Journal of the Brazilian Computer Society, 2026, 32:1, doi: 10.5753/jbcs.2026.4907

A Simple U-Diffusion Inpainting Structure

Jugurta Montalvão [**Federal University of Sergipe** | ***jmontalvao@academico.ufs.br***]
Gabriel F.A. Bastos [**Federal University of Sergipe** | ***gabrielfab210102@gmail.com***]
Israel J. Santos Filho [**Université Paris-Saclay, Inria** | ***israel.jesus-santos-filho@inria.fr***]

Department of Electrical Engineering, Federal University of Sergipe, Av. Marechal Rondom, s/n, Jardim Rosa Elze, São Cristóvão, SE, 49.100-000, Brazil.

Received: 05 September 2024 • **Accepted:** 27 March 2026 • **Published:** 07 May 2026

Abstract The inpainting problem is addressed in this work through a very simplified version of the approach based on low-dimensional manifold model (LDMM), in which the actual working principle of the LDMM is put into evidence, namely, the simulated diffusion of image pixels that takes place in a manifold from which patches are drawn to form a given image. The simplicity of this principle is translated into a straightforward algorithm that borrows ideas from the Locally Linear Embedding (LLE) method, commonly used for dimensionality reduction and data visualization. The equivalence between this much simpler algorithm and the original LDMM is illustrated through visual inspection and experimental measurements of peak signal-to-noise ratios. By maintaining the key components of LDMM while reducing conceptual and computational complexity, the proposed method offers a streamlined solution for image inpainting tasks. Additionally, a (U-shaped) multi-scale use of the proposed algorithm is presented as a significantly better initializer for missing pixels, thus reducing the number of algorithmic iterations for convergence.

Keywords: LDMM, Multiscale, Inpainting, LLE

1 Introduction

The term *inpainting* refers to the task of automatically filling in missing or deteriorated samples from a signal or image without introducing noticeable changes from its undamaged version. Although this problem has been around for many years as a research hotspot, it has recently garnered a lot of attention due to its broad range of applications in various domains, such as object removal in information security , as shown by Zhang *et al*. [2018], audio and image restoration, as shown by Yao *et al*. [2022] and Adler *et al*. [2011], and denoising, as shown by Adam *et al*. [2017]. Moreover, despite usually being thought as an image processing matter, there are plenty of works that use the term inpainting to describe the restoration of distorted portions in other types of signals such as audio and seismograms, as shown by Adler *et al*. [2011] and Gadylshin *et al*. [2020].

Mathematically, the inpainting problem is described as the recovery of a signal $\mathbf{u} \in \mathbb{R}^n$ from incomplete measurements

$$\mathbf{x} = S(\mathbf{u}) + \eta, \tag{1}$$

where η is the observation noise and $S(\cdot)$ stands for a linear operator that causes subsampling in the original signal $\mathbf{u}$. Recovering $\mathbf{u}$ from the observed $\mathbf{x}$ is often an ill-posed problem, because even for $\eta = 0$, a given observed $\mathbf{x}$ may came from many possible signals $\mathbf{u}$, and some kind of regularizing *a priori* is needed. Indeed, a standard approach is to include a regularization term and solve the optimization problem

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \|\mathbf{x} - \mathbf{S}(\mathbf{u})\|_{\mathbf{2}}^{\mathbf{2}} + \lambda \mathcal{R}(\mathbf{u}), \tag{2}$$

where $\mathcal{R}$ is a function that represents some prior knowledge about the target signal (i.e. $\mathcal{R}(\mathbf{u})$ is known to be small for every possible signal of interest) and λ is a parameter that controls the compromise between data fidelity and prior knowledge. For natural images, a common choice for $\mathcal{R}$ is the total variation norm (see Getreuer [2012] for an example), which can eventually be replaced by nonlocal total variation in order to recover high frequency information (see Gilboa and Osher [2009] for an example). Another class of priors are the ℓ_1 regularizers. It was shown by Donoho [2006] that the ℓ_1 norm is a good approximation for the ℓ_0 norm, and can thus support many techniques that impose sparsity in a given transform domain or over a learned dictionary as prior for signal reconstruction, as in Vishwakarma [2023] and Tauböck *et al*. [2020]. Over the years, priors learned by deep convolutional neural networks in a data-driven manner have also been applied to achieve outstanding results, as shown by Burger *et al*. [2012], specially autoencoding models, such as the U-NET, a fully convolutional neural network originally developed by Ronneberger *et al*. [2015], that keeps yielding outstanding inpainting results in many application fields.

All of the aforementioned priors exploit one fundamental gear, namely, the inherent redundancy present in natural signals, which are known for having repetitive patterns and restrictions on their constructions. This redundancy may manifest in various ways, one of which is the generation of low dimensional structures, as stated by Peyré [2009]. Accordingly, patches from natural images and signals tend to lie along low dimensional manifolds, in spite of their apparent high-dimensionality.

Inspired by Peyré's findings, a method called low dimensional manifold model (LDMM) was proposed by Osher *et al*. [2017] for solving inverse problems, such as inpainting. In LDMM, the dimension of the manifold of patchs (chunks of

a given signal/image) is directly used as a regularizer. In Yin *et al*. [2017], this same dimensionality criterion was interpreted as an imposition of energy concentration of convolution framelet coefficients. A more systematic exploration of this energy concentration property yielded a new improved version of LDMM, called rw-LDMM.

In this paper, we propose a simpler alternative to the LDMM by revisiting the principles stated by Peyré [2009]. We show that almost equivalent results can be achieved by simple diffusion algorithms applied to the point cloud generated by signal patches, borrowing ideas from the Locally Linear Embedding (LLE) method, proposed by Roweis and Saul [2000]. The term diffusion, as used in this work, refers to the iterative computational transformation of variables to gradually satisfy local restrictions, as in numerical simulations of physical diffusions, which is a broad concept that also includes many image restoration and synthesis methods. Additionally, it is also shown that by using available redundancies at different scales of observations, in a standard coarse to fine multi-scale approach, the number of inpainting iterations can be significantly reduced. This multi-scale approach has been traditionally used in many imaging applications, such as texture synthesis and image deblurring, as it can be seen in Pan *et al*. [2014].

2 Related Work

In this section, we briefly discuss the key aspects of the references in which this work is grounded on, namely Peyré's work (Manifolds Models for Signals and Images), Low Dimensional Manifold Model and Locally Linear Embeddings.

2.1 Manifold Models for Signals and Images

Given a signal or an image $\mathbf{u}$, the assumption that the patches extracted from $\mathbf{u}$ lie on (or near to) a smooth manifold is present in several studies, such as in Lee *et al*. [2003] and Carlsson *et al*. [2008]. We refer to this manifold as the patch manifold, and denote it by $\mathcal{M}(\mathbf{u})$.

In Peyré [2009], models for patch manifolds of some suitable classes of signals and images were proposed, including smooth signals and images, cartoon images, locally stationary sounds, locally parallel textures and signals/images whose patches can be written as a sparse expansion over a learned dictionary. These models may not directly apply to natural images and signals, but rather to local structures that can be found in their compositions. Peyré's modeling consisted of explicitly parameterizing these manifolds. That is, for each patch manifold, he proposed a smooth function $\varphi : \mathbb{R}^m \to \mathcal{M}$ that uniquely reconstructs any point in $\mathcal{M}$ from a set of m parameters, where m is the dimension of the manifold.

Additionally, in that work, inpainting experiments were done using images from one of the aforementioned classes by iteratively solving the optimization problem described in Equation 2, with the regularizing function $\mathcal{R}(\mathbf{u})$ being the average distance between each patch in $\mathbf{u}$ and its projection onto a suitable manifold model.

2.2 Low Dimensional Manifold Model

Unfortunately, finding explicit parameterizations for patch manifolds of natural images is not feasible. However, one notable feature consistently observed across all the manifold models developed in Peyré [2009] is low-dimensionality. Inspired by that, Osher *et al*. [2017] took a step forward and proposed the use of this expected low dimensionality of the patch manifold as prior knowledge to regularize the inversion of the subsampling operator. This idea, called the Low Dimensional Manifold Model (LDMM), gives an optimization problem of the form

$$\hat{\mathbf{u}} = \arg\min_{\mathbf{u}} \|\mathbf{x} - \mathbf{S}(\mathbf{u})\|_{\mathbf{2}}^{\mathbf{2}} + \lambda \dim(\mathcal{M}(\mathbf{u})). \tag{3}$$

Using tools from differential geometry, the authors derive a formula to calculate the dimension of a smooth manifold $\mathcal{M} \subset \mathbb{R}^d$ given by

$$\dim(\mathcal{M}) = \sum_{i=1}^{d} |\nabla \alpha_i(\mathbf{p})|^2, \tag{4}$$

where α_i is the coordinate function, that is, $\alpha_i(\mathbf{p}) = p_i$ for every $\mathbf{p} = (p_1, p_2, \ldots, p_d) \in \mathcal{M}$ and each p_i is a coordinate element of $\mathbf{p}$.

With this expression, the optimization problem in Equation 3 is highly nonlinear and nonconvex. Therefore, an iterative approximate solution is required. In LDMM, the authors propose an alternating direction iterative procedure, where, in each iteration, first the manifold is fixed and the image and the coordinate functions are updated, then the manifold is updated with fixed images and coordinate functions. The first subproblem is formulated as a linear optimization problem with constraints, and the idea of split bregman iteration, from Gilboa and Osher [2009], is used to update $\mathbf{x}$ and α_i sequentially, while enforcing the constraints. Updating the image $\mathbf{x}$ requires solving a Laplace-Beltrami equation over an unstructured point set (the set of all image patches from the previous iteration). In LDMM, the point integral method (PIM), proposed by Li *et al*. [2017], is used to approximate this PDE by an integral equation, which is then discretized over the point cloud, allowing for an approximate solution.

2.3 Locally Linear Embedding

Locally Linear Embedding (LLE) is a manifold learning technique that was developed by Roweis and Saul [2000] in the context of dimensionality reduction and data visualization. From a collection of N points in a D-dimensional space, this method aims to project each point onto a space of lower dimensionality (in the context of visualization, this dimensionality has to be 2 or 3), while preserving the intrinsic structure of the underlying manifold from which these points are assumed to be (approximately) sampled.

This technique also relies on the assumption that the manifold on which the data points approximately lie has an intrinsic dimensionality d smaller than the ambient dimension D. Thus, around every point, this structure resembles a d-dimensional hyperplane. Following this reasoning, LLE seeks the optimal weights that reconstruct each D-dimensional data point as a linear combination of its K nearest neighbors (due

to the locally linear property, small reconstruction errors are expected when $K \geq d$, as long as the manifold is well sampled). As a second step, LLE seeks for a collection of N points in the euclidean space of lower dimensionality such that the same set of weights that reconstructs each point in the high dimensional space also reconstructs the projected version of this point in the low-dimensional space as a linear combination of the same neighbors.

2.4 Connections to our work

We now turn our attention to how the three aforementioned works relate to ours. First, we believe that the exploration of the low-dimensionality of the patch manifolds of natural images as prior regularization for inverse problems proposed by Osher *et al*. [2017] is remarkable and still makes room for improvements on inpainting applications – as well as other inverse problems. On the other hand, the exploration of this property in LDMM, which yields impressive inpainting results, relies on a highly sophisticated theoretical background, from the use of differential geometry tools to obtain an expression for the dimension of a smooth manifold, to the complexity of the steps and derivations required to solve the optimization problem described in Equation 3.

By contrast, our main motivation is to propose a simpler alternative to LDMM by developing an algorithm that explores the same fundamental gear – i.e. the expected low-dimensionality of the patch manifold of the target image – using three simple fundamental principles: (i) Natural images are expected to have local patterns that repeat themselves across the spatial domain. (ii) As a consequence of the previous observation, for any given patch, one may expect to find other similar patches in the same image, which in turn are expected to be near points in the patch manifold. (iii) As in LLE, one should be able to approximately reconstruct any patch as a linear combination of a reasonably small number of similar patches from the same image. Indeed, our approach is based on an iterative estimation of the unknown pixels such that the third principle is met.

In the next section, we illustrate the application of these principles to the inpainting problem through a simple naive diffusion procedure (under some arbitrary restrictions) on the patch manifold of the undersampled images, while in Section 4 we build upon the principles established in Section 3 to build an inpainting algorithm that performs a more refined form of diffusion.

3 Restricted diffusion

Low dimension of patch manifolds is a key aspect for the impressive results presented in Peyré [2009], Osher *et al*. [2017] and Yin *et al*. [2017], because it significantly increases the likelihood of finding two similar patches in a single signal instance, even if they are far apart in their independent variable domain (e.g. similar patches far apart in images, or similar sound segments far apart in time). Therefore, because similar patches are close points in their corresponding low-dimensional manifolds, for inpainting purposes, simulated diffusion through these manifolds can be used to replace missing values (e.g. missing pixels) with "diffused" near observations, as explained next through a toy example.

Toy example 01: For the purpose of this example, let $x(n)$ be a deterministic signal obtained through the logistic difference equation in Eq. 5,

$$x(n) = rx(n-1)(1-x(n-1)), \quad n = 1, 2, \ldots, \qquad (5)$$

with parameter $r = 3.7$ for the signal to be chaotic, and initial value $x(0) = 0.7$. Due to its chaotic nature, one should expect the inpainting of missing values in a single signal sample (without knowing the difference equation that generates it) to be a rather difficult task. Now, consider signal segments, or patches, $\mathbf{p}_n = [x(n-1), x(n), x(n+1)]$, corresponding to $x(n)$ and neighboring elements inside a lateral margin $L = 1$, as illustrated in Figure 1, for a signal with $N = 15$ sequential observations, where two similar patches can be seen, $\mathbf{p}_2$ and $\mathbf{p}_{13}$, although they are neither close to each other, in terms of the independent variable n, nor the result of a clear signal periodicity, for this signal is chaotic.

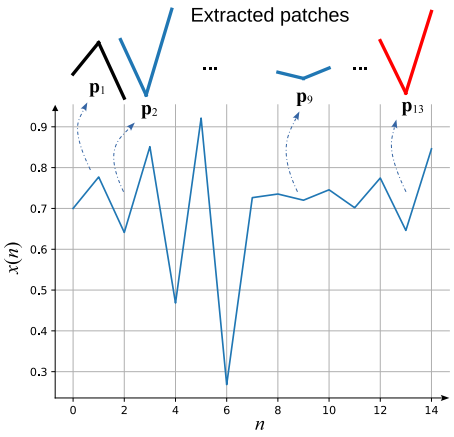

Figure 1. Short segment of $N = 15$ observations from the signal generated according to Eq. 5. Patches of length $L = 1$ are also illustrated, thus $\mathbf{p}_2$ and $\mathbf{p}_{13}$ can be visually compared.

Those patches are points in the $\mathbb{R}^3$, and because this short segment of signal yields only 13 patches, without any prior, the probability of randomly finding two close points (i.e. two similar patches) inside the 3D space of patches in only 13 trials is very low, for any reasonable closeness measure. But for this signal, all patches lie in an one-dimensional manifold embedded in the in $\mathbb{R}^3$, as seen in Figure 2.

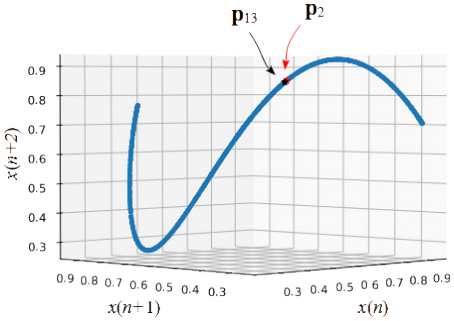

Figure 2. Low-dimensional manifold formed by patches from $x(n)$. Patches $\mathbf{p}_2$ and $\mathbf{p}_{13}$ are indicated as two close points in the low-dimensional manifold.

Indeed, $x(n)$ is obtained according to the restrictions imposed by Eq. 5, which yields the low-dimensional manifold represented in Fig. 2, and this simple example illustrates a broader perspective, whenever low-dimensional manifold models can be presumed, as the LDMM acronym suggests. Accordingly, low-dimensional manifolds significantly increase the probability of finding similar (redundant) patches even in small amounts of observations, such as in sound segments or single images where strong redundancies of any nature are found, for these redundancies allow for either local and non-local diffusions, both useful for signal inpainting, as explained next.

Local diffusion is well fitted for a specific kind of low-dimensional structure, associated with the predominance of low-frequency components of a target signal, which allows the reconstruction of degraded signals from near patches, because they are expected to be similar in slowly changing signals through time or space. By contrast, the non-local diffusion only requires low-dimensional manifolds of patches. In other words, it is assumed that any kind of dependency among signal samples – including, but not limited to low frequency range – is enough to allow signal reconstruction, because it allows the finding of similar patches in parsimonious amounts of signals, due to the limited volume of the manifold they actually occupy.

In the toy example, assume, for instance, that samples $x(2) = 0.64$, $x(9) = 0.72$ and $x(10) = 0.75$ are missing. Both local and non-local diffusion may be applied to estimate these values. First, every $x(n)$ is associated to either a flag $f_k(n) = 1$, if $x(n)$ is known, or to a flag $f_k(n) = 0$, otherwise. Additionally, for algorithmic purposes, another array of flags for estimated/known values is initialized as $f_e(n) = f_k(n),\ n = 0, 1, 2, \ldots, N$. Then we can use any measure of dissimilarity between patches, like, for instance, the supremum norm, as in Eq. 6.

$$d(\mathbf{p}_m, \mathbf{p}_n) = \begin{cases} \max\limits_{-L \le k \le L} |x(n+k) - x(m+k)| & \text{if } \delta_{m,n} \neq 0 \\ \infty & \text{otherwise.} \end{cases} \tag{6}$$

where $\delta_{m,n} = \sum\limits_{k=-L}^{L} f_e(n+k) f_e(m+k)$ is null for patches without at least two aligned elements that are known or already estimated. For instance, before any diffusion is done, we have $d(\mathbf{p}_2, \mathbf{p}_{13}) = 0.1$.

Distances between patches are then used to compute positive diffusion weights, as in Eq. 7,

$$w_{m,n} = \frac{e^{-\alpha d(\mathbf{p}_m, \mathbf{p}_n)}}{\sum\limits_{n \neq m} e^{-\alpha d(\mathbf{p}_m, \mathbf{p}_n)}} \tag{7}$$

where α is an arbitrary positive parameter in $\mathbb{R}$. Notice that the computation of weights through a softmax function assures that $\sum\limits_n w_{m,n} = 1$.

Finally, straightforward illustrations of inpainting through either local or non-local diffusions can be:

LD Local diffusion: Patches centered at missing values and their local neighbors are gathered. For instance, for the missing value $x(2)$, at the center of patch $\mathbf{p}_2$ the neighboring patches are $\mathbf{p}_1 = [0.70,\ 0.78,\ *]$ and $\mathbf{p}_3 = [\,*,\ 0.85,\ 0.47]$, with distances $d(\mathbf{p}_2, \mathbf{p}_1) = 0.08$ and $d(\mathbf{p}_2, \mathbf{p}_3) = 0.38$, respectively. Because distances are in the range of one tenth, α is arbitrarily set to 10, thus yielding $w_{2,1} =\approx 0.95$ and $w_{2,3} =\approx 0.5$, thus reflecting the perception that patch 1 is much closer to patch 2 than patch 3.
Now, $x(2)$ can be estimated as

$$\hat{x}(2) = w_{2,1} x(1) + w_{2,3} x(3) \approx 0.78$$

As compared to the actual $x(2) = 0.64$, this estimate is rather poor, but this diffusion step can be iterated throughout all elements of $x(n)$, many times, until the diffusion process stabilizes. For instance, Figure 3 shows the inpainting result for missing elements $x(2)$, $x(9)$ and $x(13)$ after 10 iterations of local diffusion.

NLD Non-local diffusion: Unlike the local diffusion, in which patches are assumed to be similar for neighboring values of n, in non-local diffusion, similar patches may occur anywhere, therefore some kind of near-neighbor search must be done. Thus, a single step of non-local diffusion can be defined as

$$\hat{x}(n) = \sum_{m \neq n} w_{n,m} x(m)$$

Figure 3 also shows the inpainting result for missing elements $x(2)$, $x(9)$ and $x(10)$ after 10 iterations of non-local diffusion.

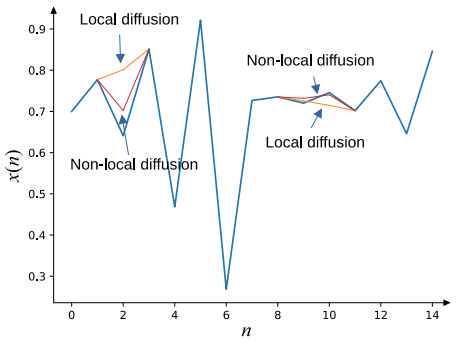

Figure 3. Inpainting of of $x(2)$, $x(9)$ and $x(10)$ through local and non-local diffusion, after 10 iterations, with $\alpha = 10$.

To further illustrate the effects of non-local diffusion, we borrowed from Peyré [2009] the coral image, with the same inpainted set of pixels as input image, and we restored it, as shown in Fig. 4, by taking every 5×5 patch centered at an unknown pixel, comparing it to all other 5×5 patches centered at a known or previously estimated pixel, and replacing the unknown pixel with the average of central pixels of the 3 near neighboring patches. Therefore, known pixels diffuses toward black spots, through patches of images that are neighbors in the underlying manifold.

Here, pixels are in the range from 0 (black) to 1 (white), but 10% of them are missing. Thus, to assure a proper finding of neighbors, we arbitrarily impose that a distance between two patches are to be computed only when at least 30% of known pixels in both patches have same row and column.

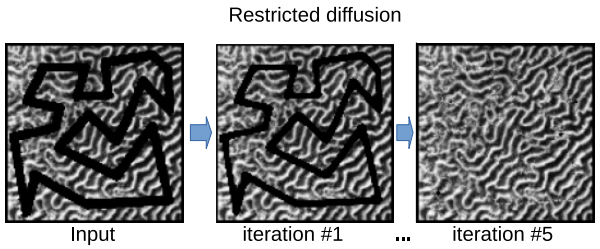

Figure 4. Illustration of the equivalence between the inpainting approach proposed by Peyré and a simple restricted diffusion. This inpainting result is to be compared to Fig. 16 in Peyré [2009].

Furthermore, only patches at supremum distances below 0.2 (in a range from 0 to 1) are taken as near neighbors. Because of these restrictions in distance computation between patches, we refer to it as a *restricted diffusion*, which can be iterated, so that missing values not estimated in the first interaction are gradually approached by estimated values, until all missing values are estimated, and this diffusion of known pixels through the manifold of patches can be stopped. An iteration of the restricted diffusion process described above can be summarized in the following steps:

(i) For every unknown pixel of the image, take the 5×5 patch centered around it. The goal is to estimate this target pixel.
(ii) Find the set of all 5×5 image patches centered around known pixel such that at least 30% of its known pixels lie in the same position of a known pixel of the patch centered around the target pixel. From this set, take as near neighbors of the patch centered around the target pixel the ones with supremum distance below 0.2
(iii) Estimate the target pixel as an average of the center pixel from all patches in this set of near neighbors. If this set is empty, the target pixel will continue to be considered unknown.

In every iteration, we go through all the unknown pixels. At first, the amount of known pixels is small, and the likelihood of the set described in step (ii) being empty is high. Therefore, a lot pixels may continue to be missing after the first iteration. However, as we go forward with more iterations, the amount of known pixels progressively increases, until the image is fully reconstructed and the diffusion process stops.

As illustrated in Fig. 4, this restricted diffusion does the most relevant work regarding missing values estimation. Indeed, if we consider the visual result presented in subfigure (c') of Figure 1 of Osher *et al*. [2017], obtained with the LDMM – which is itself built upon the principles established in Peyré [2009] –, this simple restricted diffusion alone, under restrictions similar to those applied in Fig. 4, yields a noisy but similar texture reconstruction, as shown in Fig. 5, including parallel details in the shawl, from 10% of known pixels, randomly taken.

Unfortunately, this straightforward form of restricted diffusion yields a noisy inpainting, because similar patches all over the image are prone to small value fluctuations. However, we reinforce that the goal of the experiments presented in this Section, using the procedures that we referred to as restricted diffusion, was by no means to introduce an algorithm able to yield the best high-quality reconstructions. Instead, we wanted to show how a simple diffusion algorithm over the

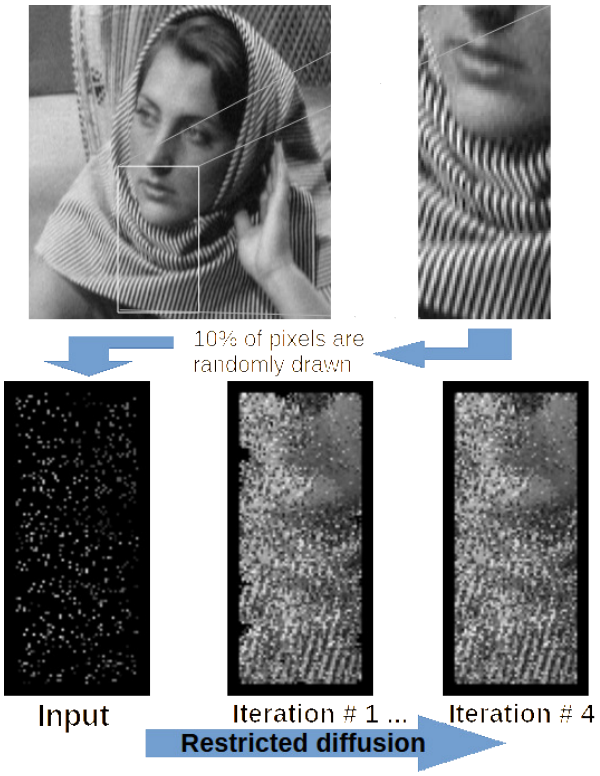

Figure 5. Illustration of how a restricted diffusion gathers non-local redundancies from known values, yielding image details reconstruction. This inpainting result is to be compared to Fig. 1 in Osher *et al*. [2017].

patches of the signal can do most of the relevant work in filling in the missing samples, allowing us to go from a highly undersampled piece of signal to some sort of recognizable reconstruction.

Therefore, we will not numerically measure the accuracy of the reconstructions yielded by these simple restricted diffusion procedures. Instead, their results and the insights gained from them motivate our search for an algorithm to obtain smoother versions of reconstructed images while taking advantage of the assumed locally low dimensionality of the patch manifold and imposing that near patches in the manifold are points lying on a hyperplane. To keep it as simple as possible, we use the same rationale behind the LLE method, as detailed in section 4.

4 Proposed algorithm

As the restricted diffusion used to obtain Figures 4 and 5, for every patch in the image whose center is an unknown value, a set of near neighboring patches are gathered as the first algorithmic step. Here again, computing distances between patches with missing values is an important concern. Unlike the previous restricted diffusion, however, a less constrained approach is taken, where it is assumed that all patches are fulfilled with either known or previously estimated values, then distances can be computed without restrictions at this algorithmic step, while restrictions are moved to the next steps, through a dumping factor Φ that controls the importance of the sampled and the missing pixels in the estimation of each target pixel. Thus, after the K_{NN} nearest neighbors of

the target patch are found[1], their distances are compared to a threshold Δ_{NN}, for a refined selection of near neighbor patches. Finally, following the same idea used in the LLE method, it is assumed that the target patch can be obtained as a weighted sum of its near neighbors (because the manifold is assumed to be locally almost flat). This assumption is expressed in Eq.8.

$$\hat{\mathbf{p}} = \mathbf{C}^T\mathbf{w}, \tag{8}$$

where $\hat{\mathbf{p}}$ is the estimated target patch represented as a vector ℓ-dimensional dimensional vector, $\mathbf{C}$ is a $K \times \ell$ matrix where each row is a selected near neighbor of the actual target, $\mathbf{p}$ (K is the number of near neighbors), and $\mathbf{w}$ is a K-dimensional vector of weights to be optimized. Thus, an error vector can be defined as $\mathbf{e} = \mathbf{p} - \hat{\mathbf{p}}$.

Because some elements of $\mathbf{e}$ result from unknown values of $\mathbf{p}$, the parameter Φ is finally used to reduce the importance of these error values, through the multiplication of $\mathbf{e}$ by a $d \times d$ diagonal matrix $\mathbf{D}$ whose entries are $\mathbf{D}(k,k) = |\mathbf{f}(k) - \Phi|$, where $\mathbf{f} \in \{0,1\}^d$ is a vector of flags indicating whether a pixel is known, with flag 1, or unknown, with flag 0. For instance, if Φ is set to 0.2, error values from known pixels in $\mathbf{p}$ become four times more relevant than those from unknown pixels.

Weight vector $\mathbf{w}$ is then optimized to minimize the norm of the (weighted) vector of errors through pseudo-inversion, as in Eq. 10. The minimum weighted error vector, in the least squares sense, is orthogonal to the column space of $\mathbf{C}^T$.That is,

$$\mathbf{C}D\mathbf{e} = \mathbf{C}D(\mathbf{p} - \hat{\mathbf{p}}) = \mathbf{C}D(\mathbf{p} - \mathbf{C}^T\mathbf{w}_O) = 0. \tag{9}$$

From that, we get

$$\mathbf{w}_O = (\mathbf{C}\mathbf{D}\mathbf{C}^T + \epsilon\mathbf{I})^{-1}\mathbf{C}\mathbf{D}\mathbf{p}. \tag{10}$$

where ϵ is a small positive real to prevent matrix inversion singularity ($\epsilon = 10^{-4}$ in all experiments reported in this paper), and $\mathbf{I}$ stands for the identity matrix.

The central pixel of the optimal reconstructed patch, $\hat{\mathbf{p}} = \mathbf{C}^T\mathbf{w}_O$, is averaged with the previous estimate, and the result is then used to re-estimate the central pixel of the target patch, as detailed in Algorithm 1.

Despite its simplicity, this algorithm yields experimental results almost equivalent to those proposed in Osher *et al*. [2017] and Yin *et al*. [2017], which can be illustrated with the same 10% of pixels from the Barbara image. In Fig. 6, the performance of Algorithm 1 is illustrated, following the initialization of unknown pixels with gaussian random noise, which is also the standard initialization used in LDMM. After 100 iterations of the algorithm, the obtained peak signal-to-noise ratio (PSNR) was 23.32 dB, whereas more elaborated algorithms in Osher *et al*. [2017] and Yin *et al*. [2017] yielded 24.74 dB and 25.41 dB, respectively, for the same task. But it is noteworthy that the visual quality of the presented result is equivalent to those presented in the mentioned works (see Figure 1, in Osher *et al*. [2017], and Figure 9, in Yin *et al*. [2017]), which is highlighted by zoomed spots in Fig. 6.

Besides, the simplicity of this algorithm allows for a parameter selection based on some further analyses. For instance, parameter Φ can be initially set according to the proportion of known values/pixels in the image. In the former illustration with Barbara image, Φ would be set around $1/11$ to compensate for the expected proportion of only 10% of known pixels in each patch (i.e. each known pixel would have 10 times the weight of an unknown one). Indeed, $\Phi = 1/11$ works fine in this case, but we fine tuned it to 0.2 in order to slightly improve the final PSNR.

[1]The metric used here is still the supremum distance, instead of the usual euclidean distance.

Algorithm 1: Restricted Diffusion Algorithm (single iteration)

Data: $\mathbf{X}_0$: Inpainted image, $\mathbf{F}_0$: Flags of known values/pixels, L: Padding size ($(2L+1) \times (2L+1)$ is the patch size), Δ_{NN}: Maximum supremum distance, K_{NN}: Maximum number of nearest neighbors, Φ: Weight of unknown values

$M_0 \leftarrow$ number of rows of $\mathbf{X}_0$;
$N_0 \leftarrow$ number of columns of $\mathbf{X}_0$;
Concatenate L zero-valued pixels at every border of $\mathbf{X}_0$ to obtain a zero-padded $(M_0 + 2L \times N_0 + 2L)$ matrix $\mathbf{X}$;
Concatenate L zero-valued pixels at every border of $\mathbf{F}_0$ to obtain a zero-padded $(M_0 + 2L \times N_0 + 2L)$ matrix $\mathbf{F}$;
$\mathbf{P} \leftarrow$ matrix with $M_0 \cdot N_0$ rows and $(2L+1)^2$ columns, where each row corresponds to a patch of $\mathbf{X}$;
$\mathbf{P_f} \leftarrow$ matrix with $M_0 \cdot N_0$ rows and $(2L+1)^2$ columns, where each row corresponds to a patch of $\mathbf{F}$;
$center \leftarrow \frac{(2L+1)^2+1}{2}$;
foreach *m and n where $\mathbf{F}_0(m,n) = 0$ (unknown value/pixel)* **do**
  $i \leftarrow m + L$
  $j \leftarrow n + L$
  $r \leftarrow M_0(n-1) + m$, row index of the vectorized patch centered around $\mathbf{X}(i,j)$.
  $\mathbf{p} \leftarrow r$-th row of $\mathbf{P}$, which contains the vectorized patch centered around $\mathbf{X}(i,j)$.
  $\mathbf{f} \leftarrow r$-th row of $\mathbf{P_f}$, which contains the vectorized patch centered around $\mathbf{F}(i,j)$.
  Find the K_{NN} near neighbors of $\mathbf{p}$ among the rows of $\mathbf{P}$;
  $r_k \leftarrow$ row index of the k-th collected near neighbor ($k \in \{1,2,\ldots,K\}, K \leq K_{NN}$) whose supremum distance to $\mathbf{p}$ is $\leq \Delta_{NN}$. Here, K is the number of near neighbors collected. Thus, $K = 0$ if all distances are higher than Δ_{NN}).
  if $K > 0$ **then**
    $\mathbf{D} \leftarrow$ diagonal matrix whose entries are $\mathbf{D}(k,k) = |\mathbf{f}(k) - \Phi|$;
    $\mathbf{C} \leftarrow$ matrix whose K rows are taken from $\mathbf{P}$, at indexes $r_1, r_2, \ldots, r_K$;
    $\hat{\mathbf{p}} \leftarrow \mathbf{C}^T(\mathbf{C}\mathbf{D}\mathbf{C}^T + \epsilon\mathbf{I})^{-1}\mathbf{C}\mathbf{D}\mathbf{p}$
    $\mathbf{X}_0(i,j) \leftarrow \frac{1}{2}(\mathbf{X}_0(i,j) + \hat{\mathbf{p}}(center))$
  end
end

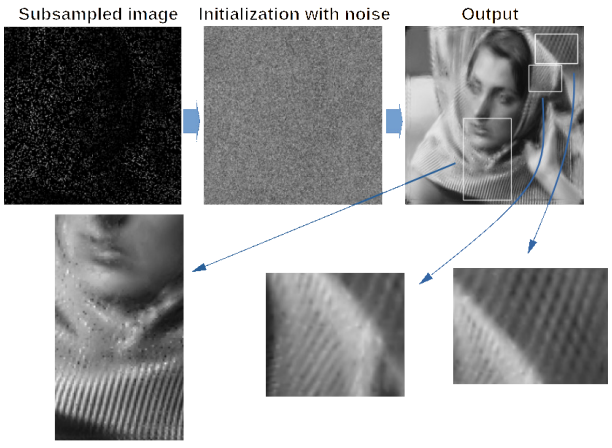

Figure 6. Performance illustration for the proposed inpainting algorithm, where 90% of pixels are missing (left). These missing pixels are initialized with noise (center) and given as input to the algorithm, whose output (right) has a PSNR of 23.32 dB. Reconstruction details are zoomed in for better visual comparisons.

To set parameter Δ_{NN}, one may use the intrinsic dimensionality analysis, as explained in Montalvão *et al*. [2020], based on the Grassberger-Procaccia algorithm. Accordingly,

a multiscale analysis can be done, as shown in Fig. 7, with a subsample of patches randomly drawn from the image. For this example, 5000 patches (17×17) were randomly drawn from the image, and two relevant estimates are shown in Fig. 7. These two relevant estimates suggest that patches gathered in hypercubes of edge around $\Delta = 2^{-2.5}$ tend to be on a flat surface whose local dimension is between 2 and 3. Indeed, a local dimension around 2D is expected due to the superposition of patches, as explained by Peyré [2009]. The second estimate, in a larger scale of analysis, around $\Delta = 2^{-1}$, seems to be a more interesting one, for inpainting purposes, for it suggests that patches gathered in hypercubes of edge around $\Delta = 1/2$ tend to be on a flat 5-D surface, which seems to be the consequence of image texture redundancies, beyond the 2 degrees of freedom due to patch superposition. Therefore, we may initially set Δ_{NN} to 0.5 to a reasonably inpainting. From this initial value, we fine-tuned it to 0.7, for a better result in terms of PSNR. The flip side of this approach is that, because the perfect image is known only in controlled experiments, but not in real applications of image recovery from their sparse representations, the Grassberger-Procaccia should be applied to similar images, when available. Otherwise, Δ_{NN} must be arbitrarily set in the range from 0 to 1.

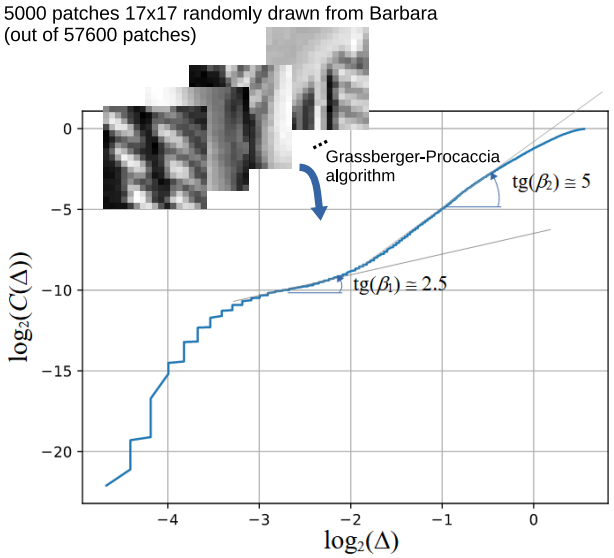

Figure 7. Grassberger-Procaccia analysis of intrinsic local dimension. Δ stands for the supremum distance between patches whereas $C(\Delta)$ is the counting of pairwise patch coincidences (or collisions) inside a hypercube of edge Δ. Two main trends: around $\Delta = 2^{-2.5}$ and $\Delta = 2^{-1}$, for local intrinsic dimension around 2.5 and 5, respectively.

4.1 U-shaped restricted diffusion

Initialization of unknown pixels with gaussian noise, with the same mean and variance of the known pixels, is a usual step for preparing input images in LDMM and rw-LDMM, but a standard coarse-to-fine multiscale approach can replace it with advantages, where both the initial subsampled image and matrix of flags – level 0 – are reduced through a max-pooling downsampling until the number of unknown pixels is null – level B. Then the inpainting algorithm is applied to the image at level $B - 1$. After stabilization of the diffusion process at this level, the restored image is upsampled, through interpolation using a standard transposed convolution with a 2×2 kernel of ones and stride 2, to level $B - 2$, where the diffusion process takes place again, and this is repeated until level 0 is attained, as illustrated in Fig. 8. In this illustration, $B - 1 = 3$, where only 3 pixels are to be inpainted. Unlike usual multiscale inpainting approaches based on convolutional neural networks, such as the ones from Li *et al.* [2023] and Ye *et al.* [2018], neither a learning database, nor a training step is required here.

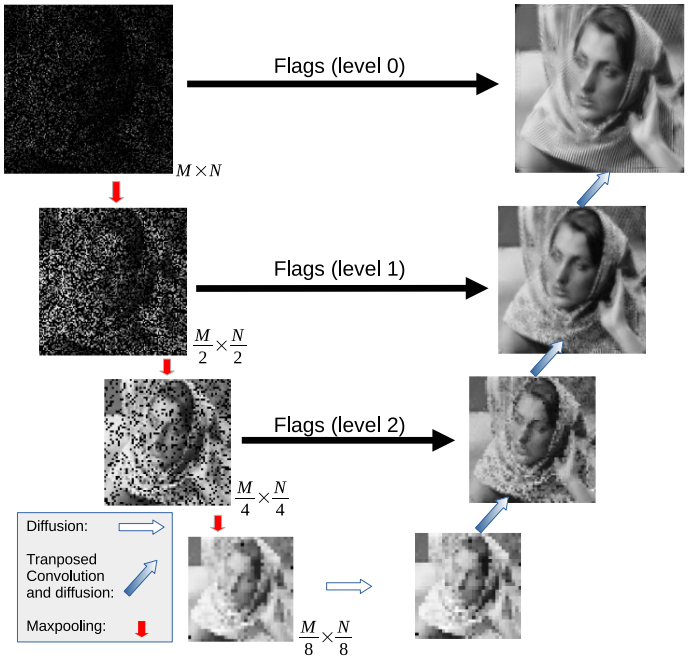

Figure 8. Illustration of the U-shaped structure of multiscale diffusion.

Fig. 10 provides a perspective on the rapid convergence of the proposed U-shaped approach toward its final performance.

A relevant observation regarding our reconstruction can be made using the Barbara image. As noted by Osher *et al.* [2017], the background texture behind the figure resembles the stitching patterns of a chair. We draw attention to this effect to mitigate any potential misinterpretation that our method creates artifacts or introduces new frequency patterns that are non-existent in the reference image.

Indeed, a key property of our approach is that it does not create or "hallucinate" information where sufficient statistical support is lacking, which is partly a consequence of the simplified strategy we propose. As a side effect, our proposed approach tends to struggle to inpaint image patches lying on poorly sampled regions of the patch manifold. This characteristic can be verified with one of the test images, "checkerboard". The algorithm is unable to reconstruct some portions of this image, even from 50% of the pixels, due to the absence of self-similar regions, indicating that an insufficient number of similar patches are available to restore certain areas, as shown in Figure 9.

4.2 Code Availability

An implementation of the proposed algorithm (including the multiscale approach) along with scripts to reproduce the experiments presented in the next section, is

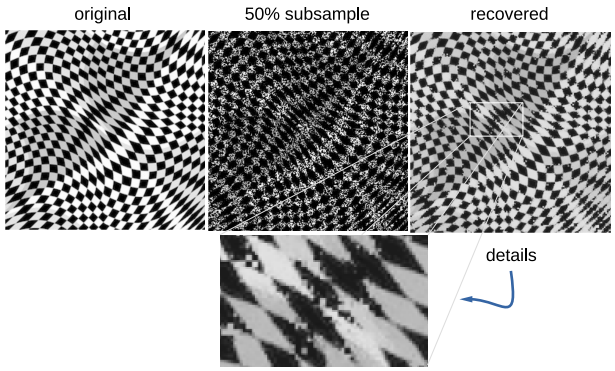

Figure 9. Reconstruction of the checkerboard image from 50% of its pixels, which reveals a potential limitations of the proposed algorithm

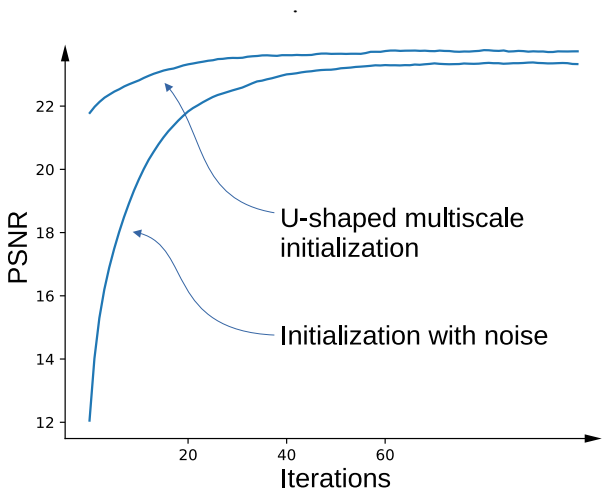

Figure 10. Comparison between diffusion stabilization through iterations. The U-shaped approach provides a much better initialization at level 0 than initialization with noise. It also provides a slightly better final PSNR.

available at: `https://github.com/gabrielfab0022/A-Simple-U-Diffusion-Inpainting-Structure`.

5 Experiments

5.1 Inpainting Experiments

Inpainting experiments were done with images also used in Osher *et al*. [2017] and Yin *et al*. [2017] with 10% and 50% of the pixels, to allow direct comparisons in terms of PSNR, as in Table 1 and Table 2, respectively.

Table 1. Inpainting PSNR (dB) from 10% of the pixels.

Image	LDMM	rw-LDMM	U-Diffusion
Barbara	24.74–24.78	25.34–25.59	24.91
Boat	23.08–23.21	23.31–23.66	23.35
Couple	23.04–23.78	23.62–24.27	23.94
Checkerboard	12.18–12.37	13.74–14.29	12.29
Fingerprint	21.79-21.93	22.20–22.60	22.96
Man	23.84-24.21	23.87–24.62	24.26
Hill	28.26–28.71	28.37–29.07	29.23
House	28.19–29.30	28.83–29.93	28.31
Swirl	19.02–19.31	20.83–21.60	19.08

The term U-Diffusion refers to the proposed algorithm, including the U-shaped chain of multiscale diffusions. Results for the LDMM and rw-LDMM were taken from the references, whereas results from the proposed U-Diffusion were

Table 2. Inpainting PSNR (dB) from 50% of the pixels.

Image	LDMM	rw-LDMM	U-Diffusion
Barbara	28.62–36.10	27.01–36.20	33.47
Boat	28.59–32.78	28.22–32.77	30.87
Couple	28.41–32.29	28.51–32.32	30.87
Checkerboard	27.41–28.11	26.80–27.46	15.78
Fingerprint	27.10-27.84	26.90–27.94	31.91
Man	29.61-32.30	29.51–32.34	30.69
Hill	32.95–35.06	32.78–35.18	35.15
House	35.96–38.69	34.57–38.72	35.46
Swirl	32.75–35.39	32.71–35.39	29.15

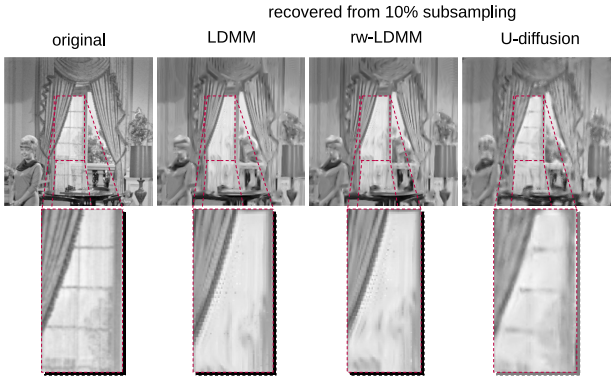

Figure 11. Direct comparison of detail reconstructions between U-Diffusion and competing methods in the image “Couple”.

not optimized. Those results suggest that PSNR values from all methods are in the same range, while the recovered images are qualitatively alike, as illustrated in Figures 12 and 13, where we show the inpainting results of our method in four test images, and their corresponding LDMM and rw-LDMM reconstructions, from 10% and 50% of the pixels, respectively. By carefully taking a look at these Figures, one may observe that the overall quality of the reconstructed images are, indeed, similar, even though some specific parts of the images may be better reconstructed by a given method. For instance, if we take a look at the test image “Couple”, we can see that the U-Diffusion presents a better reconstruction of the details from the windows, as highlighted in Figure 11.

Besides, by looking at Table 2, the disparity of the results from the checkerboard test image may stand out. With 10% of the pixels, the final PSNR of LDMM and of the proposed approach are on the same range for this image, with rw-LDMM presenting a slightly superior performance. However, with the increase of the proportion of known pixels to 50%, the final PSNR of LDMM and rw-LDMM shows an increase of about 15 dB, while the proposed U-diffusion only improve its performance in about 3 dB. As discussed in Section 4.1, this reveals a potential limitation of the proposed approach, namely the inability to recover image patches represented by points in regions of the patch manifold that are poorly sampled (in the statistical sense). We conjecture that this limitation comes from the choice of filling in the missing pixels through linear interpolation using the near patches, which may not be enough to satisfactorily reconstruct patches with too few near neighbors on the same image.

To finish this section, we point out that thoroughly comparing the performance of the proposed algorithm to those of deep learning models is beyond the scope of the present paper. However, as an illustration, it is worthy mentioning the work

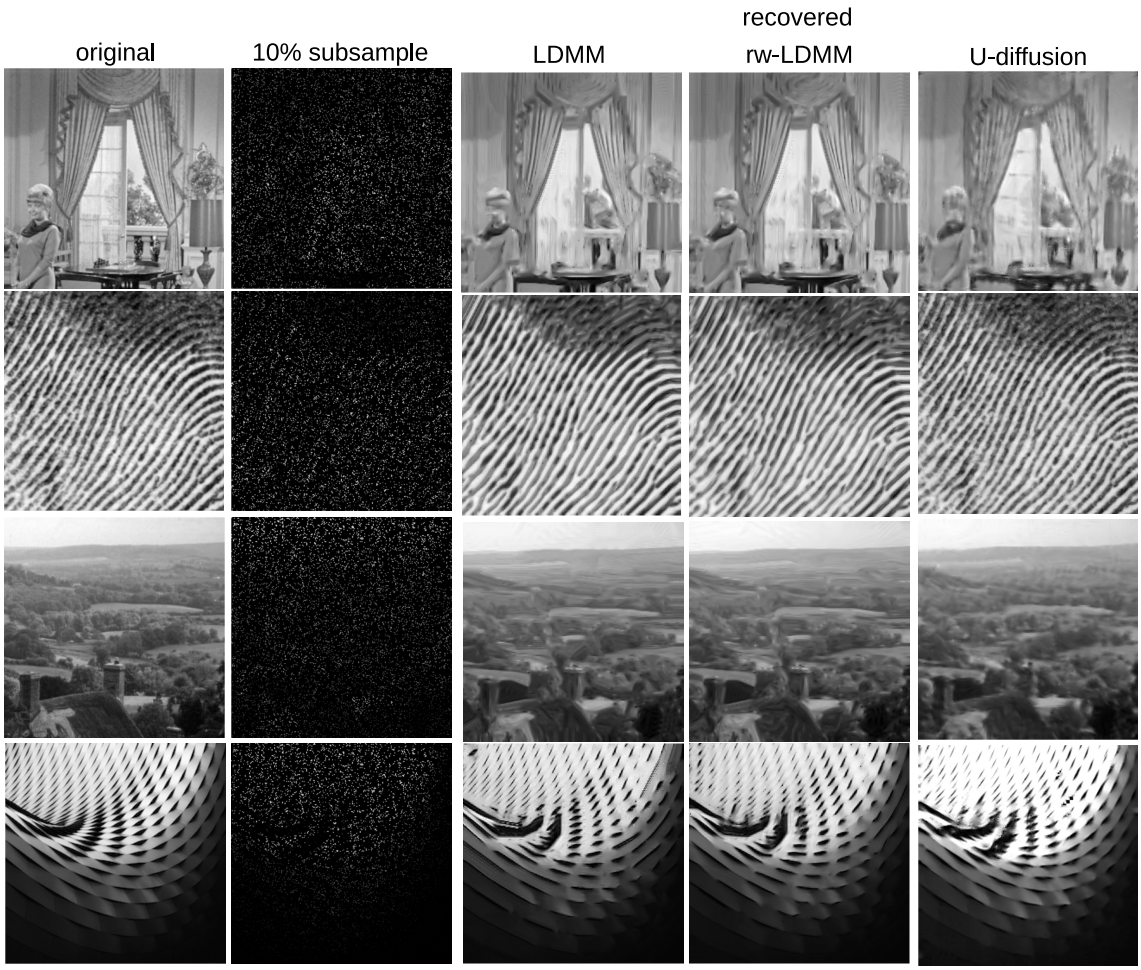

Figure 12. Inpainting results from 10% of the pixels from the following four images, from top to bottom: Couple, Fingerprint, Hill, Swirl.

by Ulyanov *et al*. [2018], who leveraged a U-Net architecture, which is the backbone of many state of the art image restoration models, to build deep image priors for image inpainting. In that work, the authors also report experiments using random 50% subsampling masks applied to some of the images that we also used here. Namely, they tested their model on the images "Barbara", "Boat", "House", "Couple", "Fingerprint", "Hill" and "Man", yielding PSNRs of 32.22 dB, 33.06 dB, 39.16 dB, 32.52 dB, 32.84 dB and 32.20 dB, respectively, which are also at similar ranges compared to ours. It is important to say, though, that despite being a deep learning-based approach, their reconstruction is not data driven, as it doesn't require a training dataset. Instead, the authors argue that the inherent structural prior of a U-Net architecture suffices to ensure that the network will synthesize a solution to this ill-posed problem that satisfies the constraints of a natural image, even when the processing is done on a single damaged image. This reinforces the idea that multi-scale representations are highly effective for this task.

5.2 Parameter sensitivity analysis

We now turn our attention to the evaluation of how varying the parameters of the algorithm alters the final result. For this analysis, we consider the Barbara image in both sampling scenarios – 10% and 50%.

To start with, the parameters are fixed at $L = 8, Knn = 20, \Phi = 0.2$ and $\Delta_{NN} = 0.7$ for the experiments with 10% of the pixels, and at $L = 2, Knn = 20, \Phi = 0.2$ and $\Delta_{NN} = 0.5$ for the experiments with 50% percent of the pixels. Then, we vary a single parameter while keeping the others fixed. When varying the parameters $L, Knn, \Phi, \Delta_{NN}$, they are chosen from the sets $\{2, 4, 6, 8, 10\}$, $\{10, 15, 20, 30, 40\}$, $\{0.1, 0.15, 0.2, 0.25, 0.3\}$ and $\{0.3, 0.4, 0.5, 0.6, 0.7\}$, respectively. The choice of these sets is because in the experimental results reported in subsection 5.1, for all the images, the parameters were always in these ranges.

The results of the final PSNR after 100 iterations at level 0 in every parameter setting tested are shown in Figures 14 and 15 for experiments with 10% and 50% of the pixels, respectively.

By taking a look at these graphics, one may observe that the parameters Knn and L are the ones with the most influence on the final result. For instance, when varying the values of Knn in the experiments with 50% of the pixels, the best final PSNR was obtained at $Knn = 15$, whereas a much inferior result was obtained when $Knn = 40$. On the other hand, the algorithm seems to be more stable when the parameters Δ_{NN} and Φ are varied.

5.3 Computational complexity

Besides its conceptual simplicity, another important advantage of the proposed algorithm is its reduced computational

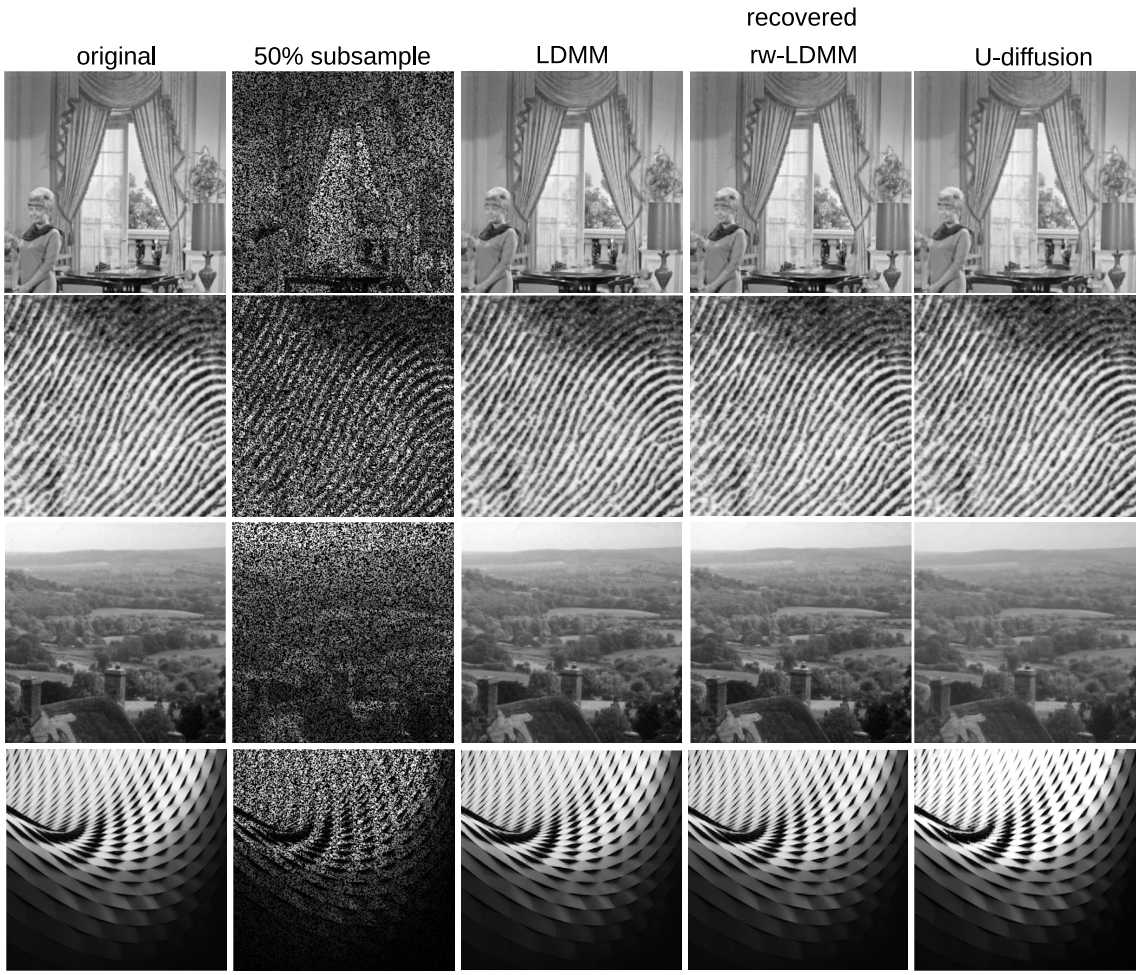

Figure 13. Inpainting results from 50% of the pixels from the following four images, from top to bottom: Couple, Fingerprint, Hill, Swirl.

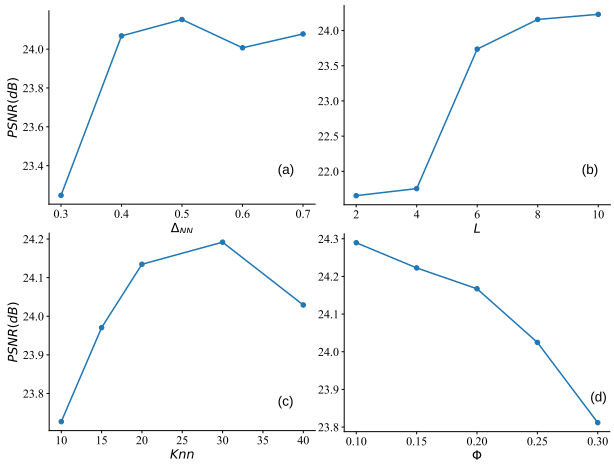

Figure 14. Final PSNR after 100 iterations for experiments with 10% of the pixels. In these experiments, all parameters are kept fixed except (a) Δ_{NN}, (b) L, (c) Knn, and (d) Φ.

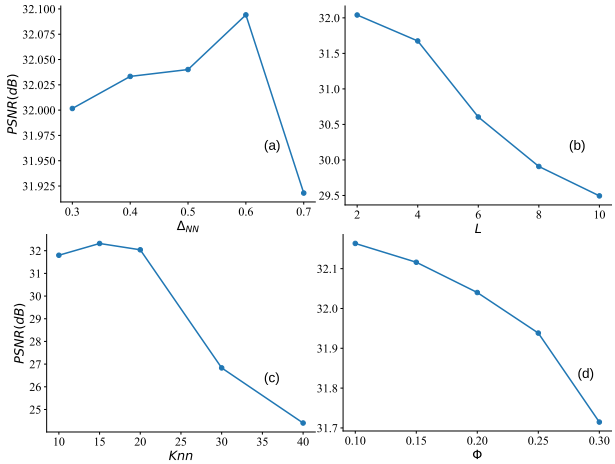

Figure 15. Final PSNR after 100 iterations for experiments with 50% of the pixels. In these experiments, all parameters are kept fixed except (a) Δ_{NN}, (b) L, (c) Knn, and (d) Φ.

cost. To inpaint an $N \times N$ image, the proposed algorithm requires at most N^2 pseudo-inversions of $\ell \times K_{NN}$ matrices at every iteration (for the number of near neighbors can be less than K_{NN}), where ℓ is the patch size (i.e. $\ell = (2L+1)^2$), which has a computational complexity of $\mathcal{O}(N^2 \ell K_{NN}^2)$, in addition to nearest neighbors searches, which are common for all algorithms. In contrast, every iteration of the LDMM requires solving $\ell\, N^2 \times N^2$ linear systems. Using GMRES, proposed by Saad and Schultz [1986], the time complexity of it is $\mathcal{O}(\ell k nnz)$, where k is the number GMRES iterations and nnz is the number of nonzero entries in the weighted adjacency matrix, which is usually $50N^2$, since only the 50 nearest neighbors are considered to make the matrix sparse. Considering the typical values of the parameters mentioned in both papers, and a reasonable amount of GMRES iterations($k = 50$), we estimate that, without considering nearest neighbor queries, an iteration of our method can be about 2 to 3 times faster than a LDMM one. As for rw-LDMM, the cost is even higher,

since $\ell + r$ systems are solved, and calculating the partial singular value decomposition of a $N^2 \times \ell$ matrix is also required.

6 Conclusion

A new inpainting algorithm was proposed, which is built upon the seminal ideas found in Peyré [2009], but also guided by a search for simplicity. We gathered experimental evidences that the very simple principle of diffusion through a manifold of patches, under some restrictions imposed by missing values, is the main gear behind the inpainting results obtained with the LDMM and the rw-LDMM, both presented in a convoluted articulation of abstract concepts. In this work, it was shown that a much simpler perspective (and implementation) is enough to yield similar results, in terms of PSNR, with equally similar visual results.

Besides, a U-shaped structure for the algorithm was proposed, in which a multiscale handling of subsampled images yields a much better starting point for algorithmic iterations at the highest resolution scale, many dB of PSNR above usual initialization with noise. This initial advantage can be used to significantly reduce computational burden. Moreover, by comparing the computational burden of each iteration, the proposed algorithm shows to be less time consuming, as compared to both the LDMM and the rw-LDMM.

Given the myriad of potential applications of a method such as the LDMM, which is able to solve other ill-posed problems, besides inpainting – e.g. denoising, super resolution –, we believe that the simplicity of the method proposed in this work, including its multiscale U-shaped structure, may be a useful tool for researchers and practitioners with many different theoretical backgrounds, for it can be easily implemented and modified for further experimentation and development of new related ideas.

Declarations

Acknowledgements

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil.

Authors' Contributions

JM contributed to the conception of the study, including investigation, data analysis, experiment coding and discussion of the results, and is the lead author of this research. GFAB contributed with writing, conceptual discussion, algorithm implementation and design of experiments. IJSF contributed with writing/reviewing, conceptual discussions, algorithm implementation and design of experiments. All authors read and approved the final version of the manuscript.

Competing interests

The authors declare that they have no competing interests.

Availability of data and materials

An implementation of the proposed algorithm and scripts to reproduce experiments are available at: `https://github.com/gabrielfab0022/A-Simple-U-Diffusion-Inpainting-Structure`.

References

Adam, R. D., Peter, P., and Weickert, J. (2017). Denoising by inpainting. In *Scale Space and Variational Methods in Computer Vision: 6th International Conference, SSVM 2017, Kolding, Denmark, June 4-8, 2017, Proceedings 6*, pages 121–132. Springer. DOI: 10.1007/978-3-319-58771-4_10.

Adler, A., Emiya, V., Jafari, M. G., Elad, M., Gribonval, R., and Plumbley, M. D. (2011). Audio inpainting. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(3):922–932. DOI: 10.1109/TASL.2011.2168211.

Burger, H. C., Schuler, C. J., and Harmeling, S. (2012). Image denoising: Can plain neural networks compete with bm3d? In *2012 IEEE conference on computer vision and pattern recognition*, pages 2392–2399. IEEE. DOI: 10.1109/CVPR.2012.6247952.

Carlsson, G., Ishkhanov, T., De Silva, V., and Zomorodian, A. (2008). On the local behavior of spaces of natural images. *International journal of computer vision*, 76(1):1–12. DOI: 10.1007/s11263-007-0056-x.

Donoho, D. L. (2006). For most large underdetermined systems of equations, the minimal l1-norm near-solution approximates the sparsest near-solution. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 59(7):907–934. DOI: 10.1002/cpa.20132.

Gadylshin, K., Silvestrov, I., and Bakulin, A. (2020). Inpainting of local wavefront attributes using artificial intelligence for enhancement of massive 3-d pre-stack seismic data. *Geophysical Journal International*, 223(3):1888–1898. DOI: 10.1190/segam2019-3214642.1.

Getreuer, P. (2012). Total variation inpainting using split bregman. *Image Processing On Line*, 2:147–157. DOI: 10.5201/IPOL.2012.G-TVI.

Gilboa, G. and Osher, S. (2009). Nonlocal operators with applications to image processing. *Multiscale Modeling & Simulation*, 7(3):1005–1028. DOI: 10.1137/0706985.

Lee, A. B., Pedersen, K. S., and Mumford, D. (2003). The nonlinear statistics of high-contrast patches in natural images. *International Journal of Computer Vision*, 54(1):83–103. DOI: 10.1023/a:1023705401078.

Li, J., Huang, C., Chan, R., Feng, H., Ng, M. K., and Zeng, T. (2023). Spherical image inpainting with frame transformation and data-driven prior deep networks. *SIAM Journal on Imaging Sciences*, 16(3):1177–1194. DOI: 10.1137/22M152462X.

Li, Z., Shi, Z., and Sun, J. (2017). Point integral method for solving poisson-type equations on manifolds from point clouds with convergence guarantees. *Communications in Computational Physics*, 22(1):228–258. DOI: 10.4208/cicp.111015.250716a.

Montalvão, J., Canuto, J., and Miranda, L. (2020). Bias-compensated estimator for intrinsic dimension and differential entropy. *Journal of Communication and Information Systems*, 35(1):300–310. DOI: 10.14209/jcis.2020.30.

Osher, S., Shi, Z., and Zhu, W. (2017). Low dimensional manifold model for image processing. *SIAM Journal on Imaging Sciences*, 10(4):1669–1690. DOI: 10.1137/16M1058686.

Pan, J., Hu, Z., Su, Z., and Yang, M.-H. (2014). Deblurring text images via l0-regularized intensity and gradient prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2901–2908. DOI: 10.1109/cvpr.2014.371.

Peyré, G. (2009). Manifold models for signals and images. *Computer vision and image understanding*, 113(2):249–260. DOI: 10.1016/j.cviu.2008.09.003.

Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical image computing and computer-assisted intervention–MICCAI 2015: 18th international conference, Munich, Germany, October 5-9, 2015, proceedings, part III 18*, pages 234–241. Springer. DOI: 10.1007/978-3-319-24574-4_28.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding. *science*, 290(5500):2323–2326. DOI: 10.1126/science.290.5500.2323.

Saad, Y. and Schultz, M. H. (1986). Gmres: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal on scientific and statistical computing*, 7(3):856–869. DOI: 10.1137/0907058.

Tauböck, G., Rajbamshi, S., and Balazs, P. (2020). Dictionary learning for sparse audio inpainting. *IEEE Journal of Selected Topics in Signal Processing*, 15(1):104–119. DOI: 10.1109/JSTSP.2020.3046422.

Ulyanov, D., Vedaldi, A., and Lempitsky, V. (2018). Deep image prior. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9446–9454. DOI: 10.1109/cvpr.2018.00984.

Vishwakarma, A. (2023). Denoising and inpainting of sonar images using convolutional sparse representation. *IEEE Transactions on Instrumentation and Measurement*, 72:1–9. DOI: 10.1109/TIM.2023.3246527.

Yao, D., McLaughlin, S., and Altmann, Y. (2022). Patch-based image restoration using expectation propagation. *SIAM Journal on Imaging Sciences*, 15(1):192–227. DOI: 10.1137/21M1427541.

Ye, J. C., Han, Y., and Cha, E. (2018). Deep convolutional framelets: A general deep learning framework for inverse problems. *SIAM Journal on Imaging Sciences*, 11(2):991–1048. DOI: 10.1137/17M1141771.

Yin, R., Gao, T., Lu, Y. M., and Daubechies, I. (2017). A tale of two bases: Local-nonlocal regularization on image patches with convolution framelets. *SIAM Journal on Imaging Sciences*, 10(2):711–750. DOI: 0.1137/16M1091447.

Zhang, D., Liang, Z., Yang, G., Li, Q., Li, L., and Sun, X. (2018). A robust forgery detection algorithm for object removal by exemplar-based image inpainting. *Multimedia Tools and Applications*, 77:11823–11842. DOI: 10.1007/s11042-017-4829-0.