our results to the ones from [11]. The authors of the latter paper propose to enforce the gradient distribution using an approach that is different from ours, based on a Kullback-Leibler divergence or alternatively on an accumulation of deviation terms (see [11] for more details). They argue that these terms succeed in imposing the distribution, however the choice of these terms and their effect are purely empirical. In comparison, the Wasserstein distance is a robust mathematical tool permitting us to set the gradient distribution without requiring additional cost functions. Since no code is available for the approach from [11], the images shown on Fig. 4.5 are directly extracted from the PDF version of [11], hence they may suffer from JPG compression artifacts. Figure 4.5

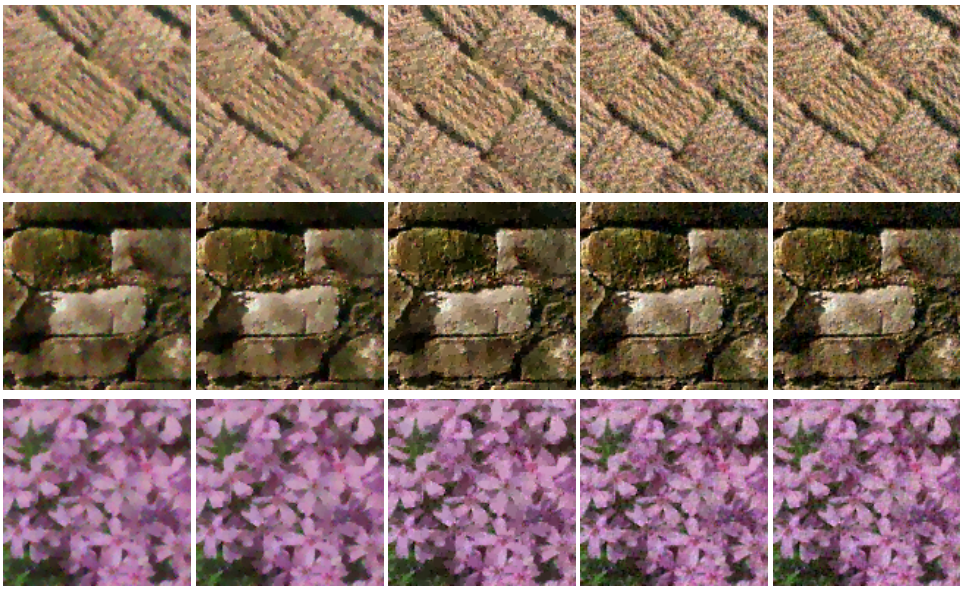


Fig. 4.3: Images denoising using the Wasserstein term in the cost function (4.1) for several values of the trade-off parameter λ. The model varies from the TV regularization ($\lambda = 1$, left) to Wasserstein term alone ($\lambda = 0$, right).

shows on the first row the original image and the segmentation from [11]; the following rows show, from left to right: a crop of the original image, a noisy version, a crop of the denoising result of [11], the result of our algorithm using only the texture fidelity term (i.e. $\lambda = 0$), and using the parameter $\lambda = 1‰$. Note that instead of using the noisy images of [11], we generate a noise with the same variance because the images extracted from the PDF have a non-Gaussian noise due to the JPG compression. Note also that [11] performs a segmentation (upper right image on Fig. 4.5); to compare our results to their approach, we simply extracted sub-images in each region.

The authors of [11] claim that their approach does not outperform the state-of-the-art in image denoising but is better at preserving textures. They support this argument by a massive user-study. Our approach removes less noise but is more texture-preserving, as may be seen in Fig. 4.5.

We emphasize that our texture-preserving term is generic and can easily be adapted to other variational approaches. We demonstrate its effect along with the TV variational approach because it is both simple and used in several contexts (see Sect. 1.1 or references in [40]).

Histograms of denoised images. We propose in this sub-section a closer look at the statistics of the denoised images: we analyze the resulting gradients and gray-level distributions, and then the noise removed by the algorithms.

Figure 4.6 shows the histograms of the gray-level and of the (horizontal and vertical) gradients of the input image and of the denoised images obtained by several algorithms.

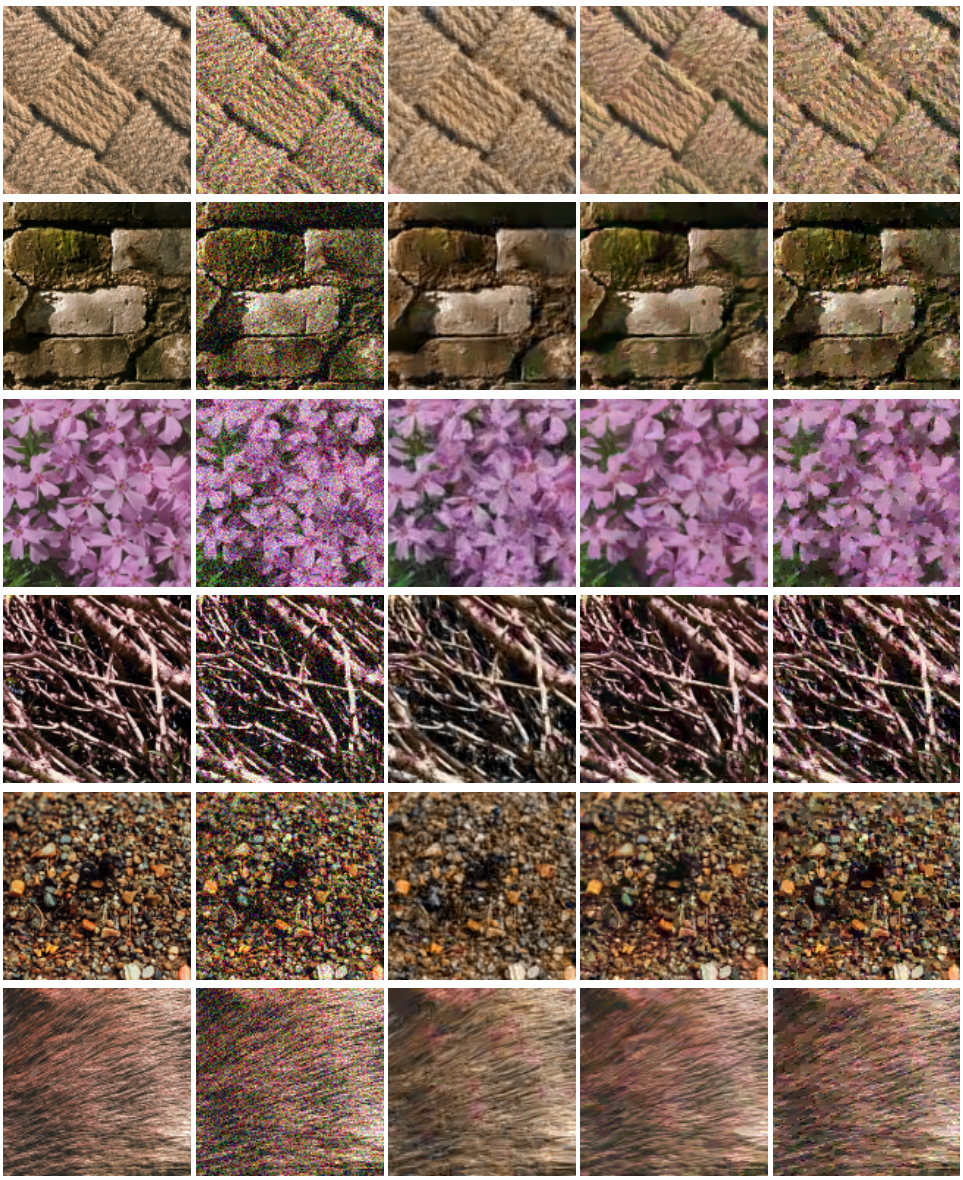


Fig. 4.4: Comparison to state-of-the-art denoising approaches on color images. From left to right: original image, noisy observation, denoising using the Piecewise Linear Estimator [72], denoising using the NL-Bayes algorithm [37], denoising using our approach (with $\lambda = 1/1000$). The average PSNR are 23.5 dB for PLE, 23.8 dB for NL-Bayes, and 22.7 for us. Although the PSNR is lower, our approach is better at preserving the texture and details.

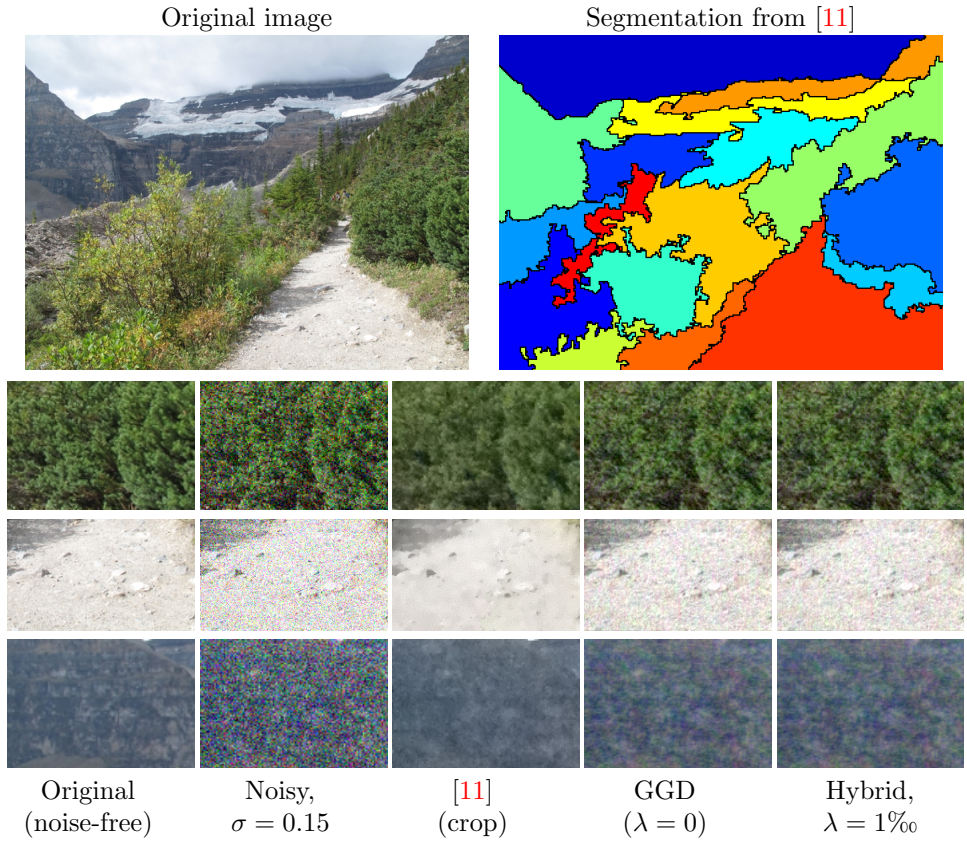


Fig. 4.5: Comparison to [11]. Top row: original image (left) and the segmentation from [11] into homogeneous texture regions (right). Next rows, from left to right: crop of the original image; noisy image; result of [11] (on the segmented image, then cropped); result using our texture fidelity term only (i.e. $\lambda = 0$, run on the cropped image); result of our hybrid approach (with $\lambda = 1/1000$, run on the cropped image). Note that the input image and the results of [11] were only available as JPG images. Note also that we launch our algorithm on the cropped images.

Gray-level histogram.. It shows that our algorithm preserves the gray-level distribution of the image; as a particular case, it also preserves the dynamic and the contrast of the image, which is perceptually important. On the opposite, both TV and NL-Bayes tends to compress the dynamic of the image and to diminish the contrast. This is clearly visible on the fur image in the previous figures. Imposing the gray-level histogram is proposed in [61] through a histogram fidelity term based on the Wasserstein distance.

Gradient histograms. They show that our texture fidelity term succeed in preserving the gradient distributions of the noise-free image, which is responsible for the texture preservation. The TV term tends to produce a piecewise-constant image, whose gradient histograms have a large peak around zero; the NL-Bayes approach smoothes the image and makes the gradient histogram narrower, resulting in a less contrasted image and a loss of details; both TV and NL-Bayes produce an image

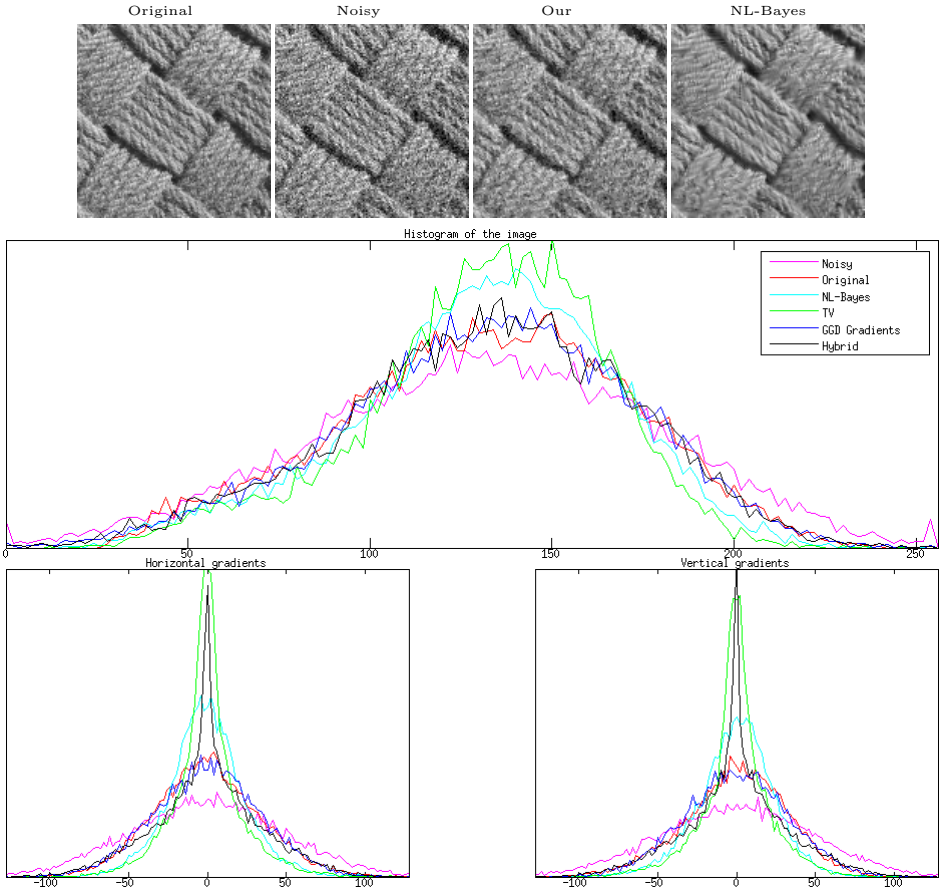


Fig. 4.6: Comparison of the histograms of gradients (bottom) and of pixel values (middle) for several images (top). The considered images are: original image u (red, 1^{st} top image), observed image v (pink, 2^{nd} top image), denoised by TV ($\lambda = 1$, green), denoised by our texture fidelity term only ($\lambda = 0$, blue), denoised by our hybrid approach ($\lambda = 1/1000$, black, 3^{rd} top image), and denoised by NL-Bayes ([37], blue, 4^{th} top image).

whose gradient histograms have too thin tails. Our hybrid algorithm respects the heavy tails of the distribution to preserve details and textures as much as possible, while producing a small peak around zero as the result of the TV regularization.

5. Conclusion. In this paper, we have introduced a generic framework to impose statistical constraints in variational problems routinely encountered in imaging sciences. We show in particular applications to image synthesis, denoising and super-resolution. We believe that a key aspect and contribution of our work is the use of a Wasserstein loss between discrete probability distributions, which is both simple to use and robust. It is of course possible to consider more advanced image processing problems, and we refer for instance to [54] where the same framework is used to perform texture mixing by minimizing a sum of Wasserstein losses learned from several input exemplars.

The proposed approach is generic and could be associated with other regularization terms for image restoration, for instance the non-local variational approaches from [23, 30]. It can also act as a post-processing step, possibly in combination with approaches such as NL-Bayes [37]. Last, as mentioned above, the approach can be generalized to other restoration tasks than denoising. While the extension to interpolation or super-resolution problems is relatively easy to develop, other tasks such as image deblurring may face some difficult estimation problems.

Acknowledgements. The work of Gabriel Peyré has been supported by the European Research Council (ERC project SIGMA-Vision).

REFERENCES

[1] Cecilia Aguerrebere, Yann Gousseau, and Guillaume Tartavel. Exemplar-based texture synthesis: the Efros-Leung algorithm. *Image Processing On Line*, 3:223–241, October 2013.

[2] Pablo Arias, Gabriele Facciolo, Vicent Caselles, and Guillermo Sapiro. A variational framework for exemplar-based image inpainting. *International Journal of Computer Vision*, 93(3):319–347, 2011.

[3] Jean-François Aujol, Gilles Aubert, Laure Blanc-Féraud, and Antonin Chambolle. Image decomposition into a bounded variation component and an oscillating component. *Journal of Mathematical Imaging and Vision (JMIV)*, 22(1):71–88, 2005.

[4] Suyash P. Awate and Ross T. Whitaker. Unsupervised, information-theoretic, adaptive image filtering for image restoration. *Transactions on Pattern Analysis and Machine Intelligence*, 28(3):364–376, March 2006.

[5] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *Journal on Imaging Sciences*, 2(1):183–202, 2009.

[6] Peter Blomgren, Tony F. Chan, Pep Mulet, and Chak-Kuen Wong. Total variation image restoration: numerical methods and extensions. In *International Conference on Image Processing (ICIP)*, volume 3, pages 384–384. IEEE, 1997.

[7] Nicolas Bonneel, Julien Rabin, Gabriel Peyré, and Hanspeter Pfister. Sliced and radon wasserstein barycenters of measures. *Journal of Mathematical Imaging and Vision (JMIV)*, pages 1–24, 2014.

[8] Hilton Bristow, Anders Eriksson, and Simon Lucey. Fast convolutional sparse coding. In *Proc. CVPR*, pages 391–398. IEEE, 2013.

[9] Antoni Buades, Bartomeu Coll, and Jean-Michel Morel. A non-local algorithm for image denoising. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 60–65. IEEE, June 2005.

[10] Scott S. Chen, David L. Donoho, and Michael A. Saunders. Atomic decomposition by basis pursuit. *Journal on Scientific Computing*, 20(1):33–61, 1998.

[11] Taeg Sang Cho, Charles Lawrence Zitnick, Neel Joshi, Sing Bing Kang, Richard Szeliski, and William T. Freeman. Image restoration by matching gradient distributions. *Transactions on Pattern Analysis and Machine Intelligence*, 34(4):683–694, April 2012.

[12] Wanhyun Cho, SeongChae Seo, and Jinho You. Edge-preserving denoising method using variation approach and gradient distribution. In *International Conference on Big Data and Smart Computing (BIGCOMP)*, pages 139–144, January 2014.

[13] Ronald R. Coifman and David L. Donoho. Translation-invariant de-noising. In *Wavelets and Statistics*, volume 103 of *Lecture Notes in Statistics*, pages 125–150. Springer, 1995.

[14] Laurent Condat. A primal–dual splitting method for convex optimization involving lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications*, 158(2):460–479, 2013.

[15] Laurent Condat and Akira Hirabayashi. Cadzow denoising upgraded: A new projection method for the recovery of dirac pulses from noisy linear measurements. *Sampling Theory in Signal and Image Processing*, 14(1):17–47, 2015.

[16] Kostadin Dabov, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian. Image denoising by sparse 3-d transform-domain collaborative filtering. *Transactions on Image Processing*, 16(8):2080–2095, August 2007.

[17] Charles Deledalle, Samuel Vaiter, Gabriel Peyré, and Jalal Fadili. Stein unbiased gradient estimator of the risk (sugar) for multiple parameter selection. *SIAM Journal on Imaging Sciences*, 7(4):2448–2487, 2014.

[18] David L. Donoho and Jain M. Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.

[19] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *International Conference on Computer Vision (ICCV)*, volume 2, pages 1033–1038. IEEE, 1999.

[20] Mireille El Gheche, Jean-François Aujol, Yannick Berthoumieu, and Charles-Alban Deledalle. Texture reconstruction guided by the histogram of a high-resolution patch. Technical report, 2016.

[21] Michael Elad and Michal Aharon. Image denoising via learned dictionaries and sparse representation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 895–900. IEEE, June 2006.

[22] Michael Elad, Peyman Milanfar, and Ron Rubinstein. Analysis versus synthesis in signal priors. *Inverse problems*, 23(3):947, 2007.

[23] Abderrahim Elmoataz, Olivier Lezoray, and Sébastien Bougleux. Nonlocal discrete regularization on weighted graphs: a framework for image and manifold processing. *Transactions on Image Processing*, 17(7):1047–1060, July 2008.

[24] Kjersti Engan, Sven O. Aase, and John Hakon Husoy. Method of optimal directions for frame design. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 5, pages 2443–2446. IEEE, 1999.

[25] Kjersti Engan, Sven Ole Aase, and John Hakon Husoy. Method of optimal directions for frame design. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 5, pages 2443–2446 vol.5, 1999.

[26] Sira Ferradans, Nicolas Papadakis, Gabriel Peyré, and Jean-Francois Aujol. Regularized discrete optimal transport. *SIAM Journal on Imaging Sciences*, 7(3):1853–1882, 2014.

[27] Bruno Galerne, Yann Gousseau, and Jean-Michel Morel. Random phase textures: Theory and synthesis. *Transactions on Image Processing*, 20(1):257–267, January 2011.

[28] Bruno Galerne, Ares Lagae, Sylvain Lefebvre, and George Drettakis. Gabor noise by example. *Transactions on Graphics*, 31(4):73:1–73:9, July 2012.

[29] Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. Texture synthesis using convolutional neural networks. In *Advances in Neural Information Processing Systems 28*, May 2015.

[30] Guy Gilboa and Stanley Osher. Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028, 2008.

[31] Yuanhao Gong and Ivo F. Sbalzarini. Gradient distribution priors for biomedical image processing. *arXiv preprint arXiv:1408.3300*, 2014.

[32] Rémi Gribonval, Volkan Cevher, and Mike E. Davies. Compressible distributions for high-dimensional statistics. *Transactions on Information Theory*, 58(8):5016–5034, August 2012.

[33] Frédéric Guichard and François Malgouyres. Total variation based interpolation. In *European Signal Processing Conference (EUSIPCO)*, volume 3, pages 1741–1744, September 1998.

[34] David J. Heeger and James R. Bergen. Pyramid-based texture analysis/synthesis. In *Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 229–238. ACM, 1995.

[35] Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. Texture optimization for example-based synthesis. *Transactions on Graphics*, 24(3):795–802, July 2005.

[36] Marc Lebrun, Antoni Buades, and Jean-Michel Morel. Implementation of the "non-local bayes" (NL-Bayes) image denoising algorithm. *Image Processing On Line*, 3:1–42, 2013.

[37] Marc Lebrun, Antoni Buades, and Jean-Michel Morel. A nonlocal bayesian image denoising algorithm. *Journal on Imaging Sciences*, 6(3):1665–1688, 2013.

[38] Sylvain Lefebvre and Hugues Hoppe. Parallel controllable texture synthesis. *Transactions on Graphics*, 24(3):777–786, July 2005.

[39] Adrian S. Lewis and Michael L. Overton. Nonsmooth optimization via quasi-newton methods. *Mathematical Programming*, 141(1–2):135–163, October 2013.

[40] Cécile Louchet and Lionel Moisan. Posterior expectation of the total variation model: Properties and experiments. *Journal on Imaging Sciences*, 6(4):2640–2684, 2013.

[41] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[42] Yang Lu, Song-chun Zhu, and Ying Nian Wu. Learning frame models using cnn filters. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[43] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro. Online learning for matrix factorization and sparse coding. *Journal of Machine Learning Research*, 11:19–60, March 2010.

[44] Stéphane Mallat. A theory for multiresolution signal decomposition: the wavelet representation.

Transactions on Pattern Analysis and Machine Intelligence, 11(7):674–693, July 1989.

[45] Mila Nikolova. Model distortions in bayesian MAP reconstruction. *Inverse Problems and Imaging*, 1(2):399–422, 2007.

[46] Jorge Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.

[47] Bruno A. Olshausen and David J. Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381:607–609, June 1996.

[48] Michael L. Overton. HANSO: Hybrid algorithm for non-smooth optimization, 2010.

[49] Gabriel Peyré, Sebastian Bougleux, and Laurent D. Cohen. Non-local regularization of inverse problems. *Inverse Problems and Imaging*, 5(2):511–530, 2011.

[50] Javier Portilla and Eero P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70, 2000.

[51] Javier Portilla, Vasily Strela, Martin J. Wainwright, and Eero P. Simoncelli. Image denoising using scale mixtures of gaussians in the wavelet domain. *Transactions on Image Processing*, 12(11):1338–1351, November 2003.

[52] Michael J. D. Powell. Some global convergence properties of a variable metric algorithm for minimization without exact line searches. *Nonlinear programming*, 9:53–72, 1976.

[53] Julien Rabin and Gabriel Peyré. Wasserstein regularization of imaging problem. In *International Conference on Image Processing (ICIP)*, pages 1541–1544. IEEE, September 2011.

[54] Julien Rabin, Gabriel Peyré, Julie Delon, and Marc Bernot. Wasserstein barycenter and its application to texture mixing. *Proc. SSVM'12*, 6667:435–446, May 2012.

[55] Justin K. Romberg, Hyeokho Choi, and Richard G. Baraniuk. Bayesian tree-structured image modeling using wavelet-domain hidden markov models. *Transactions on Image Processing*, 10(7):1056–1068, July 2001.

[56] Stefan Roth and Michael J. Black. Fields of experts: A framework for learning image priors. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 860–867. IEEE, June 2005.

[57] Leonid I. Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1–4):259–268, 1992.

[58] Eero P. Simoncelli. Bayesian denoising of visual images in the wavelet domain. In *Bayesian inference in wavelet-based models*, volume 141 of *Lecture Notes in Statistics*, pages 291–308. Springer, 1999.

[59] Eero P. Simoncelli and Edward H. Adelson. Noise removal via bayesian wavelet coring. In *International Conference on Image Processing (ICIP)*, volume 1, pages 379–382. IEEE, September 1996.

[60] Andrew Stuart. Inverse problems: A bayesian perspective. *Acta Numerica*, 19:451–559, 5 2010.

[61] Paul Swoboda and Christoph Schnörr. Convex variational image restoration with histogram priors. *Journal on Imaging Sciences*, 6(3):1719–1735, 2013.

[62] Guillaume Tartavel. *Variational Models for Textures: Applications to Synthesis and Restoration*. PhD thesis, Telecom ParisTech, 2015.

[63] Guillaume Tartavel, Yann Gousseau, and Gabriel Peyré. Constrained sparse texture synthesis. In *International Conference on Scale Space and Variational Methods in Computer Vision (SSVM)*, volume 7893 of *Lecture Notes in Computer Science*, pages 186–197. Springer, June 2013.

[64] Guillaume Tartavel, Yann Gousseau, and Gabriel Peyré. Variational texture synthesis with sparsity and spectrum constraints. *Journal of Mathematical Imaging and Vision (JMIV)*, pages 1–21, November 2014.

[65] Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society (Series B)*, pages 267–288, 1996.

[66] Gert Van de Wouwer, Paul Scheunders, and Dirk Van Dyck. Statistical texture characterization from discrete wavelet representations. *Image Processing, IEEE Transactions on*, 8(4):592–598, 1999.

[67] Cédric Villani. *Topics in Optimal Transportation*. Graduate studies in mathematics. American Mathematical Society, 2003.

[68] Yi-Qing Wang. The implementation of SURE guided piecewise linear image denoising. *Image Processing On Line*, 3:43–67, 2013.

[69] Li-Yi Wei and Marc Levoy. Fast texture synthesis using tree-structured vector quantization. In *Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 479–488. ACM, 2000.

[70] Johachim Weickert. *Anisotropic Diffusion in Image Processing*. Teubner-Verlag, 1998.

[71] Oliver J. Woodford, Carsten Rother, and Vladimir Kolmogorov. A global perspective on map

inference for low-level vision. In *International Conference on Computer Vision (ICCV)*, pages 2319–2326. IEEE, September 2009.

[72] Guoshen Yu, Guillermo Sapiro, and Stéphane Mallat. Solving inverse problems with piecewise linear estimators: From gaussian mixture models to structured sparsity. *Transactions on Image Processing*, 21(5):2481–2499, May 2012.

[73] Qidan Zhu and Lei Sun. An improved non-blind image deblurring methods based on foes. In *International Conference on Machine Vision (ICMV)*, volume 8783, pages 87830G–6. SPIE, March 2013.

[74] Song-Chun Zhu, Yingnian Wu, and David Mumford. Filters, random fields and maximum entropy (FRAME): Towards a unified theory for texture modeling. *International Journal of Computer Vision*, 27(2):107–126, 1998.