

 Open access • Journal Article • DOI:10.1007/S11263-010-0418-7

## **A Variational Framework for Exemplar-Based Image Inpainting** — [Source link](#)

Pablo Arias, Gabriele Facciolo, Vicent Caselles, Guillermo Sapiro

**Institutions:** Pompeu Fabra University, University of Minnesota

**Published on:** 01 Jul 2011 - International Journal of Computer Vision (Springer US)

**Topics:** Inpainting, Real image and Image processing

Related papers:

- [Region filling and object removal by exemplar-based image inpainting](#)
- [Texture synthesis by non-parametric sampling](#)
- [Image inpainting](#)
- [Space-Time Completion of Video](#)
- [PatchMatch: a randomized correspondence algorithm for structural image editing](#)

Share this paper:    

View more about this paper here: <https://typeset.io/papers/a-variational-framework-for-exemplar-based-image-inpainting-49k6nnclrv>

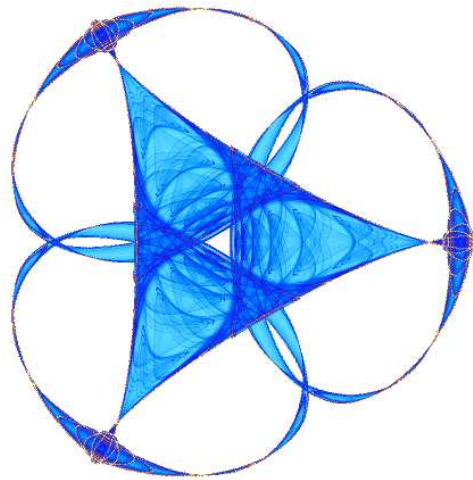
**A VARIATIONAL FRAMEWORK FOR  
EXEMPLAR-BASED IMAGE INPAINTING**

By

**Pablo Arias**  
**Gabriele Facciolo**  
**Vicent Caselles**  
and  
**Guillermo Sapiro**

**IMA Preprint Series # 2306**

(April 2010)



**INSTITUTE FOR MATHEMATICS AND ITS APPLICATIONS**

UNIVERSITY OF MINNESOTA  
400 Lind Hall  
207 Church Street S.E.  
Minneapolis, Minnesota 55455-0436

Phone: 612-624-6066 Fax: 612-626-7370

URL: <http://www.ima.umn.edu>

---

# A Variational Framework for Exemplar-Based Image Inpainting

Pablo Arias · Gabriele Facciolo · Vicent Caselles · Guillermo Sapiro

**Abstract** Non-local methods for image denoising and inpainting have gained considerable attention in recent years. This is in part due to their superior performance in textured images, a known weakness of purely local methods. Local methods on the other hand have demonstrated to be very appropriate for the recovering of geometric structures such as image edges. The synthesis of both types of methods is a trend in current research. Variational analysis in particular is an appropriate tool for a unified treatment of local and non-local methods. In this work we propose a general variational framework non-local image inpainting, from which important and representative previous inpainting schemes can be derived, in addition to leading to novel ones. We explicitly study some of these, relating them to previous work and showing results on synthetic and real images.

**Keywords** Inpainting · Variational methods · Self-similarity · Non-local methods · Exemplar-based methods

## 1 Introduction

*Image inpainting*, also known as *image completion* or *disocclusion*, is an active research area in the image processing field. The purpose of inpainting is to obtain a visually plausible image interpolation in a region in which data are missing due to damage or occlusion.

---

P. Arias, G. Facciolo, V. Caselles  
Dept. of Information and Communication Technologies, Universitat Pompeu Fabra  
Barcelona, Spain  
E-mail: {pablo.arias,gabriele.facciolo,vicent.caselles}@upf.edu

G. Sapiro  
Dept. of Electrical and Computer Engineering, University of Minnesota  
Minneapolis, USA E-mail: guille@umn.edu

Usually, to solve this problem, the only available data is the image outside the region to be inpainted. In addition to its theoretical interest, image inpainting has applications to image and video editing and restoration.

Most inpainting methods found in the literature can be classified into two groups: *geometry-* and *texture-oriented* methods. We now briefly review the developments in both types of approaches, with emphasis in texture-oriented methods. This review will help for motivating the proposed formulation.

*Geometry-oriented methods.* Images are modeled as functions with some degree of smoothness, expressed for instance in terms of the curvature of the level lines or the total variation of the image. The interpolation is performed by continuing and imposing this model inside the inpainting domain, usually by means of a partial differential equation (PDE). Such PDE can be derived from variational principles, as for instance in [6, 19, 20, 28, 44, 45], or inspired by physical processes [8, 11, 54]. These methods show good performance in propagating smooth level lines or gradients, but fail in the presence of texture. They are often referred to as *structure* or *cartoon* inpainting.

Geometry-oriented methods are *local* in the sense that the associated PDEs only involve interactions between neighboring pixels on the image grid. An implication of this is that among all the data available in the image, these methods only use that around the boundary of the inpainting domain.

*Texture-oriented methods.* Texture-oriented inpainting was born as an application of texture synthesis, e.g., [26, 35]. Its recent development was triggered in part by the works of Efros and Leung [26] and Wei and Levoy [55] using non-parametric sampling techniques (parametric models have also been considered, e.g. [40]). In these

works texture is modeled as a two dimensional probabilistic *graphical model*, in which the value of each pixel is conditioned by its neighborhood. These approaches rely directly on a sample of the desired texture to perform the synthesis.

In practice these methods work progressively by expanding a region of synthesized texture. The value of a target pixel  $x$  is copied from the center of a (square) patch in the sample image, chosen among those that best match the available portion of the patch centered at  $x$ . Levina and Bickel [41] provided a probabilistic theoretical justification for this strategy.

This method (as explained above or with some modifications) has been extensively used for inpainting [9, 10, 22, 25, 26, 48]. As opposed to geometry-oriented inpainting, these so-called *exemplar-based* approaches, are *non-local*: To determine the value at  $x$ , the whole image may be scanned in the search for a matching patch.

Since these texture approaches are *greedy* procedures (each hole pixel is visited only once), the results are very sensitive to the order in which pixels are processed [22]. This issue was addressed in [39, 56] where the inpainting problem is stated as a probabilistic inference problem in a graphical model. In both cases, the image is modeled using a pair-wise Markov Random Field (MRF). In [39] an energy considering the overlap error of adjacent patches is minimized using *belief propagation*. The method in [56] can be seen as an approximated Expectation-Maximization (EM) method. A similar scheme was further explored in [38], however with a different variational justification, closely related to the models addressed in this work.

The algorithms of [38, 56] require an initial completion which, as it turns out, has a great impact on the final result. In both cases the authors resort to a *multiscale approach*: The image is filtered and subsampled, possibly several times. Starting from the coarsest, the inpainting algorithm is then applied sequentially to each scale, using the upsampled result of the previous scale as initialization. This so-called multiscale approach yields very good results, and has been also applied successfully to other exemplar-based methods [30].

Another variational justification for texture-based methods was presented in [24], where the inpainting problem is reformulated as that of finding a *correspondence map*  $\Gamma : O \rightarrow O^c$ ,  $O$  being the inpainting domain and  $O^c$  its complement w.r.t. the image domain. Denoting the image by  $u$ , the inpainted value at position  $x \in O$  is then given by  $u(x) = u(\Gamma(x))$ , being  $\Gamma(\cdot)$  the correspondence map. The authors proposed a continuous energy functional in which the unknown is the

correspondence map itself:

$$E(\Gamma) = \int_O \int_{\Omega_p} (u(\Gamma(x-y)) - u(\Gamma(x)-y))^2 dy dx,$$

where  $\Omega_p$  is the patch domain (centered at  $(0,0)$ ). Thus  $\Gamma$  should map a pixel  $x$  and its neighbors in such a way that the resulting patch is close to the one centered at  $\Gamma(x)$ . This model has been the subject of further analysis by Aujol *et al.* [4], where extensions are proposed and the existence of a solution in the set of piecewise roto-translation maps  $\Gamma$  is proved. The authors also discuss the use of different regularizers which include a regularization of the curvature of the level lines (or surfaces) of the image. These approaches are theoretical and no numerical optimization scheme is available so far.

A different variational model was presented in [49]. Images are modeled as ensembles of patches on a given patch manifold. For inpainting, the patch manifold can be learned from the set of patches in the hole's complement. The method is iterative, with each iteration having two steps. First, the patches in the hole are projected onto the manifold. Since this is done for each patch independently, the projected patches are not necessarily coherent with each other, *i.e.* overlapping patches may have different values in the overlap region. Therefore, in the second step, an image is computed by averaging the patches in the ensemble.

Another front of activity is given by the techniques based on the *sparseland* model [1, 16], in which the image is restricted to have a sparse representation over an overcomplete basis or *dictionary* [1, 27, 43]. The main difference between dictionary-based and exemplar-based methods lies in where the missing information is obtained from. Dictionary based methods look for the missing data in the dictionary (as a linear combination of a few atoms), whereas exemplar-based methods assume that the information needed lies elsewhere in the image itself (or in a database of images [33]). These methods perform well in problems with small or scattered inpainting domains, but fail with a large holes.

Exemplar-based methods provide impressive results in recovering textures and repetitive structures. However, their ability to recreate the geometry without any example is limited and not well understood. Therefore, different strategies have been proposed which combine geometry and texture inpainting [9, 17, 25, 37]. These methods usually decompose the image in some sort of structure and texture components. Structure is reconstructed using some geometry-oriented scheme, and this is used to guide the texture inpainting.

*Contributions.* Despite these combined methods, geometry and texture inpainting are still quite separate fields,

each one with its own analysis and implementation tools. Variational models as the one introduced in this paper can provide common tools allowing a unified treatment of both trends. We therefore propose a variational framework for non-local image inpainting as a contribution to the modeling and analysis of texture-oriented methods.

Like the *non-local means* denoising algorithm [5, 14] we encode the image redundancy and self-similarity (measured as patch similarity) as a non-local weight function  $w : O \times O^c \rightarrow \mathbb{R}$ . This function serves as a fuzzy correspondence, and differs from the works [4, 24], although a (eventually multivalued) correspondence map can be approximated as a limit of our model. The proposed formulation is rather general and different inpainting schemes can be derived naturally from it, via the selection of the appropriate patch similarity criterion. In this work we present four of them, *patch NL-means*, *-medians*, *-Poisson* and *-gradient medians*, corresponding to similarity criterions based on  $L^2$ - and  $L^1$ -norms between patches or their gradients. We provide an comprehensive empirical comparison on real and synthetic problems, showing the benefits and limitations of each model.

The *patch NL-means* is related to the method of [38, 56] and can be interpreted in terms of the *mean shift* [21] and the manifold models of [49]. The other schemes are, to the best of our knowledge, novel. Both gradient-based methods, *patch NL-Poisson* and *-gradient medians* combine the exemplar-based interpolation with local PDE-based diffusion schemes. This results in a smoother continuation of the information across the boundary and inside the inpainting domain, and in a better propagation of structures.

Finally, we also present an initial discussion along the line of providing a formal modelling of the multi-scale scheme. We argue that this approach is not just some heuristic to find a good minimum of the single scale inpainting functional, but constitutes an inpainting criterion itself.

Although the focus of this work lies on inpainting, the framework we are introducing can be adapted for its application in other contexts. In particular it is related to graph-based non-local regularizers. Inspired by graph regularization techniques [57], these approaches model the image as a graph characterized by the similarity weights [32, 42]. Recently, our formalism [3] was extended by Peyré *et al.* in [51] to provide a variational justification to the case in which the graph is adaptive. On the other hand the patch NL-means scheme provides a model for an iterative version of the non-local means algorithm with adaptive similarity weights. Similar approaches have been applied to denoising [5, 52],

super-resolution [53], texture denoising [12], and demosaicing [15] among others. We will discuss the relation with these models in the text.

The rest of this document is organized as follows. In Section 2 we introduce the proposed variational framework, together with the derivation and the empirical comparison of the different inpainting schemes. The links with related work are discussed in Section 3. In Section 4 we present and discuss the multiscale approach. And in Section 5 we present experimental results on real images allowing to compare our results with the state of the art. Concluding remarks and future work is discussed in Section 6.

We note that a preliminary version of this work has been presented in [3].

*Notation.* Images are denoted as functions  $u : \Omega \rightarrow \mathbb{R}$ , where  $\Omega$  denotes the image domain, usually a rectangle in  $\mathbb{R}^2$ . Pixel positions are denoted by  $x$ ,  $\hat{x}$ ,  $z$ ,  $\hat{z}$  or  $y$ , the latter for positions inside the patch. A patch of  $u$  centered at  $x$  is denoted by  $p_u(x) = p_u(x, \cdot) : \Omega_p \rightarrow \mathbb{R}$ , where  $\Omega_p$  is a rectangle centered at  $(0, 0)$ . The patch is defined by  $p_u(x, y) = u(x + y)$ , with  $y \in \Omega_p$ .  $O \subset \Omega$  is the hole or inpainting domain, and  $O^c = \Omega \setminus O$ . We still denote by  $u$  the part of the image  $u$  inside the hole, while  $\hat{u}$  is the part of  $u$  in  $O^c$ :  $\hat{u} = u|_{O^c}$ . Additional notation will be introduced in the text.

## 2 Variational framework

Our variational framework is inspired by the following non-local functional

$$F_w(u) = \int_O \int_{O^c} w(x, \hat{x})(u(x) - \hat{u}(\hat{x}))^2 d\hat{x} dx \quad (1)$$

where  $w : O \times O^c \rightarrow \mathbb{R}^+$  is a weight function that measures the similarity between patches centered in the inpainting domain and in its complement.

Let us assume for the moment that the weights are known. The minimum of (1) should have a low pixel error  $(u(x) - \hat{u}(\hat{x}))^2$  whenever the similarity  $w(x, \hat{x})$  is high. In this way the similarity weights drive the information transfer from known to unknown pixels. A similar functional was proposed in [31] as a non-local regularization energy in the context of image denoising. It models the non-local means filter [5, 14] when the weights are *Gaussian*

$$w(x, \hat{x}) \propto \exp\left(-\frac{1}{h} \|p_u(x) - p_{\hat{u}}(\hat{x})\|^2\right),$$

where  $\|\cdot\|$  is a weighted  $L^2$ -norm in the space of patches and  $h$  is a parameter that determines the selectivity of the weights  $w$ .

In [31] the weights are considered known and remain fixed through all the iterations. While this might be appropriate in applications where they can be estimated from the noisy image, in the image inpainting scenario here addressed, the weights are not available and have to be inferred together with the image (as in [50,53]). One of the novelties of the proposed framework is the inclusion of adaptive weights in a variational setting.

For this reason, we will consider the weight function  $w$  as an additional unknown. Instead of prescribing explicitly the Gaussian functional dependence of  $w$  w.r.t.  $u$ , we will do it implicitly, as a component of the optimization process. In doing so, we obtain a simpler functional, avoiding to deal with the complex, non-linear dependence between  $w$  and  $u$ . In our formulation,  $w(x, \cdot)$  is a probability density function,

$$\int_{O^c} w(x, \hat{x}) d\hat{x} = 1,$$

and can be seen as a relaxation of the one-to-one correspondence map of [4,24], providing a fuzzy correspondence between each  $x \in O$  and the complement of the inpainting domain.

In this setting, we propose an energy which contains two terms, one of them is inspired by (1) and measures the coherence between the pixels in  $O$  and those in  $O^c$ , for a given similarity weight function  $w$ . This permits the estimation of the image  $u$  from the weights  $w$ . The second term allows us to compute the weights given the image. The complete proposed functional is

$$E(u, w) = \frac{1}{h} \tilde{F}_w(u) - \int_{\tilde{O}} H_w(x) dx, \quad (2)$$

where

$$\tilde{F}_w(u) = \int_{\tilde{O}} \int_{\tilde{O}^c} w(x, \hat{x}) \varepsilon(p_u(x) - p_{\hat{u}}(\hat{x})) d\hat{x} dx, \quad (3)$$

$\varepsilon(\cdot)$  is an error function for image patches, and

$$H_w(x) = - \int_{\tilde{O}^c} w(x, \hat{x}) \log w(x, \hat{x}) d\hat{x},$$

is the entropy of the probability  $w(x, \cdot)$ .

We take  $\tilde{O}$ , the *extended inpainting domain*, as the set of centers of patches that intersect the hole, *i.e.*  $\tilde{O} = O + \Omega_p = \{x \in \Omega : (x + \Omega_p) \cap O \neq \emptyset\}$ . Thus, patches  $p_{\hat{u}}(\hat{x})$  centered in  $\hat{x} \in \tilde{O}^c$  are entirely outside  $O$  (Figure 1). Accordingly, we consider that the weight function  $w$  is defined over  $\tilde{O} \times \tilde{O}^c$  and  $\int_{\tilde{O}^c} w(x, \hat{x}) d\hat{x} = 1$ . For a simplified presentation, we assume that  $\tilde{O} + \Omega_p \subseteq \Omega$ , *i.e.* every pixel in  $\tilde{O}$  supports a patch centered on it and contained in  $\Omega$ . Analogously, we also shrink  $\tilde{O}^c$  to have  $\tilde{O}^c + \Omega_p \subseteq \Omega$ .

Let us now make some additional comments on the functional. We observe that the term  $(u(x) - \hat{u}(\hat{x}))^2$  in (1), that penalizes differences between pixels, is substituted in (3) by the patch error function  $\varepsilon(p_u(x) - p_{\hat{u}}(\hat{x}))$ . This has two consequences. First, minimizing (3) with respect to the image will force patches  $p_u(x)$  to be similar to  $p_{\hat{u}}(\hat{x})$  for each pair  $x, \hat{x}$  such that  $w(x, \hat{x})$  is large. The other implication has to be understood together with the inclusion of the second term, which integrates the entropy of each probability  $w(x, \cdot)$  over  $\tilde{O}$ . For a given completion  $u$ , and for each  $x \in \tilde{O}$ , the optimum weights minimize the mean patch error for  $p_u(x)$ , given by

$$\int_{\tilde{O}^c} w(x, \hat{x}) \varepsilon(p_u(x) - p_{\hat{u}}(\hat{x})) d\hat{x},$$

while maximizing the entropy. For instance taking  $\varepsilon$  as the squared  $L^2$ -norm of the patch, then the resulting weights are Gaussian. This can be related to the *principle of maximum entropy* [36], widely used for inference of probability distributions. The parameter  $h$  controls the trade-off between both terms and is also the selectivity parameter of the Gaussian weights. The trivial minima of  $E$  with  $w(x, \hat{x}) = 0$  everywhere is discarded by restricting  $w(x, \cdot)$  to be a probability.

*The patch error function.* Patches are functions defined on  $\Omega_p$ , and the error function  $\varepsilon : \mathbb{R}^{\Omega_p} \rightarrow \mathbb{R}^+$  is defined as the weighted sum of pixel-wise errors  $e : \mathbb{R} \rightarrow \mathbb{R}^+$

$$\varepsilon(p_u(x) - p_{\hat{u}}(\hat{x})) := \int_{\Omega_p} g_a(y) e(u(x+y) - \hat{u}(\hat{x}+y)) dy, \quad (4)$$

where the *intra-patch* weight function  $g_a$  is a Gaussian centered at the origin with standard deviation  $a$ . We will consider the  $L^1$ - and the squared  $L^2$ -norms as particular cases of  $\varepsilon(\cdot)$ , with  $e(\cdot) = |\cdot|$  and  $e(\cdot) = |\cdot|^2$  respectively.

We will also consider patch error functions involving the gradient of the image. In an abuse of notation we will denote the gradient's patch and pixel-wise error functions as  $\varepsilon : \mathbb{R}^{2\Omega_p} \rightarrow \mathbb{R}^+$  and  $e : \mathbb{R}^2 \rightarrow \mathbb{R}^+$ , respectively. It will be clear from the argument which case is intended, as in this example  $\varepsilon(p_{\nabla u}(x) - p_{\nabla \hat{u}}(\hat{x})) := \int_{\Omega_p} g_a(y) e(\nabla u(x+y) - \nabla \hat{u}(\hat{x}+y)) dy$ .

Additionally, we can impose an additive penalization of the distance  $\varphi(x - \hat{x})$  as in [38] by modifying the patch distance function  $\varepsilon_d(p_u(x) - p_{\hat{u}}(\hat{x}), x, \hat{x}) = \varepsilon(p_u(x) - p_{\hat{u}}(\hat{x})) + \varphi(x - \hat{x})$  to penalize the use of distant patches. We will not consider this modification in the current work.

As it will be discussed below, the patch error function determines not only the similarity criterion but



also the image synthesis, and thus is a key element in the proposed framework.

## 2.1 Minimization of $E$

We have formulated the inpainting problem as the constrained optimization

$$(u^*, w^*) = \arg \min_{u, w} E(u, w) \quad (5)$$

$$\text{subject to } \int_{\tilde{O}^c} w(x, \hat{x}) d\hat{x} = 1 \quad \forall x \in \tilde{O},$$

where  $E$  is the inpainting energy defined in (2). For energies involving image gradients we will consider Dirichlet conditions at the boundary between the inpainting domain and the image, and Neumann conditions at the boundary of the image, *i.e.*  $u(x) = \hat{u}(x)$  in  $\partial O \setminus \partial \Omega$  and  $\nabla u(x) \cdot n(x) = 0$  for  $x \in \partial O \cap \partial \Omega$ .

To minimize the energy  $E$ , we use an alternate minimization algorithm. At each iteration, two optimization steps are solved: The constrained minimization of  $E$  with respect to  $w$  while keeping  $u$  fixed; and the minimization of  $E$  with respect to  $u$  with  $w$  fixed. This procedure yields the following iterative scheme:

---

**Algorithm 1** Alternate minimization of  $E(u, w)$ .

---

**Require:** *Initial Condition*  $u_0(x)$  with  $x \in O$ .

1: **repeat**

2: *Update Weights:*  $w_k = \arg \min_w E(u_k, w)$ ,  
s.t.  $\int_{\tilde{O}^c} w(x, \hat{x}) d\hat{x} = 1$ .

3: *Update Image:*  $u_{k+1} = \arg \min_u E(u, w_k)$ .

4: **until** *Stop Criterion:*  $\|u_{k+1} - u_k\| < \text{tolerance}$ .

---

In the weights update step, the minimization of  $E$  w.r.t.  $w$  yields:

$$w_k(x, \hat{x}) = \frac{1}{q(x)} \exp\left(-\frac{1}{h} \varepsilon(p_{u_k}(x) - p_{\hat{u}}(\hat{x}))\right), \quad (6)$$

where  $q(x) = \int_{\tilde{O}^c} \exp\left(-\frac{1}{h} \varepsilon(p_{u_k}(x) - p_{\hat{u}}(\hat{x}))\right)$  is a normalization factor that makes  $w(x, \cdot)$  a probability. For the functionals defined with gradient patches, the weight update equation is analogous to (6) replacing the patch error function with  $\varepsilon(p_{\nabla u_k}(x) - p_{\nabla \hat{u}}(\hat{x}))$ .

The parameter  $h$  determines the selectivity of the similarity. If  $h$  is large, maximizing the entropy becomes more relevant, yielding weights which are less selective. In the limit, when  $h \rightarrow \infty$ ,  $w(x, \cdot)$  becomes a uniform distribution over  $\tilde{O}^c$ . On the other hand, a small  $h$  yields weights which concentrate on the patches similar to  $p_u(x)$ . When  $h \rightarrow 0$ , we compute the weights

as  $\lim_{h \rightarrow 0} w(x, \hat{x}) = \frac{1}{|n(x)|} \delta(\hat{x} - n(x))$ , where  $n(x) \subseteq \tilde{O}^c$  is the set of nearest neighbors of  $x$ , defined as

$$n(x) = \arg \min_{\hat{x} \in \tilde{O}^c} \varepsilon(p_u(x) - p_{\hat{u}}(\hat{x})).$$

That is  $w(x, \cdot)$  can be considered as an approximation to a multivalued correspondence. For simplicity, in practice we assume that  $|n(x)| = 1$ , *i.e.* the nearest neighbor is unique.

The image update step deserves more attention and is described next.

### 2.1.1 Image update step.

In this section we present the derivation of the image update step corresponding to the four patch error functions mentioned earlier. First we will present the cases when image patches are compared using the squared  $L^2$ -norm and the  $L^1$ -norm. We refer to the resulting algorithms as *patch-wise non-local means* (patch NL-means), and *medians* (patch NL-medians). Then we consider functionals involving the gradients of the patches both with the squared  $L^2$ -norm and the  $L^1$ -norm. These methods will be referred as *patch-wise non-local Poisson* (patch NL-Poisson), and *gradient medians* (patch NL-GM).

Before moving to the derivation of these schemes, let us remark that with the change of variables  $z = x + y$ ,  $\hat{z} = x + y'$ , the image energy term can be expressed as an accumulation of pixel-wise errors:

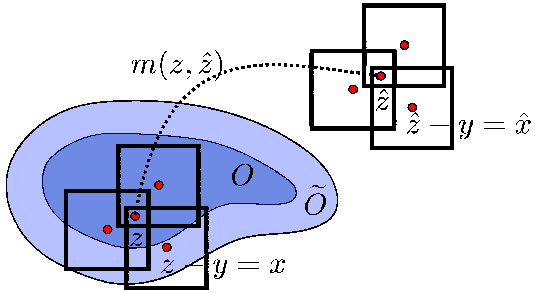
$$\begin{aligned} \tilde{F}_w(u) &= \int_{\tilde{O}} \int_{\tilde{O}^c} w(x, \hat{x}) \\ &\quad \int_{\Omega_p} g_a(y) e(u(x+y) - \hat{u}(\hat{x}+y)) dy d\hat{x} dx \\ &= \int_O \int_{O^c} m(z, \hat{z}) e(u(z) - \hat{u}(\hat{z})) d\hat{z} dz + C, \end{aligned} \quad (7)$$

where  $C$  is a constant term. We have introduced the *pixel-wise influence weights*  $m : O \times O^c \rightarrow \mathbb{R}^+$  defined as

$$m(z, \hat{z}) := \int_{\Omega_p} \chi_{\tilde{O}^c}(\hat{z} - y) g_a(y) w(z - y, \hat{z} - y) dy. \quad (8)$$

The function  $\chi_{\tilde{O}^c}$  takes the value 1 on  $\tilde{O}^c$  and 0 on  $\tilde{O}$ . An analogous expression can be computed for gradient patch error functions. This rewriting simplifies the following derivations and provides some insights on the implications of using patch-wise errors.

For each pair of pixels  $(z, \hat{z}) \in O \times O^c$ ,  $m(z, \hat{z})$  weights the effective contribution of the pixel-wise error between  $u(z)$  and  $\hat{u}(\hat{z})$  in the total value of the energy. The quantity  $m(z, \hat{z})$  is computed by integrating the similarity  $w(z-y, \hat{z}-y)$  between all patches that overlap



**Fig. 1 Patch-wise non-local inpainting.** The value at  $z \in O$  is computed using contributions from all the patches that overlap it, these are patches centered at  $x \in \tilde{O}$  such that  $z = x + y$  with  $y \in \Omega_p$ . The influence function  $m(z, \hat{z})$  accumulates all the contributions  $w(z - y, \hat{z} - y)$  from patches centered at  $\hat{z} - y$  to  $z - y$ .

$\hat{z}$  and those that overlap  $z$  in the *same relative position* (shown in Figure 1). It tells us how much evidence there is supporting a correspondence between the locations  $z$  and  $\hat{z}$ . Also observe that for each  $z$ ,  $\int_{O^c} m(z, \hat{z}) d\hat{z} = 1$ , then  $m(z, \cdot)$  can also be interpreted as a probability density function. Note that the energy (7) corresponds to (1), with the patch similarity weights  $w$  being substituted with the pixel-wise influence weights  $m$ , a sort of spatial convolution of  $w$  with kernel  $g_a$ .

*Patch non-local means.* If we use a weighted squared  $L^2$ -norm as a patch error function  $\varepsilon(p_u(x) - p_{\hat{u}}(\hat{x})) := \int_{\Omega_p} g_a(y) |u(x + y) - \hat{u}(\hat{x} + y)|^2 dy$  in (3) then the image energy term (7) is quadratic on  $u$ , and its minimum for fixed weights  $w$  can be computed explicitly as a non-local average:

$$u(z) = \frac{1}{c(z)} \int_{O^c} m(z, \hat{z}) \hat{u}(\hat{z}) d\hat{z}, \quad (9)$$

for  $z \in O$ , where the normalization constant  $c(z) := \int_{O^c} m(z, \hat{z}) d\hat{z}$ . Although  $c(z) = 1$  with the current definition of  $w$  and  $g_a$ , we will keep a generic notation in the following derivations.

The formal similarity with the non-local means equation hides some important differences, which are a direct consequence of the use of a patch error function in the image energy term. To obtain more insight about this let us expand  $m$  to obtain:

$$u(z) = \frac{1}{c(z)} \int_{\Omega_p} g_a(y) \int_{O^c} w(z - y, \hat{x}) \hat{u}(\hat{x} + y) d\hat{x} dy.$$

There are two averaging processes involved in the synthesis. The outer integral goes through all patches  $p_u(z - y)$  overlapping the target pixel  $z$ . Each patch *suggests* a value for  $z$  resulting from the inner sum: A non-local average of the pixel at position  $y$  in all patches  $p_{\hat{u}}(\hat{x})$  in  $\tilde{O}^c$ . This sum is weighted by the similarity between the patch  $p_u(z - y)$  and each  $p_{\hat{u}}(\hat{x})$ .

Therefore, we can distinguish two types of pixel interactions. Interactions due to the patch overlap of nearby pixels in the image lattice and non-local interactions driven by the similarity weights. The latter can be controlled by the selectivity parameter  $h$ , but the extent of the overlap interactions is given by the patch size. In particular when  $h \rightarrow 0$  Eq. (9) yields

$$u(z) = \int_{\Omega_p} g_a(y) \hat{u}(n(z - y) + y) dy$$

(recall we are assuming a unique nearest neighbor). For each  $x \in \tilde{O}$ , the nearest neighbor of  $p_u(x)$  is centered at  $x$  and all these patches are averaged according to  $g_a$ . This blending may cause some blur, which leads us to consider the  $L^1$ -norm in the search of a more robust image synthesis.

In Appendix A we prove the existence of minima for the Patch NL-Means energy in the continuous setting.

*Patch non-local medians.* The  $L^1$ -norm patch error function in the image energy term corresponds to taking  $e(x) = |x|$  in (7). The Euler equation for  $u$ , given the influence function  $m$ , can be formally written as

$$[\delta_u \tilde{F}_w(u)](z) = \int_{O^c} \text{sign}[u(z) - \hat{u}(\hat{z})] m(z, \hat{z}) d\hat{z} \ni 0.$$

This expression is multivalued, since  $\text{sign}(r) = r/|r|$  if  $|r| > 0$  and  $\text{sign}(r) \in [-1, 1]$  if  $r = 0$ . Its solution for each  $u(z)$ ,  $z \in O$  is obtained as a weighted median of the pixels of the complement  $O^c$ , with weights  $m(z, \cdot)$ .

Both schemes presented so far perform inpainting by transferring (by averages or medians) known gray levels into the inpainting domain. As we will see next, using a patch error function based on the gradient of the image yields methods which transfer gradients and compute the resulting image as the solution of a PDE. This results in better continuation properties of the solution, in particular at the boundary of the inpainting domain.

*Patch non-local Poisson.* The squared  $L^2$ -norm of the gradient in the image energy term (3) corresponds to taking  $e(\cdot) = \|\cdot\|^2$  in (7), where  $\|\cdot\|$  is the Euclidean norm in  $\mathbb{R}^2$ . The energy term becomes

$$\tilde{F}_w(u) = \int_O \int_{O^c} m(z, \hat{z}) \|\nabla u(z) - \nabla \hat{u}(\hat{z})\|^2 d\hat{z} dz. \quad (10)$$

Recall that we consider Dirichlet conditions at the boundary between the inpainting domain and the known data region, and Neumann conditions at the boundary of the image, *i.e.*  $u(x) = \hat{u}(x)$  in  $\partial O \setminus \partial \Omega$  and  $\nabla u(x) \cdot n(x) = 0$  for  $x \in \partial O \cap \partial \Omega$ . The Euler equation w.r.t.  $u$  is given



by a *non-local* Poisson equation, *i.e.* a Poisson equation with non-local coefficients:

$$\nabla \cdot [c(z)\nabla u(z)] = \nabla \cdot [c(z)\mathbf{v}(z)], \quad (11)$$

for all  $z \in O$ , where  $u|_{O^c} = \hat{u}$ ,  $c(z) = \int_{O^c} m(z, \hat{z})d\hat{z}$  and the field  $\mathbf{v} : O \rightarrow \mathbb{R}^2$  is given by

$$\mathbf{v}(z) = \frac{1}{c(z)} \int_{O^c} m(z, \hat{z})\nabla \hat{u}(\hat{z})d\hat{z}. \quad (12)$$

The solution of (11) is computed with a conjugate gradient algorithm.

Observe that the solutions of (11) are minimizers of

$$\int_O c(z)\|\nabla u(z) - \mathbf{v}(z)\|^2 dz.$$

Therefore,  $u$  is computed as the image with the closest gradient (in the  $L^2$  sense) to the *guiding* vector field  $\mathbf{v}$ , which corresponds to a non-local weighted average of the gradients in the complement. See [47] for further uses of the Poisson equation in image editing.

*Patch non-local gradient medians.* To complete our study of the image energy term we consider the  $L^1$ -norm of the gradient, that is

$$\tilde{F}_w(u) = \int_O \int_{O^c} m(z, \hat{z})\|\nabla u(z) - \nabla \hat{u}(\hat{z})\|d\hat{z}dz,$$

with the same boundary conditions as with the patch-wise NL-Poisson. To minimize it we perform an implicit gradient descent:

$$\min_{u^{t+1}} \int_O \int_{O^c} m(z, \hat{z})\|\nabla u^{t+1}(z) - \nabla \hat{u}(\hat{z})\|d\hat{z}dz + \frac{1}{2\delta t}\|u^{t+1} - u^t\|^2, \quad (13)$$

where at each step  $u^{t+1}$  is computed using the fixed point algorithm described in the Appendix B, based on the projection method presented in [18].

As patch NL-Poisson, this scheme transfers gradients and interpolates the gray levels using the boundary conditions. With the use of the  $L^1$  error function, we expect the solution of patch NL-GM to retain more small scale detail than that of patch NL-Poisson.

Notice that for both gradient-based methods the patch similarity weights  $w$  are computed based only on the gradients (and thus also the pixel-wise influence weights  $m$ ). In most cases however, the gradient is not a good feature for measuring the patch similarity, and it would be desirable to consider also the gray level data. This can easily be achieved within our variational framework by modifying the patch error function to take into account both gradients and gray level patches.

*Functionals combining intensity and gradients.* We can use any of the gradient-based energies in conjunction with the patch NL-means energy by considering a linear combination of the corresponding patch error functions. This yields

$$\tilde{F}_\lambda(u) = \int_O \int_{O^c} m(z, \hat{z})[(1-\lambda)\|\nabla u(z) - \nabla \hat{u}(\hat{z})\|^p + \lambda|u(z) - \hat{u}(\hat{z})|^2]d\hat{z}dz \quad (14)$$

where the parameter  $\lambda \in (0, 1)$  controls the mixture and  $p$  is either 1 or 2. The resulting algorithms update the weights  $w$  considering both the intensity and the gradient, therefore improving their selectivity.

Regarding the image update, notice that (14) can be rewritten as

$$\tilde{F}_\lambda(u) \propto \int_O \int_{O^c} m(z, \hat{z})\|\nabla u(z) - \nabla \hat{u}(\hat{z})\|^p d\hat{z}dz + \frac{\lambda}{1-\lambda} \int_O c(z)|u(z) - f(z)|^2 dz,$$

where  $f(z) = c(z)^{-1} \int_{O^c} m(z, \hat{z})\hat{u}(\hat{z})d\hat{z}$  is the solution of the patch NL-means image update step. Thus we see that the combination with the squared  $L^2$  patch error function translates into a patch NL-means attachment term.

For the case of patch NL-Poisson the Euler equation w.r.t.  $u$  becomes:

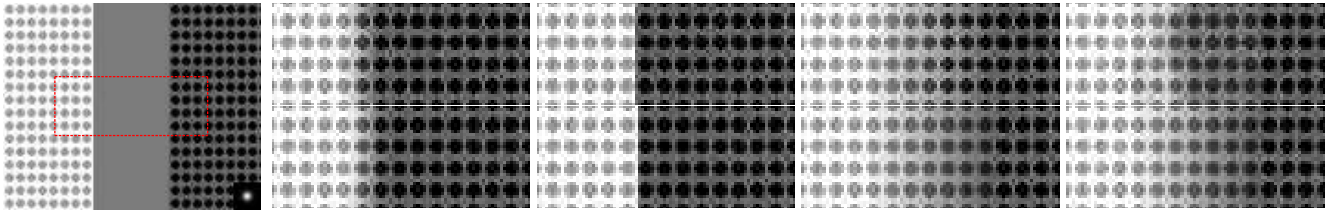
$$\begin{aligned} \nabla \cdot [c(z)\nabla u(z)] - \frac{\lambda}{(1-\lambda)}c(z)u(z) &= \\ &= \nabla \cdot [c(z)\mathbf{v}(z)] - \frac{\lambda}{(1-\lambda)}c(z)f(z), \end{aligned} \quad (15)$$

which is linear and can be solved with a conjugate gradient scheme.

The patch NL-GM combined functional has basically the same form as (13) and therefore can also be solved with the fixed point scheme described in Appendix B.

## 2.2 Comparison of proposed schemes

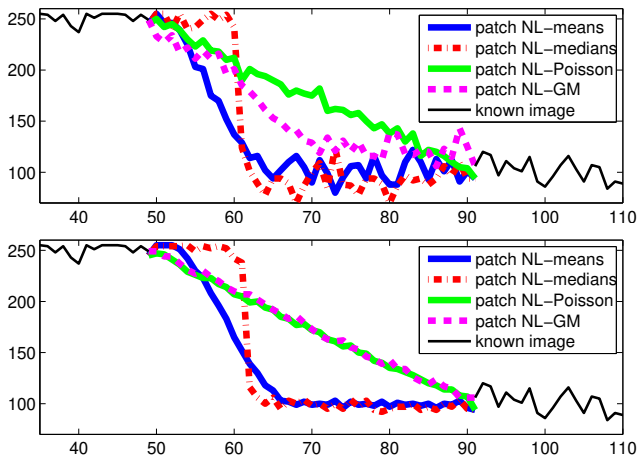
Let us advance some results on synthetic problems to highlight the main characteristics of the proposed methods. First we consider the inpainting of a regular texture (shown in Figure 2) with two different mean intensities, where the inpainting domain hides all patches on the boundary between the dark and bright textures. With this example we can test the ability of each method to *create* an interface between both regions. Situations like these are common in real inpainting problems, for instance due to inhomogeneous lighting conditions. We have also added Gaussian noise with standard deviation



**Fig. 2 Inpainting of a synthetic texture.** The initial condition is shown in the first column. The other four columns show a zoom (region in the red rectangle) of the results of patch NL-means, -medians, -Poisson and -GM. Top row,  $h = 0$ , bottom row  $h = 200$ ,  $h = 14$ ,  $h = 400$  and  $h = 20$ , respectively. The intra-patch weight kernel  $g_a$  is shown in the bottom right corner of the initial condition, it has a standard deviation  $a = 5$  and the patch size is  $s = 15$ .

$\sigma = 10$  to show the influence of the selectivity parameter  $h$ . Each column of Figure 2 shows the results of the four methods described in the previous section. We have tested each method with  $h = 0$  (top row), and  $h > 0$ , chosen approximately to match the expected deviation of each patch error due to the noise.

The first notorious difference is on how the methods handled the transition between the dark and bright textures. Patch NL-means produces a smooth transition whereas a sharp step is obtained with the patch NL-medians. On the other hand, both gradient based methods yield a much smoother shading of the texture. This is due to the fact that the image update step is computed as the solution of a PDE which diffuses the intensity values present at the boundary of the inpainting domain. These PDEs are driven by a vector field estimated non-locally and therefore combine *non-local* exemplar based inpainting with *local* interpolation PDEs. For the case of patch NL-Poisson this interpolation is linear, since this is a solution of the homogeneous Poisson equation (*i.e.* Laplace equation).

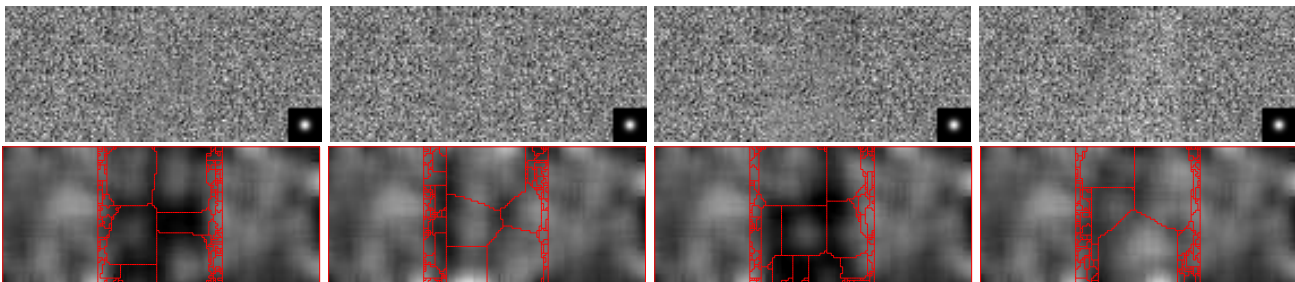


**Fig. 3 Profiles of the results in Figure 2.** The profiles are taken from an horizontal line going between the circles in Figure 2. Top: results with  $h = 0$  and bottom: results with  $h > 0$ .

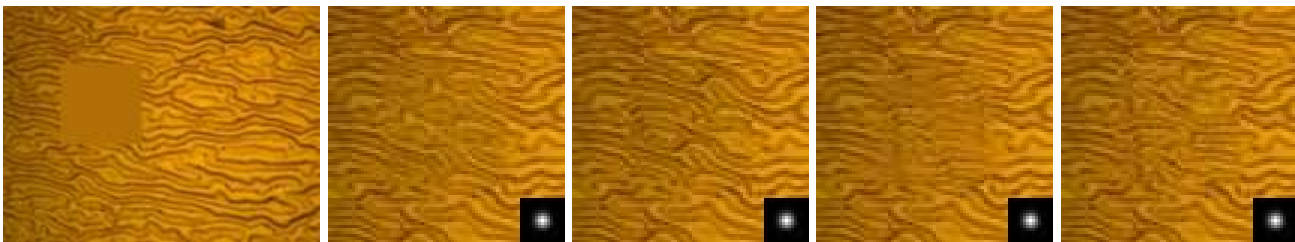
As expected, the results using a higher  $h$  show some denoising, since for larger  $h$  the patch similarity weights are less selective. This effect can be better appreciated in the profiles shown in Figure 3, which depict the image values for a horizontal line between the circles. In the usual context of inpainting, in which the available data is not perturbed by noise, this denoising translates into an undesirable loss of texture quality. For that reason we use  $h = 0$ .

Recall that in the limit when  $h = 0$ , the weights  $w(x, \hat{x})$  converge to a Dirac's delta function at the set  $n(x)$  of nearest neighbors of  $p_u(x)$ . Even if we assume that the nearest neighbor is unique, the value of a pixel is still computed from a population obtained from those nearest neighbors of all patches that overlap the pixel. In the case of periodic patterns, once the minimization has reached a stable state, all values in the population will be basically the same: patches surrounding a pixel agree on its value, and in this case all schemes behave similarly. Differences arise when neighboring patches cannot agree on their suggested values. Such is the case of the step in Fig. 2. For non-periodic patterns and random textures this disagreements will be common, which may affect the perceptual similarity of the synthesized texture with the original.

Figure 4 studies this effect. The image consists of Gaussian noise. The top row shows the resulting completions, and the bottom row shows an estimate of the local variance (computed by smoothing the image of the squared differences w.r.t. the mean). The red curves require a brief explanation. Following [7], we use the term *Nearest Neighbor Field* (NNF) to refer to the vector field  $n(x) - x$ , defined over  $\tilde{O}$ , where  $n(x) \in \tilde{O}^c$  is the position of the (assumed) unique nearest neighbor of  $p_u(x)$ . In Figure 4 we show in red the boundaries of the regions with constant NNF. In those regions patches are translated rigidly from somewhere in the complement. We can observe that, for all inpainting schemes, the resulting completion consists basically of a *patch-work* of large regions of rigid translation. The interior of these regions, away from their boundaries (red curves),



**Fig. 4 Inpainting of random texture.** From left to right: patch- NL-means, NL-medians, NL-Poisson and NL-GM. Top row: inpainting results. Bottom row: local variance estimate of the noise, superimposed with the boundaries between regions of constant NNF (red curves). Notice the attenuation of the variance over the red curves, specially for  $L^2$ -based methods.



**Fig. 5 Inpainting of structured texture.** From left to right: Initial condition, result of patch NL-means, -medians, -Poisson and -GM. Results with  $s = 15$ ,  $a = 5$  and  $h = 0$ . The treatment of color images is described in Section 5.

reproduces an exact copy of the source, and thus the variance is preserved. The variance decreases along the red curves, where pixel values are synthesized using incoherent contributions. Both for the intensity models or the gradient models, the  $L^2$ -norm causes a higher decay of the variance than with the  $L^1$ -norm. This effect can also be observed in figures 2 and 3, where it can be seen that patch NL-medians and -GM preserve more of the Gaussian noise than the methods based on  $L^2$ -norms. We will refer to the regions of constant NNF as *copy fronts*. Recall however that copy occurs only away from their boundaries.

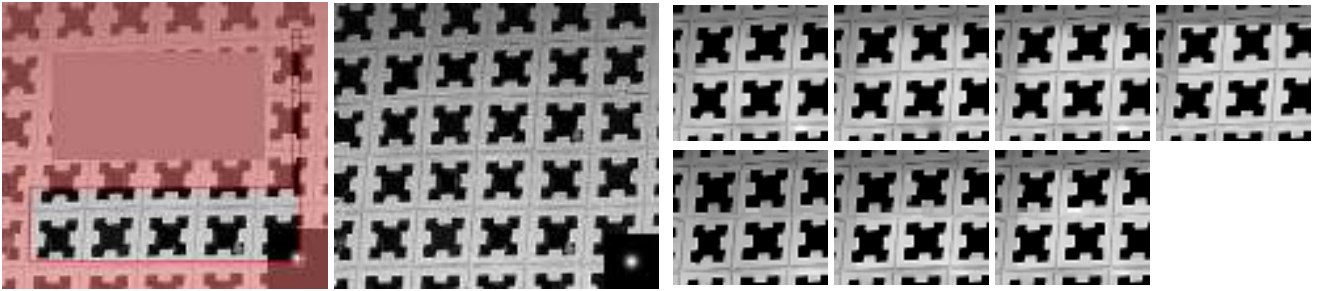
In Figure 5 we show results on a non-periodic, structured texture. We can see that patch NL-means and -Poisson show some smoothing, whereas both  $L^1$ -based schemes obtained sharper results. Both intensity based methods show discontinuities at the boundary of the inpainting domain. Notice also that for the patch NL-medians algorithm it is easy to identify the regions in the complement that have been replicated. Where two copy fronts meet a seam is produced. With the patch NL-means the spatial averaging of overlapping patches creates a smooth blending of the copy fronts.

### 2.2.1 Combined schemes

Gradient based methods produce smooth interpolations and enforce the continuity of the image at the boundary of the inpainting domain, which are generally desirable features. For this to be done variationally the similar-

ity weights  $w$  have to be computed using patches of the gradient, which in most cases does not provide a reliable measure of patch similarity. In practice better results are obtained by combining these methods with patch NL-means, allowing to take the image values into account for the computation of the patch error, and to synthesize the image with a diffusion PDE. This adds a parameter  $\lambda$ , which controls the mixture.

In Figure 6 we show some results corresponding to the combination of both gradient schemes with patch NL-means while varying the mixture coefficient  $\lambda$ . The image shows a periodic pattern with an illuminance gradient. Most of the dark exemplars are incomplete, and thus only bright exemplars from the bottom of the image are available. The rightmost detail shows the result of patch NL-means: the image has been completed using bright patches and presents a discontinuity on the upper side of the hole. On the other hand, a completion using gradients only (see result of the patch NL-GM) manages to interpolate both the texture and the shading. The small images on the right show results of both gradient methods with different values for  $\lambda$ . The value of the mixing parameter  $\lambda$  should be carefully selected since it mixes two different magnitudes (norms of gradients and gray levels). With  $\lambda \sim 0.1$  for patch NL-Poisson + NL-means and  $\lambda \sim 0.01$  for patch NL-GM + NL-means, some of the good continuation properties are preserved and enough color information is added to the patch metric.



**Fig. 6** Linear combination of gradient based methods with NL-means. First image starting from the left: Initialization. The gray rectangle is the inpainting domain. Only patches centered outside the red area are available. Second image: Result obtained with patch NL-GM, using  $\lambda = 0$ . A similar result is obtained with the patch NL-Poisson with  $\lambda = 0$  (not shown). Details: Top, from left to right: results with patch NL-Poisson with  $\lambda = 0.01, 0.05, 0.1, 1$  (the latter corresponds to patch NL-means). Bottom, patch NL-GM with  $\lambda = 0.001, 0.005, 0.01$ . All details have been linearly stretched for display. Differences are most noticeable in the top and in the left of the images.

### 2.2.2 Geometric interpolation

To evaluate the ability of each method to continue the geometry of the structures at the boundary of the inpainting domain, we consider a very simple image with a gap (shown in Table 1). The inpainting region is initialized with the background color. For this evaluation we fix the size of the patch and increase the width of the gap. For narrow gaps, the method will be able to join both ends of the green vertical line. When increasing the gap, at a certain width the method is no longer capable of recovering the vertical line. In these cases the gray initialization prevails, showing a completion in which the vertical line is interrupted. The first column of Table 1 shows the maximum height of the gap for each method. Observe that the combined schemes improve the prolongation of geometric structures, and that the optimal mixing parameter  $\lambda$  for this purpose is near to the values that we have proposed in the previous section. Basically these schemes have two propagation mechanisms: A *local* one, by diffusion of the intensity values by the PDE, and a *non-local* one by transference of gradients from  $O^c$ . When  $\lambda > 0$  these mechanisms reinforce each other: the diffused values allow a better estimation of the weights, and therefore the transference of more appropriate gradients, which will help to diffuse the intensity values further. On the other hand, intensity based methods depend only on the iteration of weights computation and image update to propagate information.

The second column of Table 1 shows the results obtained by incorporating the confidence mask later described in Section 2.3. An alternative way to prolong the geometric structures is to increase the patch size as will be discussed in Section 4.

Method	GAP	GAP $t_c = 5$
P + M $\lambda = 0$	13	40
P + M $\lambda = 0.1$	16	46
P + M $\lambda = 0.5$	11	34
P + M $\lambda = 1$	9	29
Md	7	42
GM + M $\lambda = 0$	15	> 56
GM + M $\lambda = 0.01$	21	> 56
GM + M $\lambda = 0.001$	40	> 56

**Table 1** Geometric interpolation. The inpainting domain is shown in white and the patch in the lower right corner ( $9 \times 9$  pixels). The table reports the maximum gap width for which the algorithm is capable of recovering the vertical line. P, M, Md and GM stand for patch NL-Poisson, -Means, -Medians and -Gradient Median respectively. The rightmost column shows the maximal gaps obtained with the use of a confidence masks with decay  $t_c = 5$  (see Section 2.3).

### 2.3 Extensions

*Color images.* An energy for color images can be obtained by defining a patch error function for color patches as the sum of the error functions of the three scalar components:

$$\varepsilon(p_{\mathbf{u}}(x) - p_{\hat{\mathbf{u}}}(\hat{x})) = \sum_{i=1}^3 \varepsilon(p_{u_i}(x) - p_{\hat{u}_i}(\hat{x})),$$

where  $\mathbf{u} : \Omega \rightarrow \mathbb{R}^3$  is the color image, and  $u_i$ , with  $i = 1, 2, 3$ , its components (analogously for gradient-based errors). Given the weights, each channel is updated using the corresponding scheme for scalar images. All channels are updated using the same weights.

*Confidence mask.* For large inpainting domains, it is useful to introduce a mask  $\kappa : \Omega \rightarrow (0, 1]$  which assigns



a confidence value to each pixel, depending on the certainty of its information (see also [22, 39]). This will help in guiding the flow of information from the boundary towards the interior of the hole, eliminating some local minima and reducing the effect of the initial condition. The resulting image energy term takes the form

$$\tilde{F}_w(u) = \int_{\tilde{O}} \int_{\tilde{O}^c} \kappa(x) w(x, \hat{x}) \varepsilon(p_u(x) - p_{\hat{u}}(\hat{x})) d\hat{x} dx,$$

where  $\kappa$  modulates the penalization of the incoherences between  $w$  and the error  $\varepsilon$  between patches.

The effect of  $\kappa$  on the image update step can be seen on the pixel-wise influence weights  $m$

$$m(z, \hat{z}) = \int_{\Omega_p} \chi_{\tilde{O}^c}(\hat{z} - y) g_a(y) \kappa(z - y) w(z - y, \hat{z} - y) dy.$$

Thus, the contribution of the patch  $p_u(z - y)$  to the evidence function is now weighted by its confidence. Patches with higher confidence will have a stronger influence. Notice that with the inclusion of the confidence mask, the normalization coefficient  $c(z)$  becomes:

$$c(z) = \int_{O^c} m(z, \hat{z}) d\hat{z} = \int_{\Omega_p} g_a(y) \kappa(z - y) dy.$$

This does not affect intensity-based methods, but has implications on the image update step for gradient-based ones. For example, for patch NL-Poisson the errors with respect to the non-local guiding vector field  $\|\nabla u(z) - \mathbf{v}(z)\|$  are penalized according to  $c(z)$ .

On the similarity weights, the confidence mask has the effect of modifying the selectivity parameter  $h$  by a locally varying  $h/\kappa(x)$ . If the confidence is high, the effective selectivity  $h/\kappa(x)$  will be lower, thus increasing the selectivity of the similarity measure. When  $h \rightarrow 0$  the weights tend to a Dirac's delta independently of  $\kappa$ . The same reasoning applies to the gradient based energies.

For the experiments shown in this paper, the confidence mask was set to

$$\kappa(x) = \begin{cases} (1 - \kappa_0) \exp\left(-\frac{d(x, \partial O)}{t_c}\right) + \kappa_0 & \text{if } x \in O, \\ 1 & \text{if } x \in O^c, \end{cases}$$

which shows an exponential decay w.r.t. the distance to the boundary inside the hole  $d(\cdot, \partial O)$ . Here  $t_c > 0$  is the decay time and  $\kappa_0 > 0$  determines the asymptotic value reached far away from the boundary. Setting  $t_c = 0$  amounts to using a constant confidence mask. Table 1 shows the effect of using a confidence mask with  $t_c = 5$  and  $\kappa_0 = 0.1$ , allowing the restoration of the vertical line for much wider gaps, and thus alleviating the dependence with the gray initial condition.

Other shapes of the confidence mask could be used for controlling different aspects of the dynamics of the

completion algorithm. For instance controlling the decay of the mask from certain points of the boundary allow to privilege the continuation of structures from them.

*Interpolation of sparsely sampled images.* In this work we have implicitly assumed the existence of a set of complete patches  $\tilde{O}^c$ , constituting the set of exemplars. This assumption can be removed by generalizing the patch error function  $\varepsilon(\cdot)$  so that it depends on the positions of the patches. This implies a potentially different error function for each pair of patches. This has been explored in [29], applied to the case of interpolation of sparsely sampled images. In that case the patch error function was designed to consider only those locations known in at least one of the two patches.

### 3 Discussion and connections

In this section we present various issues related to the variational framework and its connections with other models.

#### 3.1 Probabilistic-geometric model interpretation

The proposed energy has an interpretation in terms of a probabilistic model in the space of patches, which becomes apparent when rewritten using the generalized Kullback-Leibler divergence [23]. Given two positive and integrable functions  $p, q$  defined over a certain measure space  $\mathcal{X}$ , the generalized Kullback-Leibler divergence is given by:

$$\text{KL}(p, q) = \int_{\mathcal{X}} p(s) \log\left(\frac{p(s)}{q(s)}\right) ds - \int_{\mathcal{X}} p(s) ds + \int_{\mathcal{X}} q(s) ds,$$

assuming that the integrals exist. With this notation (and taking into account that  $w(x, \cdot)$  is a probability) the functional  $E$  can be written as

$$E(u, w) = \int_{\tilde{O}} \text{KL}(w(x, \cdot), r(x, \cdot)) dx - \int_{\tilde{O}} \int_{\tilde{O}^c} r(x, \hat{x}) d\hat{x} dx,$$

where  $r$  is the unnormalized Gaussian weight function

$$r(x, \hat{x}) = \exp\left(-\frac{1}{h} \varepsilon(p_u(x) - p_{\hat{u}}(\hat{x}))\right).$$

The first term integrates the divergence between the functions  $w(x, \cdot)$  and  $r(x, \cdot)$ , for each  $x \in \tilde{O}$ . The second term can be interpreted by noticing that

$$\tilde{q}(x) = \int_{\tilde{O}^c} r(x, \hat{x}) d\hat{x} \quad (16)$$

is a density estimate (in the patch space) of the set of patches in  $O^c$ : The higher the amount of patches in  $\tilde{O}^c$  close to  $p_u(x)$  (according to the scale parameter  $h$ ), the higher the value of  $\tilde{q}$ .

The minimizers  $(u^*, w^*)$  are obtained when for all  $(x, \hat{x}) \in \tilde{O} \times \tilde{O}^c$ ,  $w^*(x, \hat{x}) = r^*(x, \hat{x})/\tilde{q}^*(x)$ , (Gaussian weights normalized by (16)), and the patches of the inpainted image are in regions of high density in the patch space. This provides a geometric intuitive interpretation of our variational formulation. The image is considered as an ensemble of overlapping patches. Known patches in  $\tilde{O}^c$  are fixed, forming a patch density model used to estimate the patches in  $\tilde{O}$ . The richness of the framework is given in part by the fact that different norms in the patch space induce inpainting schemes of different nature.

### 3.2 Connections with statistical mechanics

The proposed energy can be given another interpretation in terms of statistical mechanics, as the *Helmholtz free energy* of a system of particles. In this context we consider that for each ‘‘particle’’  $x \in \tilde{O}$  there is a set of possible configurations indexed by the parameter  $\hat{x} \in \tilde{O}^c$  with a probability density  $w(x, \hat{x})$ . If we consider that each configuration has an energy  $U(x, \hat{x}) = \varepsilon(p_u(x) - p_u(\hat{x}))$ , then the probability that the particle at  $x$  is in the state  $\hat{x}$  is proportional to

$$e^{-\beta U(x, \hat{x})}/q(x)$$

with the normalization factor  $q$  given by the partition function

$$q(x) = \sum_{\hat{x}} e^{-\beta U(x, \hat{x})}.$$

Then the Helmholtz free energy for a particle  $x$  is

$$\begin{aligned} \mathcal{H}(x) &:= \overline{U(x)} + \beta^{-1} \mathcal{E}(x) := \\ &:= \sum_{\hat{x}} U(x, \hat{x}) w(x, \hat{x}) + \beta^{-1} \sum_{\hat{x}} w(x, \hat{x}) \log w(x, \hat{x}). \end{aligned}$$

Then the total Helmholtz free energy for the system of particles is

$$\begin{aligned} \mathcal{H} &:= \sum_x \mathcal{H}(x) = \sum_x \sum_{\hat{x}} U(x, \hat{x}) w(x, \hat{x}) + \\ &\quad \beta^{-1} \sum_x \sum_{\hat{x}} w(x, \hat{x}) \log w(x, \hat{x}), \end{aligned}$$

where the sums are extended to  $x \in \tilde{O}$  and  $\hat{x} \in \tilde{O}^c$ . We have used the notation  $\beta$  as  $\frac{1}{h}$ .

### 3.3 Revisiting related work

In this section we will briefly review the connections of our work with other inpainting algorithms and also with existing variational models of non-local regularization which have been proposed in contexts such as image denoising.

The method in [56] is closely related to the patch NL-means scheme of Eq. (9). The key difference lies in the underlying theoretical model. The problem is addressed as a MRF, where pixels outside the hole are observable variables, missing pixels in the hole are the parameters, and the hidden variables are given by the correspondence  $\Gamma : O \rightarrow O^c$ , which assigns a patch outside the hole to each  $x$  in  $O$ . The method can be seen as an approximate EM algorithm for maximizing the log-likelihood w.r.t. the pixels in  $O$ , and some approximations have to be taken to make the optimization tractable. Based on heuristics, the authors also propose to use more robust estimators than the mean for the synthesis of pixels. Within the framework here proposed, robust estimators (as the median) naturally result from particular choices of the patch error functions  $\varepsilon(\cdot)$ .

In [38] an energy is presented which can be seen as a limit case of the patch NL-means energy when  $h \rightarrow 0$ . The authors propose modifications of the energy which improve the results, such as some spatial localization of the similarity weights and brightness invariance. The latter is achieved by introducing a multiplicative constant that matches the mean illuminance between each pair of patches.

The patch NL-means algorithm is also related to the manifold image models of [49]. Eq. (9) can be split into two steps which are analog to the manifold and image projection steps used in [49]. First, for each patch centered in  $\tilde{O}$  we compute a new patch as a weighted average of all patches in the complement, according to the patch similarity weights  $p_u^{MS}(z) := \int_{\tilde{O}^c} w(z, \hat{z}) p_{\hat{u}}(\hat{z}) d\hat{z}$  with  $z \in \tilde{O}$ . Doing this for each hole position yields an incoherent ensemble of patches. The image is obtained by averaging these patches:  $u(z) = \frac{1}{\mathcal{A}(\tilde{O}_p)} \int_{\tilde{O}_p} p_u^{MS}(z - y, y) dy$ . We use a density model, instead of the manifold model of [49]. Indeed,  $p_u^{MS}(x)$  is the *mean shift* operator applied to  $p_u(x)$ . It is known that the iteration of this operator corresponds to an adaptive gradient ascent of the Parzen estimate of a PDF [21], which in this case is generated by the set of patches in the complement of the hole. The use of a density model entails



some advantages, mainly from the computational point of view, learning a manifold model is computationally costly. Furthermore, the assumption that patches lie on a manifold is questionable (one could think for instance in a *stratification* as a more realistic model), and its dimension is hard to determine for real images.

In the following we will comment on the relation of this model with recent works on non-local regularization.

The UINTA algorithm, presented in [5] is a non-local denoising algorithm that minimizes the entropy of the patches in the image. Casting this idea to the context of inpainting the UINTA's entropy is estimated as the sample mean

$$E_U(u) = - \int_{\tilde{O}} \log \left[ \int_{\tilde{O}^c} \exp\left(-\frac{1}{h} \|p_u(x) - p_{\hat{u}}(\hat{x})\|^2\right) d\hat{x} \right] dx$$

where the inner integral is the probability of occurrence of the patch  $p_u(x)$  obtained as a Parzen density estimate. The corresponding Euler-Lagrange equation can be solved with a fixed point iteration which coincides with the patch NL-means scheme (9). In [5] this energy is minimized by considering all patches as independent (disregarding the overlap between neighboring patches), and evolving each of them according to a gradient descent of  $E_U$ . After this, an image is formed with the centers of these new patches. The repetition of this process results in an iterative application of *pixel-wise* NL-means.

In [13] the authors use a variational principle for deriving the iterated pixel NL-means regularizer, and show its application to the restoration of texture. The underlying energy corresponds to the quadratic penalty between the solution image  $u$ , and a pixel NL-means type average of the noisy input image  $\hat{u}$ . The weights for this average are computed using  $u$ . Due to the dependence of the weights with the regularized image  $u$ , the minimizer is no longer a weighted average as NL-means, but the solution of a nonlinear optimization problem. It is shown that if the derivative of the nonlinear component is neglected, the resulting Euler-Lagrange equation matches the proposed fixed point algorithm: The iterated NL-means regularizer.

In [52] the authors presented a variational framework for image denoising consisting in non-local regularization and data adjustment terms. Inpainting could be performed by considering only the data term as follows:

$$E_P(u) = - \int_{\tilde{O}} \int_{\tilde{O}^c} \exp(\varepsilon(p_u(x) - p_{\hat{u}}(\hat{x}))) d\hat{x} dx$$

This energy is the same as the one adapted from the UINTA algorithm  $E_U$ , without the logarithm. In [52]

the Euler-Lagrange equation is solved with a fixed point iteration. This model has two differences with our framework. First it allows to use a more general nonlinearity for the computation of the weights other than the exponential. Second, even in the case of the exponential, the methods differ in the normalization, for instance, when  $\varepsilon$  is the squared  $L^2$ -norm, the resulting scheme is as the patch NL-means, with the unnormalized weights.

After its introduction in [3], our model has been interpreted as a non-local self-similarity regularizer in [51], where in conjunction with appropriate data fitting terms it has been applied to the solution of inverse problems, including inpainting, super-resolution and compressive sensing. In [51] a different patch-error function  $\varepsilon$  is used, namely the  $L^2$ -norm between patches (without squaring it). This choice is motivated as a patch-wise version of their work [50] on non-local Total Variation [32, 42, 57] with adaptive weights. This patch-wise non-local TV is defined as the  $L^1$ -norm of the non-local gradient of the patch valued image  $p_u : \Omega \rightarrow \mathbb{R}^{\Omega_p}$ . The non-local gradient is defined as a function  $\nabla_w p_u : \Omega \times \Omega \rightarrow \mathbb{R}^{\Omega_p}$  given by  $\nabla_w p_u(x, \hat{x}) = w(x, \hat{x})(p_u(x) - p_u(\hat{x}))$ . Thus, the patch-wise non-local TV reads

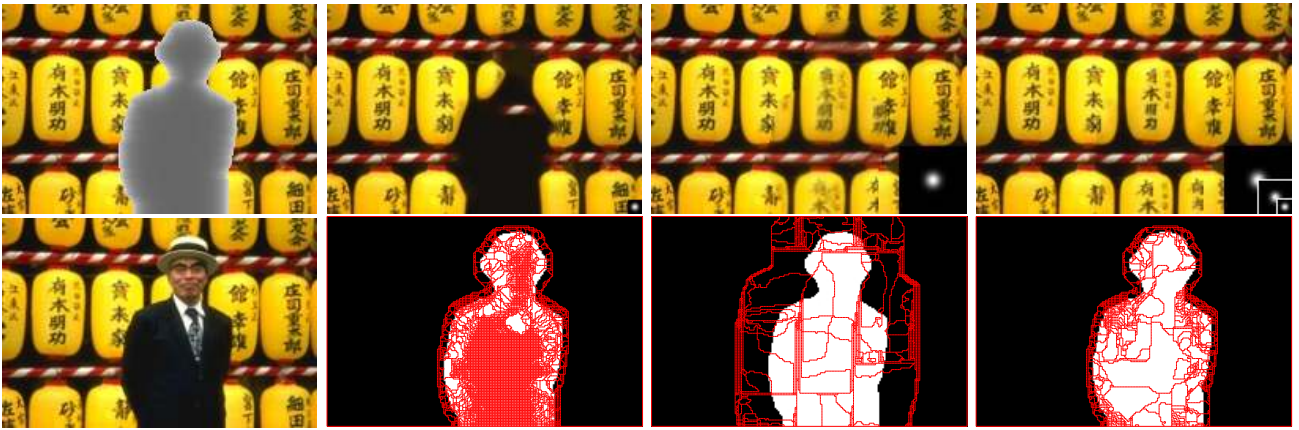
$$\|\nabla_w p_u\| := \int_{\Omega} \int_{\Omega} w(x, \hat{x}) \|p_u(x) - p_u(\hat{x})\|_2 d\hat{x} dx.$$

Note that in this sense, the model of the patch-NL medians corresponds an anisotropic version of the non-local TV where the 2-norm in the integral is replaced by the 1-norm. Our work and the work of [51] are complementary. In [51] the regularization term is fixed, and the authors focus on the possibilities given by different data terms suited for different applications. On the other hand in this work we focus on the regularization term exploring its properties with different patch error functions  $\varepsilon$ , and applying them to a problem in which the data term plays no role at all, since there is no data to adjust to.

## 4 Multiscale scheme

Exemplar-based inpainting methods show a critical dependence with the size of the patch. In Figure 7, we show completions obtained with patch NL-means using different patch sizes: Two results with a small patch ( $a = 4$ ) and one result with a large patch ( $a = 19$ ). The latter is able to reproduce the periodic pattern of the lamps, but the completion is blurry due to the spacial overlap of the patches and presents many discontinuities at the boundary of the hole.

The results with the small patch do not show these artifacts, but one of them has failed to reproduce the



**Fig. 7** Mitama Matsuri. Left column: inpainting domain and initial condition. For the rest of the columns, from left to right single scale inpainting with a  $9 \times 9$  patch with  $a = 4$ , single scale inpainting with a  $43 \times 43$  patch with  $a = 19$  and multi-scale inpainting, with  $S = 2$ , covering a patch size from  $9 \times 9$  with  $a = 4$  to  $43 \times 43$  with  $a = 19$ . All results have been computed with the patch NL-means scheme. The bottom row shows the regions in which the nearest neighbor offset is constant.

lamps. The only difference between both is the initialization. One of them was initialized with the original image shown in the bottom left, whereas the other one with the result obtained with the multiscale approach described in this section.

Figure 8 shows a synthetic problem with similar characteristics. The result with a large patch recovers the shape of the incomplete rectangle, but does not continue its texture correctly. Three different small scale results are shown. They are all minima of the energy with minimal energy,  $E = 0$ . Patches are too small to capture that these completions are different. Which one is obtained will depend on the initialization.

As done by almost all state of the art exemplar based inpainting methods (*e.g.* [38, 39, 56]), we will incorporate a multiscale scheme, described below. This is usually motivated as an heuristic to avoid local minima, to find a good initialization and/or to alleviate the computational cost. We believe however (following [34]) that, as these examples suggest, inpainting is inherently a multiscale problem: Images have structures of different sizes, ranging from large objects to fine scale textures and edges. The multiscale scheme responds to the fact that several patch sizes are needed to reproduce all these structures properly.

#### 4.1 Multiscale scheme

In the following we describe the multiscale method we adopted in this work, which goes along the lines of what is customary in the literature [30, 38, 56]. It consists on sequentially applying the inpainting scheme on a Gaussian image pyramid, starting from the coarsest scale. The result at each scale is upsampled and used as ini-

tialization for the next finer scale. The patch size is constant through scales.

Let us consider  $S$  scales, the finest denoted with  $s = 0$ . We will specify the size of the image at the coarsest level  $A_{S-1}$ . Denoting the size of the image at the finest scale by  $A_0$ , we compute the sampling rate as  $r := (A_0/A_{S-1})^{1/(S-1)} \in (0, 1)$ . The width of the Gaussian filtering is associated to the subsampling factor as in [46]. Let  $a_0$  be the size of the patch and  $E_{a_0}$  the corresponding energy. We will add the superindex  $s = 0, \dots, S - 1$  to the variables  $u$  and  $w$  to denote the scale. As before, the subindex 0 refers the initial condition, *i.e.*  $u_0^s$  is the initial condition at scale  $s$ .

---

#### Algorithm 2 Multiscale scheme.

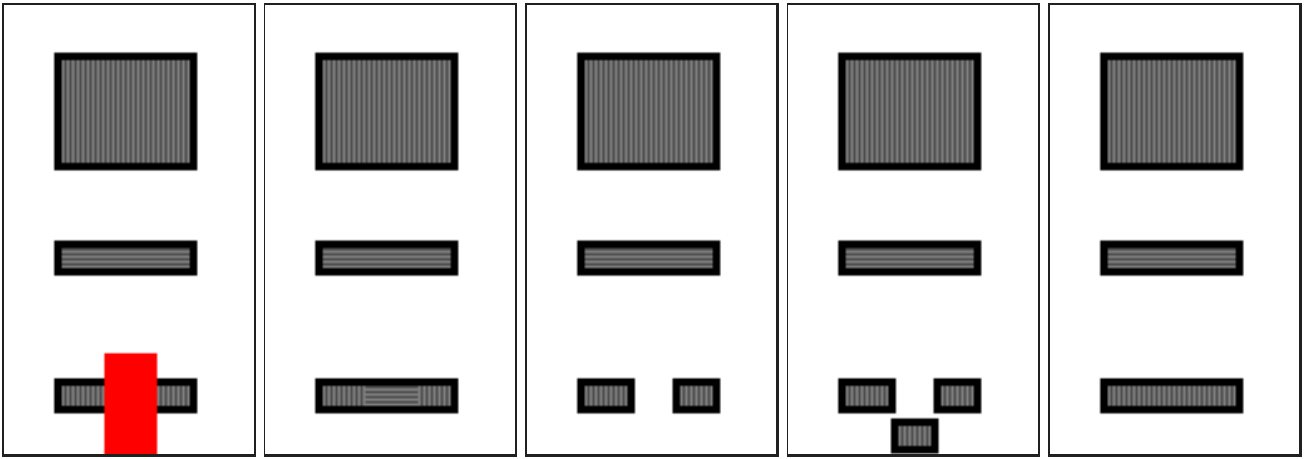
---

**Require:**  $u_0^S$ ,  $S$ ,  $a_0$  and  $A_{S-1}$

- 1: Initialize:  $(u^{S-1}, w^{S-1}) = \arg \min_{(u,w)} E_{a_0}(u, w)$
  - 2: **for** each scale  $s = S - 2, \dots, 0$  **do**
  - 3:   Upsample  $u^{s+1}$  to obtain  $u_0^s$
  - 4:    $(u^s, w^s) = \arg \min_{(u,w)} E_{a_0}(u, w)$
  - 5: **end for**
- 

The upsampling from  $s + 1$  to  $s$  is obtained as in [56]. The coarse weights  $w^{s+1}$  are first interpolated to the finer image size, yielding  $w_0^s$ . These weights are then used to solve an image update step at the new scale:  $u_0^s = \min_u E_{a_0}(u, w_0^s)$ . More conventional upsampling schemes by local interpolation (such as bilinear or splines) introduce a bias towards low-frequency non-textured regions. This exemplar-based upsampling avoids this bias.

Notice that keeping the patch size constant while filtering and reducing the image, is almost equivalent to enlarging the patch domain and filtering an image



**Fig. 8 Single scale inpainting.** Synthetic inpainting problem (the red rectangle is the inpainting domain) and several local minima of a single scale inpainting energy with a small patch.

of constant size. The process can thus be seen as the sequential minimization of a series of inpainting energies with varying patch size given by  $a_s = 1/r^s a_0$ ,  $s = 0, \dots, S - 1$ , over a corresponding series of filtered images.

The scheme starts by the coarsest scale  $S - 1$ , in which a larger portion of the inpainting domain is covered by partially known patches, which makes the inpainting task easier, and less dependent on the initialization. The energy at this scale should have fewer local minima. The dependency of the minimization process on the initial condition ensures also that each single scale solution corresponds to a local minimum close to the coarse scale initialization. Therefore the multiscale algorithm *exploits* this dependency to obtain an image  $u^0$  which is approximately self-similar for all scales (or equivalently, for all patch sizes).

Figure 7 shows a comparison between single and multiscale results with the patch NL-means scheme. As expected the multiscale result shows the benefits of large and small patch sizes. The missing lamps have been completed with the correct shape and spacing by the coarser stages, and the fine details are also taken care off by the finer scales. The bottom row shows the regions of constant NNF. The single scale result with a large patch shows an NNF partitioned in relatively large regions, showing that with a large patch size the copying is more rigid. The result with a small patch size shows that the synthesis has been performed based on small regions. The multiscale’s NNF shows an intermediate partition, with some large regions inside of the hole and a smaller ones around its boundary. The finer inpaintings work by refining the coarse partition obtained at coarser scales.

## 5 Experimental results

In this section we further demonstrate the performance of the proposed schemes on real inpainting problems. The images used were obtained from Komodakis and Tziritas [39] and from the 100 images benchmark proposed by Kawai *et al.* [38], available at <http://yokoya.naist.jp/research/inpainting/>. Due to the lack of space we focus on the methods presented in this work. Please refer to [38,39] for a comparison with their approaches.

### 5.1 Experimental setting

We consider four inpainting methods, namely patch NL-means, -medians, -Poisson and -GM. Both gradient based methods are always combined with patch NL-means with mixing parameter  $\lambda$ , as in (14). In all cases we use the multiscale approach. As stated before, to prevent blurring we set  $h$ , the selectivity of the similarity weights  $w$ , to  $h \rightarrow 0$ . In this case, the weights select the nearest neighbors of each patch in  $\tilde{O}$ . We show results with the CIE La\*b\* color space.

The calculation of the weights dominates the computational load of the algorithms. With an exhaustive search for the exact nearest neighbor, the cost of each iteration is  $\mathcal{O}(\mathcal{A}(O) \times \mathcal{A}(O^c) \times s^2)$ . However, a significant speed-up can be obtained with approximate searches, almost without any noticeable decrement in the quality of the results (see [13] and references therein).

In our implementation we use a modified version of the PatchMatch algorithm introduced in [7]. PatchMatch is an iterative algorithm that estimates the NNF jointly for all patches in the inpainting domain by exploiting the coherence of natural images. The modifi-

cation we implemented allows the estimation of a lists of the  $L$  first nearest neighbors of each patch. The algorithm has a computational cost of  $\mathcal{O}(\mathcal{A}(O) \times s^2 \times L)$  per iteration. Typically between 5 and 10 iterations are sufficient to obtain results comparable with the exhaustive search algorithm when using lists of size  $L \simeq 10$ . In Appendix C we describe the modified PatchMatch.

The image update step is fast for the patch NL-means, -median and -Poisson, but can be time consuming for patch NL-GM, in particular for small values of the mixing parameter  $\lambda$ . See Appendix B for more details.

The results are shown in figures 9, 10 and 11. The first column shows the initialization. The rest of the columns show the results obtained with patch NL-means, -medians, -Poisson and -GM in that order. Obtaining good results requires fixing the following parameters:

**Patch size.** For almost all experiments we used patches of size  $s$  between  $3 \times 3$  and  $9 \times 9$ . We did not use Gaussian intra-patch weights to shape the patch, since they need a larger support, which implies less available exemplars.

**Multiscale parameters.** The multiscale scheme has two parameters: the size of the coarsest image  $A_{S-1}$  and the number of scales  $S$ . From these,  $A_{S-1}$  is the most critical parameter. Smaller images with smaller inpainting domains are easier to complete. Care must be taken, however, since small images also imply less available exemplars to copy from. In these experiments  $A_{S-1}$  was set to a 20% of the original size when possible. Some cases required less subsampling. The number of scales  $S$  was set such that the subsampling rate  $r = (A_0/A_{S-1})^{1/(S-1)} \approx (1/2)^{1/3} \approx 0.8$  as in [56].

**Confidence mask.** The confidence mask has two parameters, the asymptotic value  $c_0$  and the decay time  $t_c$ . For all experiments we fix  $c_0 = 0.1$  and used a decay time  $t_c = 5$  except for small inpainting domains, in which we set  $t_c = 1$ .

For the mixing coefficient  $\lambda$  of gradient-based methods we tested two configurations: Low- $\lambda$  corresponding to  $\lambda = 0.01$  and  $\lambda = 0.001$  for patch-NL Poisson and -GM, and high- $\lambda$  corresponding to  $\lambda = 0.1$  and  $\lambda = 0.01$  for patch-NL Poisson and -GM. Recall that lower values of  $\lambda$  give a higher weight to the gradient component of the energy. This is appropriate for structured images with strong edges. In almost each row of figures 9, 10 and 11 we used the same  $\lambda$  configuration for both gradient-based schemes.

The rest of the parameters are also the same for each row in figures 10 and 11. Instead, for many images in Fig. 9 we used different parameters for intensity- and

gradient-based methods. In these cases, image-based methods required larger patches than gradient-based ones.

## 5.2 Observations and comments

*Gradient vs. intensity.* Gradient-based methods show a better reconstruction of structure (Fig. 9) and periodic patterns (Fig. 11). Intensity-based methods present discontinuities at the boundary of the hole and between copy fronts. Furthermore, gradient-based methods facilitate the prolongation of structures and edges, due to the reinforcement of local PDE diffusion and non-local propagation (see Section 2.2). This allows to use smaller patches, which alleviates the computational load and reflects in more available exemplars and less blending due to patch overlap (either by averages or medians).

On the other hand, gradient-based methods show problems with random textures (Fig. 10) and in some occasions between copy fronts. In these cases, the nearest neighbors of overlapping patches disagree on the gradient of a certain pixel. This may result in the attenuation or the omission of gradients, causing a “spilling effect”, particularly for the patch NL-Poisson (*e.g.* rows 1,3 and 4 of Fig. 9).

Some failures of gradient-based methods on random textures result from the use of gradients in the patch error function, as in rows 4 and 6 of Fig. 10. In the former, for instance, segments of the sky have been reproduced in the snow. Fig. 12 shows results with an inpainting scheme in which the weights are computed based only on the image values (with the squared  $L^2$ -norm), and the image is updated using patch NL-Poisson or -GM with a low value of  $\lambda$ . Such scheme is non-variational and its convergence its not guaranteed.

We should point out that in many other cases, considering gradients in the comparison criterion improves the results. See for instance rows 2 and 4 of Fig. 11.

*Means vs. medians.* It is notorious that  $L^1$ -based functionals perform better at the reproduction of fine texture. The results of the  $L^2$  methods are smoothed by the spatial averaging of overlapping patches. On the other hand, patch NL-medians creates sharp discontinuities as in Fig. 2 when different copy fronts meet (*e.g.* rows 1,3,5 and 7 in Fig. 9). These discontinuities are very noticeable and in these cases some smoothing is desirable. For the patch NL-GM method, discontinuities between copy fronts occur at the gradient level, and are eliminated after its integration. The fact that the gradient is estimated robustly helps in avoiding the spilling effect of patch NL-Poisson.



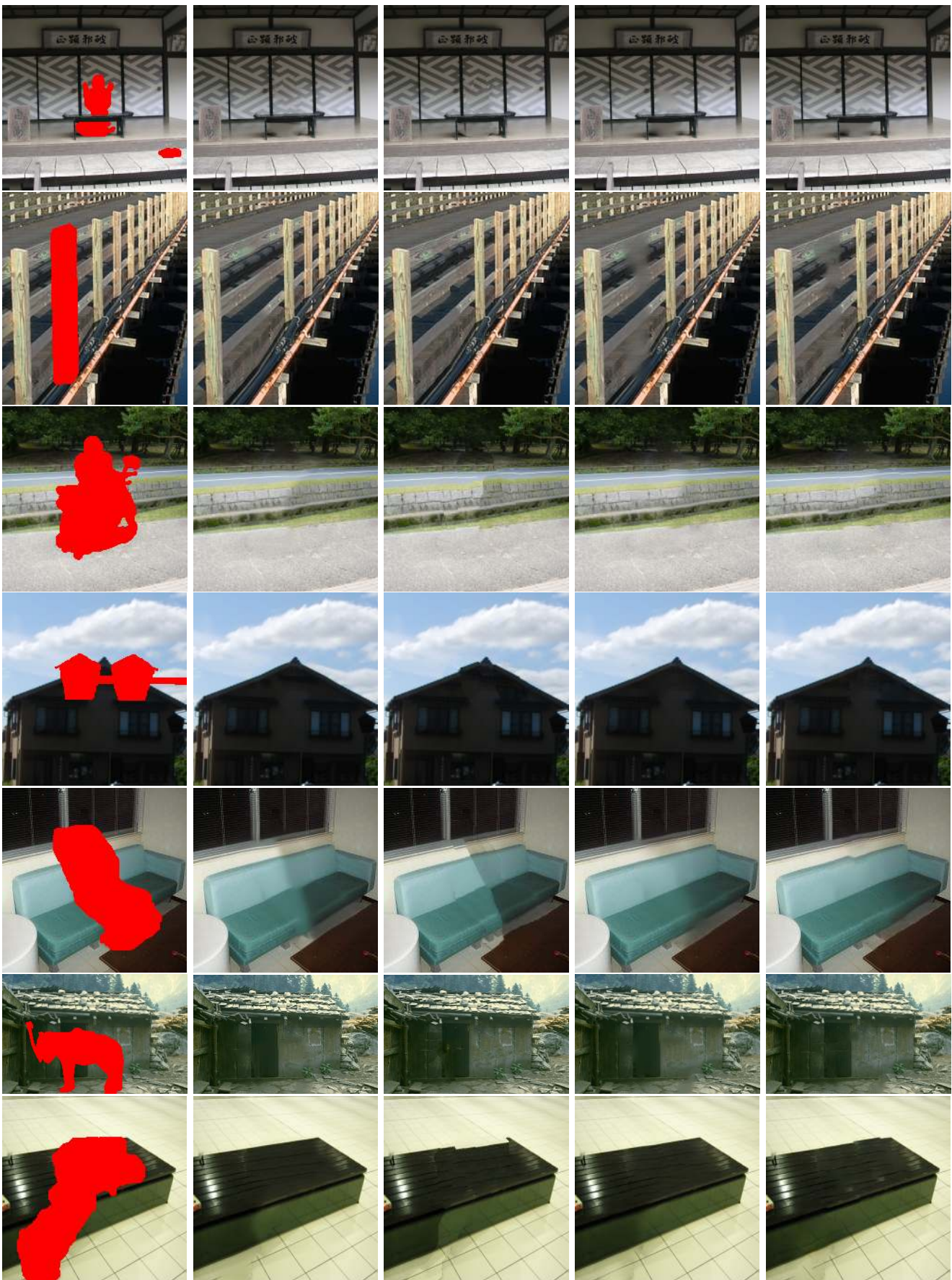
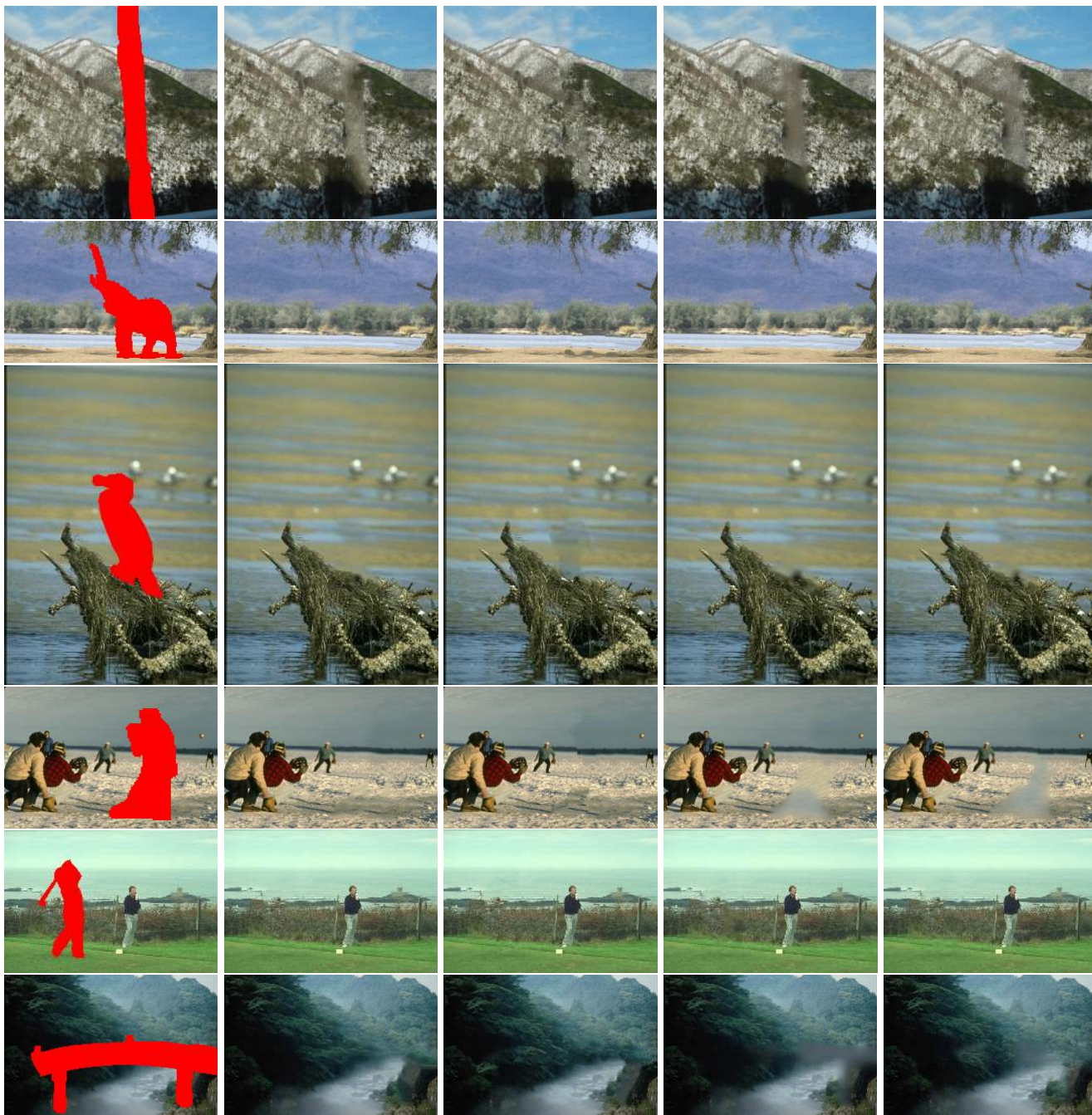


Fig. 9 Results on structured images. From left to right: Original image and mask, patch NL-means, -medians, -Poisson and -GM.

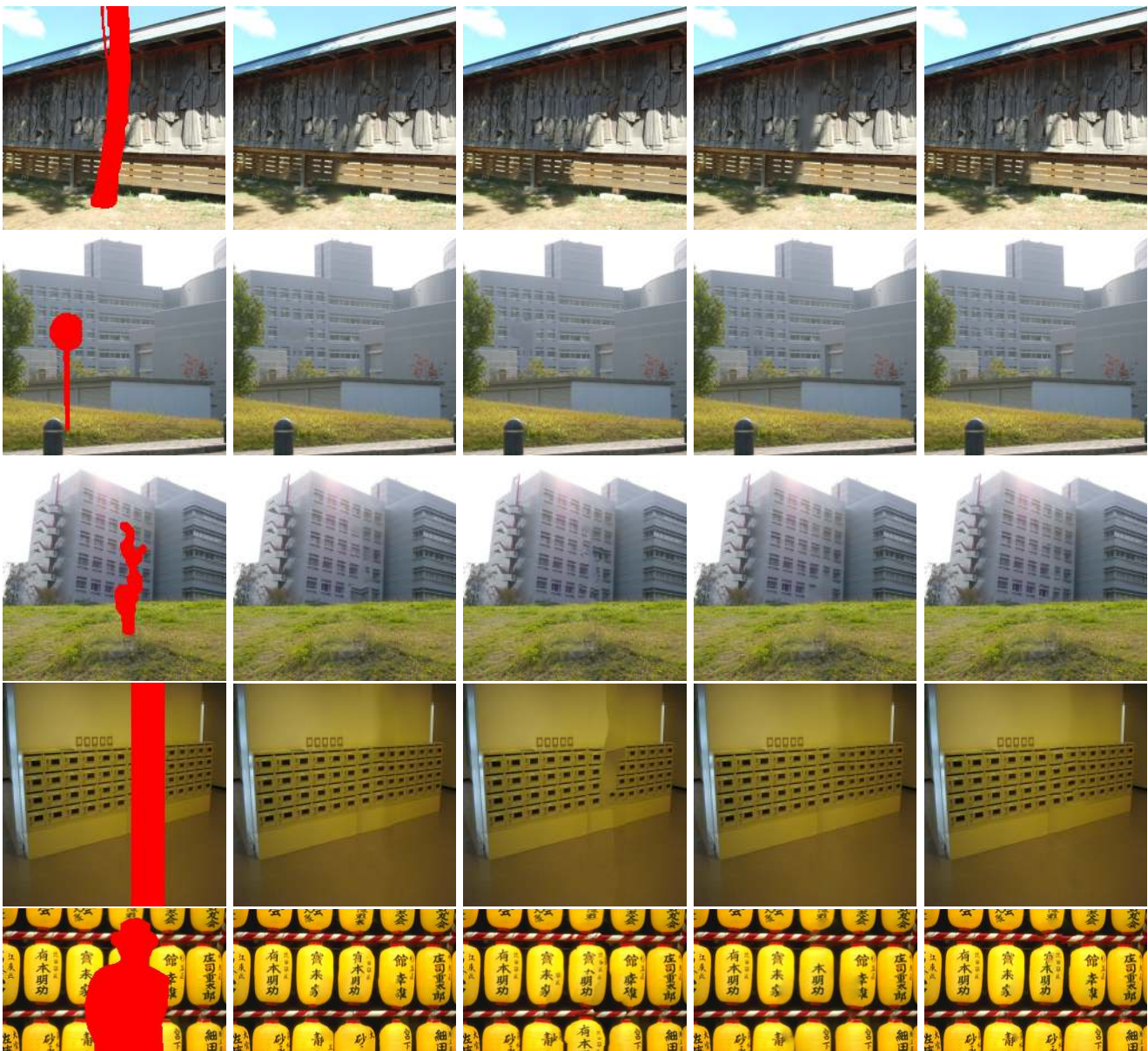


**Fig. 10 Results on random textures.** From left to right: Original image and mask, patch NL-means, -medians, -Poisson and -GM. All images in each row have been generated with the same parameters.

We have observed that  $L^1$  methods are more reluctant to make changes during the minimization process. The same robustness that allows a better performance with textures and avoids the spilling effect makes them more greedy. Once a set of neighboring patches have settled on a locally stable solution (typically a region of constant NNF), it is hard for the algorithm to change that local decision. To understand why, imagine a copy

front advancing from the boundary carrying correct information which meets an already settled copy front, which has taken an undesirable decision based on a bad initialization. Pixels on the boundary of the mistaken front start receiving contributions from patches in the advancing front. Initially these contributions will be outliers in the distribution from which the pixel value is estimated. The median will discard these outliers, and





**Fig. 11 Results on periodic textures.** From left to right: Original image and mask, patch NL-means, -medians, -Poisson and -GM. All images in each row have been generated with the same parameters.

the pixel value will not change unless of course, patches in the advancing front outnumber the mistaken ones.

Although the confidence mask diminishes this effect,  $L^1$ -based methods (specially patch NL-medians) still show more dependence on the initialization. A result of this are some misalignments in straight lines, due to subsampling artifacts of the multiscale scheme. Also patch NL-medians generally requires the use of larger patches, particularly for structured images. This is not always possible, as in row 6 of Fig. 9, where we could not find a proper set of parameters. However we found a good result by after 3 iterations of the multiscale scheme.

The focus of this work lies more on introducing and exploring the variational framework than in presenting a single inpainting algorithm. Still, we would like to comment on which would be the best method among the ones presented. Based on the previous observations, it is clear that the answer to this question will depend on the characteristics of the inpainting problem. However, *a priori*, the patch NL-Poisson seems a reasonable compromise between quality and computational cost. It is also able to provide good results for a wider range of parameters. For cases in which an accurate reproduction of random textures is important, patch NL-GM



**Fig. 12 Results with a non variational algorithm.** The gradient methods sometimes introduce blurry areas in the solution. Here we show the results using a non variational version of the patch NL-Poisson (left column) and patch NL-GM (right column) for some of those images.

may be a better option, however at a higher computational cost.

## 6 Conclusions and future work

In this work we presented a variational framework for exemplar-based non-local image inpainting. The proposed energy lends itself to intuitive interpretations in terms of probabilistic models and has connections with mean shift and statistical mechanics. We minimize it using a coordinate descent algorithm, alternating between similarity weights and the image updates. These processes are coupled through the patch error function, *i.e.* the criterion used to compare patches.

We derived from this framework four different inpainting schemes, corresponding to error functions based on the combinations of  $L^1$ - and  $L^2$ -norms, with image or gradient patches. After testing these schemes on synthetic and real problems we arrived to the following conclusions.

1. The robustness of the  $L^1$ -norm (patch NL-medians and NL-GM) makes it more appropriate for the re-

production of textures. These schemes are however, more greedy, or “stiff,” and therefore more dependent on the initialization. Conversely the squared  $L^2$ -norm introduces some blur, and show less stiffness.

2. When gradients are considered in the patch error function, the image update step is computed as the solution of a (local) diffusion PDE, driven by a non-locally estimated gradient field. This improves the synthesis in terms of continuity at the boundary and attenuation of seams. Also the local interpolation helps in the propagation of structures, such as edges at the boundary of the hole.

The proposed functional shows a critical dependence with the patch size. Furthermore, it is nonconvex and has many local minima, particularly for small patch sizes. To tackle these issues we use a multiscale approach. This is customary in the literature, usually motivated as a way to obtain a good initialization and for computational reasons. We believe however that inpainting is inherently a multiscale problem, and that underlying the multiscale approach lies an inpainting criterion. We are currently working on a variational formulation of multiscale inpainting. Other topic of current research is the use of other patch error functions based on the comparison of structure tensors, which could provide a more robust estimation of the morphological structure of the image.

## A Existence of minima for the functional (2)

Let us consider the model

$$E(u, w) = \frac{1}{h} \int_{\tilde{O}} \int_{\tilde{O}^c} w(x, \hat{x}) \|p_u(x) - p_u(\hat{x})\|^2 d\hat{x} dx + \int_{\tilde{O}} \int_{\tilde{O}^c} w(x, \hat{x}) \log w(x, \hat{x}) d\hat{x} dx. \quad (17)$$

In this section we consider the squared  $L^2$  patch error function

$$\|p_u(x) - p_u(\hat{x})\|^2 = \int_{\Omega_p} g_a(y) (u(x+y) - u(\hat{x}+y))^2 dy$$

where  $g_a$  denotes a smooth integrable kernel whose gradient is also integrable. Let

$$\mathcal{P} := \{w : \tilde{O} \times \tilde{O}^c \rightarrow \mathbb{R} : \int_{\tilde{O}^c} w(x, \hat{x}) d\hat{x} = 1 \text{ a.e. } x \in \tilde{O}\}.$$

Let us consider the admissible class of functions

$$\mathcal{A}_M := \{(u, w) : u = \hat{u} \text{ in } O^c, |u(x)| \leq M \quad w \in \mathcal{P}\}.$$

**Proposition 1** *Assume that  $\hat{u} \in BV(O^c)$ . There exists a solution of the variational problem*

$$\min_{(u, w) \in \mathcal{A}_M} E(u, w). \quad (18)$$

*Proof* Let  $(u_n, w_n)$  be a minimizing sequence of (18), i.e. a sequence such that  $E(u_n, w_n) \rightarrow \inf_{(u,w) \in \mathcal{A}_M} E(u, w)$ . Since  $\Omega$  is a bounded domain we have that

$$\int_{\tilde{O}} \int_{\tilde{O}^c} \chi_{\{w>1\}} w_n(x, \hat{x}) \log w_n(x, \hat{x}) d\hat{x} dx$$

is bounded. Hence  $w_n(1 + \log^+ w_n)$  is bounded in  $L^1$ . Then the sequence  $w_n$  is relatively weakly compact in  $L^1$  and by extracting a subsequence, if necessary, we may assume that  $w_n$  weakly converges in  $L^1(\tilde{O} \times \tilde{O}^c)$  to some  $w \in \mathcal{P}$ . Let us now prove that

$$\nabla_{x+\hat{x}} \int_{\Omega_p} g_a(y) (u(x+y) - u(\hat{x}+y))^2 dy$$

and

$$\nabla_{x-\hat{x}} \int_{\Omega_p} g_a(y) (u(x+y) - u(\hat{x}+y))^2 dy$$

are uniformly bounded when  $|u| \leq M$ . Indeed, the boundedness of the first integral is a consequence of

$$\begin{aligned} & \nabla_{x+\hat{x}} \int_{\Omega_p} g_a(y) (u(x+y) - u(\hat{x}+y))^2 dy \\ &= 2 \int_{\Omega_p} g_a(y) (u(x+y) - u(\hat{x}+y)) \\ & \quad (\nabla_x u(x+y) - \nabla_{\hat{x}} u(\hat{x}+y)) dy \\ &= 2 \int_{\Omega_p} g_a(y) (u(x+y) - u(\hat{x}+y)) \\ & \quad (\nabla_y u(x+y) - \nabla_y u(\hat{x}+y)) dy \\ &= - \int_{\Omega_p} \nabla_y g_a(y) (u(x+y) - u(\hat{x}+y))^2 dy, \end{aligned}$$

the  $L^1$  integrability of  $\nabla_y g_a(y)$  and the boundedness of  $u$ . We are also assuming that  $g_a$  vanishes at the boundary of  $\Omega_p$ . Now,

$$\begin{aligned} & \nabla_{x-\hat{x}} \int_{\Omega_p} g_a(y) (u(x+y) - u(\hat{x}+y))^2 dy \\ &= 2 \int_{\Omega_p} g_a(y) (u(x+y) - u(\hat{x}+y)) \\ & \quad (\nabla_x u(x+y) + \nabla_{\hat{x}} u(\hat{x}+y)) dy \\ &= 2 \int_{\Omega_p} g_a(y) (u(x+y) - u(\hat{x}+y)) \\ & \quad (\nabla_y u(x+y) + \nabla_y u(\hat{x}+y)) dy \\ &= \int_{\Omega_p} g_a(y) (\nabla_y u(x+y))^2 - \nabla_y u(\hat{x}+y)^2 dy \\ &+ 2 \int_{\Omega_p} g_a(y) (u(x+y) \nabla_y u(\hat{x}+y) - u(\hat{x}+y) \nabla_y u(x+y)) dy \end{aligned}$$

Observe that

$$\begin{aligned} & \int_{\Omega_p} g_a(y) (u(x+y) \nabla_y u(\hat{x}+y) - u(\hat{x}+y) \nabla_y u(x+y)) dy \\ &= - \int_{\Omega_p} g_a(y) ((u(\hat{x}+y) + 1) \nabla_y u(x+y) \\ & \quad - u(x+y) \nabla_y u(\hat{x}+y)) dy + \int_{\Omega_p} g_a(y) \nabla_y u(x+y) dy \\ &= - \int_{\Omega_p} g_a(y) \nabla_y \left( \frac{u(x+y)}{(u(\hat{x}+y) + 1)} \right) (u(\hat{x}+y) + 1)^2 dy \\ & \quad - \int_{\Omega_p} \nabla_y g_a(y) u(x+y) dy \\ &= \int_{\Omega_p} \nabla_y (g_a(y) (u(\hat{x}+y) + 1)^2) \frac{u(x+y)}{(u(\hat{x}+y) + 1)} dy \end{aligned}$$

$$\begin{aligned} & - \int_{\Omega_p} \nabla_y g_a(y) u(x+y) dy \\ &= \int_{\Omega_p} \nabla_y g_a(y) (u(\hat{x}+y) + 1) u(x+y) dy \\ &+ 2 \int_{\Omega_p} g_a(y) \nabla_y u(\hat{x}+y) u(x+y) dy - \int_{\Omega_p} \nabla_y g_a(y) u(x+y) dy. \end{aligned}$$

From our assumptions, the boundedness of the previous integral holds. Thus, the functions

$$\nabla_x \int_{\Omega_p} g_a(y) (u(x+y) - u(\hat{x}+y))^2 dy$$

and

$$\nabla_{\hat{x}} \int_{\Omega_p} g_a(y) (u(x+y) - u(\hat{x}+y))^2 dy$$

are uniformly bounded when  $|u| \leq M$ . Thus, modulo the extraction of a subsequence, we may assume that  $u_n \rightarrow u$  weakly in all  $L^p$ ,  $1 \leq p < +\infty$  and  $g_a * (u_n(x+\cdot) - u_n(\hat{x}+\cdot))^2$  converges strongly in all  $L^p$  spaces and also in the dual of  $L \text{Log}^+ L$  to some function  $F$ . Then by passing to the limit as  $n \rightarrow \infty$  we have

$$\begin{aligned} & \frac{1}{h} \int_{\tilde{O}} \int_{\tilde{O}^c} w(x, \hat{x}) F^2 + \int_{\tilde{O}} \int_{\tilde{O}^c} w(x, \hat{x}) \log w(x, \hat{x}) d\hat{x} dx \\ & \leq \liminf_n E(u_n, w_n). \end{aligned}$$

Taking test functions  $\varphi(x, \hat{x})$ , integrating in  $\tilde{O} \times \tilde{O}^c$  and using the convexity of the square function, we have

$$\int_{\Omega_p} g_a(y) (u(x+y) - u(\hat{x}+y))^2 dy \leq F.$$

Then  $E(u, w) \leq \liminf_n E(u_n, w_n)$ .  $\square$

## B Algorithm for solving Equation (13)

Following [2,18] we derive here a fixed point algorithm for a discrete version of (13).

*Discrete formulation.* We consider discrete images defined over a rectangular bounded domain  $\Omega = \{0, \dots, N\}^2$ . We are given an inpainting domain  $O \subset \Omega$  and the known portion of an image  $\hat{u} : O^c \rightarrow \mathbb{R}$ . Let us define  $O_e$  as the set of pixels with at least one neighbor in  $O$  (according to the 4-connectivity). If we think as the discrete image as a lattice of nodes joined by edges, all variable edges are between nodes in  $O_e$ . We will use the notation  $[A \rightarrow B] = \{f : A \rightarrow B\}$ , the set of functions from  $A$  to  $B$ .

We define  $\nabla : [\Omega \rightarrow \mathbb{R}] \rightarrow [\Omega \rightarrow \mathbb{R}^2]$  as

$$\nabla u_{i,j}^1 := \begin{cases} 0 & \text{if } i = N, \\ u_{i+1,j} - u_{i,j} & \text{if } i < N, \end{cases}$$

and similarly for the component  $\nabla u_{i,j}^2$ . The notation  $z = (i, j)$  refers to locations on the image. Let us also define a divergence operator  $\nabla \cdot : [\Omega \rightarrow \mathbb{R}^2] \rightarrow [\Omega \rightarrow \mathbb{R}]$ ,

$$\nabla \cdot p_{i,j} := \begin{cases} p_{i,j}^1 & \text{if } i = 0, \\ -p_{i-1,j}^1 & \text{if } i = N, \\ p_{i,j}^1 - p_{i-1,j}^1 & \text{otherwise,} \end{cases} + \begin{cases} p_{i,j}^2 & \text{if } j = 0, \\ -p_{i,j-1}^2 & \text{if } j = N, \\ p_{i,j}^2 - p_{i,j-1}^2 & \text{otherwise.} \end{cases}$$

These operators incorporate Neumann boundary conditions on the boundary of  $\Omega$ , and are dual operators, i.e. denoting  $\langle \cdot, \cdot \rangle$  the usual scalar products in  $[\Omega \rightarrow \mathbb{R}^2]$  and  $[\Omega \rightarrow \mathbb{R}^2]$ ,  $\langle \nabla u, p \rangle = -\langle u, \nabla \cdot p \rangle$ , for all  $u \in [\Omega \rightarrow \mathbb{R}]$ ,  $p \in [\Omega \rightarrow \mathbb{R}^2]$ .

Let us also define  $U := [O_e \rightarrow \mathbb{R}]$  and  $V := [O_e \rightarrow \mathbb{R}^2]$ . In these spaces we will consider the usual scalar products and

will denote them as  $\langle \cdot, \cdot \rangle_U$  and  $\langle \cdot, \cdot \rangle_V$ . Notice that, with these definitions, the gradient and divergence are not dual operators when restricted to  $O_e$ , i.e. in general  $\langle u, \nabla \cdot p \rangle_U \neq -\langle \nabla u, p \rangle_V$ .

We are looking for a completion  $u \in \tilde{U} := \{u \in U : u_z = \hat{u}_z, \forall z \in O^c\}$  as the solution of the following problem:

$$\min_{u \in \tilde{U}} \sum_{z \in O_e} \sum_{\hat{z} \in D} m_{z\hat{z}} \|\nabla u_z - v_{\hat{z}}\| + \frac{1}{2\delta t} \sum_{z \in O_e} c_z (u_z - f_z)^2, \quad (19)$$

where  $v : D \rightarrow \mathbb{R}^2$ ,  $m : O_e \times D \rightarrow \mathbb{R}^+ \cup \{0\}$ ,  $f : O \rightarrow \mathbb{R}$  and  $c : O_e \rightarrow \mathbb{R}^+$  are given. For the sake of generality we consider a generic domain  $D$  instead of the complement  $O_e^c$ . This generalization is relevant, due for instance to computational considerations as will be discussed shortly. Let us remark that, since  $c_z > 0$  for  $z \in O_e$ , the energy is convex and the minimum is unique.

For simplicity we extend  $f$  over  $O_e$  by defining  $f_z = \hat{u}_z$  for  $z \in O_e \setminus O$ . We use the fact that for any  $\eta \in \mathbb{R}^2$ ,  $\|\eta\| = \sup_{\|\varepsilon\| < 1} \eta \cdot \varepsilon$ , and write the energy as:

$$\min_{u \in \tilde{U}} \sup_{\|\varepsilon_{z\hat{z}}\| \leq 1} \sum_{z \in O_e} \sum_{\hat{z} \in D} m_{z\hat{z}} \varepsilon_{z\hat{z}} \cdot (\nabla u_z - v_{\hat{z}}) + \frac{1}{2\delta t} \sum_{z \in O_e} c_z (u_z - f_z)^2.$$

We have defined the unit field  $\varepsilon \in W := \{\varepsilon : O_e \times D \rightarrow \mathbb{R}^2\}$ .

Interchanging the minimum with the supremum, we obtain the following expression for  $u$  in the inpainting domain

$$u_z^* = f_z + \delta t c_z^{-1} \nabla \cdot \mathcal{L} \varepsilon_z, \quad z \in O. \quad (20)$$

where we have defined the linear operator  $\mathcal{L} : W \rightarrow V$  as  $\mathcal{L} \varepsilon_z = \sum_{\hat{z} \in D} m_{z\hat{z}} \varepsilon_{z\hat{z}}$ .

We can extend expression (20) to  $O_e$ , by defining a new divergence operator  $\bar{\nabla} : V \rightarrow U$  as  $\bar{\nabla} \cdot \varepsilon_z := \chi_z^O \nabla \cdot \varepsilon_z$ . Here  $\chi_z^O = 1$  if  $z \in O$  and  $\chi_z^O = 0$  otherwise. We also define a modified gradient  $\bar{\nabla} : U \rightarrow V$  as the negative adjoint operator of  $\bar{\nabla} \cdot$ , given by  $\bar{\nabla} u_z := \nabla[\chi_z^O u_z]$ . Besides allowing to extend (20) these operators are relevant since, as opposed to the usual gradient and divergence operators, they are still dual when they are restricted to functions over  $O_e$ :  $\langle \bar{\nabla} \cdot p, u \rangle_U = -\langle p, \bar{\nabla} u \rangle_V$ .

Substituting expression (20) in the energy, we arrive to the following dual formulation:

$$\min_{\|\varepsilon_{z\hat{z}}\| \leq 1} - \sum_{z \in O_e} \delta t \left( \mathcal{L} \varepsilon_z \cdot \nabla [c^{-1} \bar{\nabla} \cdot \mathcal{L} \varepsilon]_z + \frac{1}{2} c^{-1} [\bar{\nabla} \cdot \mathcal{L} \varepsilon_z]^2 \right) - \sum_{z \in O_e} \left[ \mathcal{L} \varepsilon_z \cdot \nabla f_z - \sum_{\hat{z} \in D} m_{z\hat{z}} \varepsilon_{z\hat{z}} \cdot v_{\hat{z}} \right].$$

Using that  $\nabla[c^{-1} \bar{\nabla} \cdot \mathcal{L} \varepsilon]_z = \bar{\nabla}[c^{-1} \bar{\nabla} \cdot \mathcal{L} \varepsilon]_z$  for all  $z \in \Omega$ , we can rewrite the dual problem with the more compact expression

$$\min_{\|\varepsilon_{z\hat{z}}\| \leq 1} \sum_{z \in O_e} \left[ \frac{\delta t}{2} c_z^{-1} [\bar{\nabla} \cdot \mathcal{L} \varepsilon_z]^2 - \mathcal{L} \varepsilon_z \cdot \nabla f_z + \sum_{\hat{z} \in D} m_{z\hat{z}} \varepsilon_{z\hat{z}} \cdot v_{\hat{z}} \right]. \quad (21)$$

Adding the Lagrange multipliers corresponding to the restrictions  $\|\varepsilon_{z\hat{z}}\| \leq 1$ ,  $\sum_{z, \hat{z} \in O_e \times D} \lambda_{z\hat{z}} (\|\varepsilon_{z\hat{z}}\|^2 - 1)$ , and deriving with respect to  $\varepsilon$  we get the Euler equation

$$m_{z\hat{z}} [-\nabla f_z + v_{\hat{z}} - \delta t \bar{\nabla}[c^{-1} \bar{\nabla} \cdot \mathcal{L} \varepsilon]_z] + \lambda_{z\hat{z}} \varepsilon_{z\hat{z}} = 0. \quad (22)$$

The *Karush-Kuhn-Tucker* Theorem yields the existence of the Lagrange multipliers with values  $\lambda_{z\hat{z}} = m_{z\hat{z}} \|\nabla f_z + v_{\hat{z}} - \delta t \bar{\nabla}[c^{-1} \bar{\nabla} \cdot \mathcal{L} \varepsilon]_z\|$ . The solution of (21) is computed with the semi-implicit scheme

$$\varepsilon_{z\hat{z}}^{t+1} = \frac{\varepsilon_{z\hat{z}}^t + \nu m_{z\hat{z}} [-\nabla f_z + v_{\hat{z}} - \delta t \bar{\nabla}[c^{-1} \bar{\nabla} \cdot \mathcal{L} \varepsilon^t]_z]}{1 + \nu m_{z\hat{z}} \|\nabla f_z + v_{\hat{z}} - \delta t \bar{\nabla}[c^{-1} \bar{\nabla} \cdot \mathcal{L} \varepsilon^t]_z\|}, \quad (23)$$

where  $\nu$  is a time step small enough for assuring the convergence of the fixed point iteration. Lastly the solution to the primal problem is recovered with (20).

*Convergence.* Let us first observe that  $\nabla f_z - \bar{\nabla} f_z = \nabla[(1 - \chi_z^O) f_z] = \nabla[\chi_z^{O_e \setminus O} f_z]$  for all  $z \in \Omega$ , which allows to express the dual energy (21) as

$$J^*(\varepsilon) = \frac{\delta t}{2} \|c^{-1/2} \bar{\nabla} \cdot \mathcal{L} \varepsilon + c^{1/2} f / \delta t\|_U^2 + \langle h, \varepsilon \rangle_W, \quad (24)$$

where  $h_{z\hat{z}} := m_{z\hat{z}} (v_{\hat{z}} - \nabla[\chi_z^{O_e \setminus O} f_z])$  and  $\langle \cdot, \cdot \rangle_W$  is the usual scalar product in  $W$ . This expression shows that, since  $c_z > 0$ , there is a unique minimum value for  $\bar{\nabla} \cdot \mathcal{L} \varepsilon$ . We prove now that the sequence defined in (23) converges to that minimum.

**Proposition 2** *If  $c_z = \sum_{\hat{z} \in D} m_{z\hat{z}} \leq 1$  for  $z \in O_e$  and the time step  $\nu$  verifies  $\nu < \frac{1}{4\delta t K}$ , with*

$$K := \max_{i,j \in O_e} \left\{ \chi_{i,j}^O + \frac{c_{i,j}}{c_{i+1,j}} \chi_{i+1,j}^O + \frac{c_{i,j}}{c_{i,j+1}} \chi_{i,j+1}^O \right\}, \quad (25)$$

*the sequence  $(\varepsilon^t)$  defined in (23) converges to a minimum of the energy (24).*

*Proof* For reasons of space, we only sketch the main steps of the proof. The derivations are similar to those in [2,18]. Let us first show that if  $\nu < \frac{1}{4\delta t K}$ , the energy  $J^*$  decreases. Defining  $\eta := \nu^{-1}(\varepsilon^{t+1} - \varepsilon^t)$  for a given  $t \geq 0$ , we have that the variation in the energy can be bounded by

$$\begin{aligned} J^*(\varepsilon^{t+1}) &\leq J^*(\varepsilon^t) + \frac{1}{2} \nu (\nu \delta t \|c^{-1/2} \bar{\nabla} \cdot \mathcal{L} \eta\|_U^2 - \|\eta\|_W^2) \\ &\leq J^*(\varepsilon^t) + \frac{1}{2} \|\eta\|_W^2 \nu (4\nu \delta t K - 1), \end{aligned} \quad (26)$$

where  $K$  is a constant such that  $4K \|\eta\|_W^2 \geq \|c^{-1/2} \bar{\nabla} \cdot \mathcal{L} \eta\|_U^2$  for all  $\eta \in W$ . We will derive this constant later.

Therefore, by choosing  $\nu < (4\delta t K)^{-1}$  the energy decreases with  $\varepsilon^t$ . Let  $m = \lim_{t \rightarrow \infty} J^*(\varepsilon^t) \geq 0$ . Let us prove now that  $\varepsilon^t$  also converges. Since  $\|\varepsilon_{z\hat{z}}^t\| \leq 1$  for all  $z \in O_e$ ,  $\hat{z} \in D$ ,  $t \geq 0$ , there exists a subsequence  $(\varepsilon^{t_k})$  converging to  $\bar{\varepsilon}$ . From (23) we see that  $(\varepsilon^{t_k+1})$  converges as well. Let  $\bar{\varepsilon}' = \lim_{k \rightarrow \infty} \varepsilon^{t_k+1}$ . Defining  $\bar{\eta} := (\bar{\varepsilon}' - \bar{\varepsilon})/\nu$ , we see that in the limit when  $t_k \rightarrow \infty$   $J^*(\bar{\varepsilon}') \leq J^*(\bar{\varepsilon}) + \frac{1}{2} \|\bar{\eta}\|_W^2 \nu (4\nu \delta t K - 1)$ . Since  $J^*(\bar{\varepsilon}') = J^*(\bar{\varepsilon}) = m$  we can conclude that  $\bar{\eta} = 0$  and therefore  $\bar{\varepsilon} = \bar{\varepsilon}'$ . This implies that the whole sequence  $(\varepsilon^t)$  converges. Furthermore the limit  $\bar{\varepsilon}$  verifies the Euler equation (22), and thus is a minimizer of the dual energy  $J^*$ .

We now sketch the derivation of the expression for  $K$ . Let us define the following  $c$ -weighted norms:  $\|u\|_c^2 := \langle c^{-1} u, u \rangle_U$  for  $u \in U$ ,  $\|\xi\|_c^2 := \langle c^{-1} \xi, \xi \rangle_V$  for  $\xi \in V$  and  $\|A\|_{cc}^2 := \sup_{\xi \in V} \left\{ \frac{\|A\xi\|_c^2}{\|\xi\|_c^2} \right\}$  for operators  $A : V \rightarrow U$ . For any  $\eta \in W$  we can write

$$\|c^{-1/2} \bar{\nabla} \cdot \mathcal{L} \eta\|_U^2 = \|\bar{\nabla} \cdot \mathcal{L} \eta\|_c^2 \leq \|\bar{\nabla} \cdot\|_{cc}^2 \|\mathcal{L} \eta\|_c^2.$$

On one hand we have that

$$\|\mathcal{L} \eta\|_c^2 \leq \|\eta\|_W^2 \sup_{z \in O_e} \left\{ c_z^{-1} \sum_{\hat{z} \in D} m_{z\hat{z}}^2 \right\} \leq \|\eta\|_W^2,$$

where first inequality is an application of *Cauchy-Schwarz*, then we have used that  $\sum_i a_i^2 \leq (\sum_i |a_i|)^2$ , and the fact that  $c_z = \sum_{\hat{z} \in D} m_{z\hat{z}} \leq 1$ .

On the other hand, from the definition of the divergence operator, it can be shown that  $\|\bar{\nabla} \cdot\|_{cc}^2 \leq 4K$ .  $\square$



*Computational considerations.* Let us remark that for the case of inpainting ( $D = O_e^c$ ) the domain of the dual variable  $\varepsilon$  is  $O_e \times O_e^c \subset \mathbb{R}^4$ , and therefore the direct implementation of this algorithm is prohibitive (for large images with large inpainting domains  $\varepsilon$  will not fit in memory). To circumvent this problem we threshold the pixel-wise influence weights  $m_{z\hat{z}}$ , so as to keep for each  $z$  only the  $M$  highest contributions. This reduces the size of  $\varepsilon$  to  $M|O_e|$ . Indeed, when  $h \rightarrow 0$  the  $m$  function is already sparse w.r.t.  $\hat{z}$ , being the number of non-zero elements less or equal than the patch size (in pixels). In our experiments we set  $M$  to the size of the patch, capturing exactly the function  $m$ .

## C Computation of the Nearest Neighbor Field

The computation of the nearest neighbor (or of the weight function  $w$ ) is the most time consuming step of an exemplar-based algorithm. Two key observations allow to reduce the computational load for this task. The first one is that, the nearest neighbor search can be approximated without compromising the quality of the output, allowing to trade precision for speed (see for instance [13] and references therein). The second one, as noted in [7], is that due to the coherence of natural images, nearest neighbors of patches centered on nearby pixels are likely to be located at nearby positions.

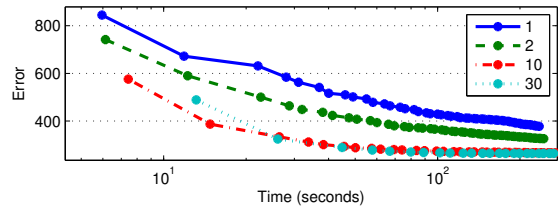
PatchMatch is a very efficient algorithm for computing approximate nearest neighbors proposed in [7]. The authors in [7] consider two regions  $O$  and  $O^c$  of the same or different images. The objective is to find for each patch centered in  $x \in O$  the location of the best match  $n(x) \in O^c$ . The authors define the Nearest Neighbor Field (NNF) as the function  $x \mapsto (n(x) - x)$  defined over  $O$ . Instead of searching for the nearest neighbor of each query patch independently, PatchMatch computes the NNF simultaneously for all query patches. It exploits the fact that since query patches overlap, the offset  $n(x) - x$  of a good match for at  $x$  is likely to lead to a good match for the adjacent points of  $x$  as well. It is an iterative algorithm which starting from a random initialization, alternates between steps of propagation of good offsets and random search (see [7] for details). For most applications with natural images, a few iterations after a random initialization are often sufficient.

We extended this algorithm following a suggestion in [7]. Instead of storing a single offset for each query patch, we store queues of  $L$  offsets in an  $L$ -Nearest Neighbors Field ( $L$ -NNF). Thus the number of initial random guesses gets multiplied by  $L$ . During the propagation step adjacent queues are merged in to a temporal queue of length  $2L$  from which only the best  $L$  offsets are kept. This allows to preserve and transfer non optimal offsets that may in turn be optimal for farther positions. Our implementation of the random search is essentially as in [7].

The use of queues represents a significant improvement of performance with respect to the original PatchMatch. Figure 13 shows the evolution of the nearest neighbor's error measured as  $\sum_{x \in O} \|p_u(x) - p_u(n(x))\|^2$  with time, using queues of different lengths. Even  $L = 2$  allows to achieve results of similar quality in less time. Clearly  $L$  cannot grow unbounded: with  $L = 30$  the first iteration got to a higher precision but it took more than 10 seconds. In practice we set  $L \leq 10$  and perform a small number of iterations.

Let us remark that each queue is an approximated neighborhood of the query patch in the patch space. This neighborhood can be interpreted as a truncated representation of the function  $w(x, \cdot)$  for each pixel  $x$ , which is useful in the non-degenerated case when  $h > 0$  in (6).

There is an interesting parallelism between this algorithm and the mechanic of Genetic Algorithms. The  $L$ -NNF problem



**Fig. 13 Performance of the modified PatchMatch.** The graph shows the evolution of the nearest neighbor's error with the iterations of the algorithm. Each graph corresponds to a different size  $L$  of the sets ( $L = 1$  corresponds to the original PatchMatch [7]). The dots indicate the completion of an iteration.

can be seen as the simultaneous minimization of many fitness functions, one for each pixel in  $O$ . Each fitness function is defined as the sum of the patch errors over the corresponding queue. If we identify the *individuals* with the queues for each pixel in  $O$ , then the *population* for single problem is composed by the queue at the pixel and its immediate neighbors (5 individuals). The random search can be interpreted as *mutations*, the propagation as an optimal *crossover* between individuals (queues). The population overlap of neighboring problems responds to the fact that adjacent problems are likely to have similar solutions.

## Acknowledgments

The authors thank Pau Gargallo for his insightful comments.

PA is supported by the FPI grant BES-2007-14451 from the Spanish MICINN. PA, GF, and VC acknowledge partial support by MICINN project, reference MTM2009-08171, and by GRC reference 2009 SGR 773. VC also acknowledges partial support by IP project "2020 3D Media: Spacial Sound and Vision", financed by EC, and by "ICREA Acadèmia" prize for excellence in research funded both by the Generalitat de Catalunya. GS is partially supported by NSF, ONR, DARPA, NGA, and ARO.

## References

- Aharon, M., Elad, M., Bruckstein, A.: The K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representation. *IEEE Trans. on Signal Processing* **54**(11), 4311–22 (2006)
- Almansi, A., Caselles, V., Haro, G., Rougé, B.: Restoration and zoom of irregularly sampled, blurred, and noisy images by accurate total variation minimization with local constraints. *Multiscale Modeling & Simulation* **5**(1), 235–272 (2006)
- Arias, P., Caselles, V., Sapiro, G.: A variational framework for non-local image inpainting. In: *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 345–358. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
- Aujol, J.F., Ladjal, S., Masnou, S.: Exemplar-based inpainting from a variational point of view. Submitted (2008)
- Awate, S., Whitaker, R.: Unsupervised, information-theoretic, adaptive image filtering for image restoration. *IEEE Trans. on PAMI* **28**(3), 364–376 (2006)
- Ballester, C., Bertalmío, M., Caselles, V., Sapiro, G., Verdera, J.: Filling-in by joint interpolation of vector fields and gray levels. *IEEE Trans. on IP* **10**(8), 1200–11 (2001)

7. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: Patchmatch: a randomized correspondence algorithm for structural image editing. In: Proc. of SIGGRAPH, pp. 1–11. ACM, New York, NY, USA (2009)
8. Bertalmío, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proc. of SIGGRAPH (2000)
9. Bertalmío, M., Vese, L., Sapiro, G., Osher, S.: Simultaneous structure and texture inpainting. *IEEE Trans. on Image Processing* **12**(8), 882–89 (2003)
10. Bornard, R., Lecan, E., Laborelli, L., Chenot, J.H.: Missing data correction in still images and image sequences. In: Proc. ACM Int. Conf. on Multimedia (2002)
11. Bornemann, F., März, T.: Fast image inpainting based on coherence transport. *J. of Math. Imag. and Vis.* **28**(3), 259–78 (2007)
12. Brox, T., Cremers, D.: Iterated non-local means for texture restoration. pp. 13–24. Springer, Germany (2007)
13. Brox, T., Kleinschmidt, O., Cremers, D.: Efficient nonlocal means for denoising of textural patterns. *IEEE Trans. on IP* **17**(7), 1057–92 (2008)
14. Buades, A., Coll, B., Morel, J.: A non local algorithm for image denoising. In: Proc. of the IEEE Conf. on CVPR, vol. 2, pp. 60–65 (2005)
15. Buades, A., Coll, B., Morel, J.M., Sbert, C.: Self-similarity driven color demosaicking. *Image Processing, IEEE Transactions on* **18**(6), 1192–1202 (2009)
16. Candes, E.J., Wakin, M.B.: An introduction to compressive sampling. *Signal Processing Magazine, IEEE* **25**(2), 21–30 (2008)
17. Cao, F., Gousseau, Y., Masnou, S., Pérez, P.: Geometrically-guided exemplar-based inpainting. Submitted (2008)
18. Chambolle, A.: An algorithm for total variation minimization and applications. *J. Math. Imaging Vis.* **20**(1-2), 89–97 (2004)
19. Chan, T., Kang, S.H., Shen, J.H.: Euler’s elastica and curvature based inpaintings. *SIAM J. App. Math.* **63**(2), 564–92 (2002)
20. Chan, T., Shen, J.H.: Mathematical models for local nontexture inpaintings. *SIAM J. App. Math.* **62**(3), 1019–43 (2001)
21. Cheng, Y.: Mean shift, mode seeking and clustering. *IEEE Trans. on PAMI* **17**(8), 790–99 (1995)
22. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based inpainting. *IEEE Trans. on IP* **13**(9), 1200–1212 (2004)
23. Csiszár, I.: Axiomatic characterizations of information measures. *Entropy* **10**(3), 261–73 (2008)
24. Demanet, L., Song, B., Chan, T.: Image inpainting by correspondence maps: a deterministic approach. Tech. rep., UCLA (2003)
25. Drori, I., Cohen-Or, D., Yeshurun, H.: Fragment-based image completion. *ACM Trans. on Graphics. Special issue: Proc. of ACM SIGGRAPH* **22**(3), 303–12 (2003)
26. Efros, A., Leung, T.: Texture synthesis by non-parametric sampling. In: Proc. of the IEEE Conf. on CVPR, pp. 1033–1038 (1999)
27. Elad, M., Starck, J., Querre, P., Donoho, D.: Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Applied and Computational Harmonic Analysis* **19**(3), 340–358 (2005)
28. Esedoglu, S., Shen, J.H.: Digital image inpainting by the Mumford-Shah-Euler image model. *European J. App. Math.* **13**, 353–70 (2002)
29. Facciolo, G., Arias, P., Caselles, V., Sapiro, G.: Exemplar-based interpolation of sparsely sampled images. In: *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pp. 331–344. Springer Berlin Heidelberg (2009)
30. Fang, C.W., Lien, J.J.J.: Rapid image completion system using multiresolution patch-based directional and nondirectional approaches. *Trans. Img. Proc.* **18**(12), 2769–2779 (2009)
31. Gilboa, G., Osher, S.: Nonlocal linear image regularization and supervised segmentation. *SIAM Mult. Mod. and Sim.* **6**(2), 595–630 (2007)
32. Gilboa, G., Osher, S.: Nonlocal operators with application to image processing. Tech. rep., UCLA (2007)
33. Hays, J., Efros, A.: Scene completion using millions of photographs (2008)
34. Holtzman-Gazit, M., Yavneh, I.: A scale-consistent approach to image completion. *Int. J. Multiscale Comput. Eng.* **6**(6), 617–628 (2008)
35. Igehy, H., Pereira, L.: Image replacement through texture synthesis. In: Proc. of the IEEE Conf. on CVPR (1997)
36. Jaynes, E.T.: Information theory and statistical mechanics. *Physical Review* **106**(4), 620–30 (1957)
37. Jia, J., Tang, C.K.: Inference of segmented color and texture description by tensor voting. *IEEE Trans. on PAMI* **26**(6), 771–86 (2004)
38. Kawai, N., Sato, T., Yokoya, N.: Image inpainting considering brightness change and spatial locality of textures and its evaluation. In: T. Wada, F. Huang, S. Lin (eds.) *Advances in Image and Video Technology, Lecture Notes in Computer Science*, vol. 5414, chap. 24, pp. 271–282. Springer Berlin Heidelberg, Berlin, Heidelberg (2009)
39. Komodakis, N., Tziritas, G.: Image completion using efficient belief propagation via priority scheduling and dynamic pruning. *IEEE Trans. on IP* **16**(11), 2649–61 (2007)
40. Levin, A., Zomet, A., Weiss, Y.: Learning how to inpaint from global image statistics. In: Proc. of ICCV (2003)
41. Levina, E., Bickel, P.: Texture synthesis and non-parametric resampling of random fields. *Annals of Statistics* **34**(4) (2006)
42. Lezoray, O., Elmoataz, A., Boughleux, S.: Graph regularization for color image processing. *Comput. Vis. Image Underst.* **107**(1-2), 38–55 (2007)
43. Mairal, J., Sapiro, G., Elad, M.: Learning multiscale sparse representations for image and video restoration. *SIAM MMS* **7**(1), 214–241 (2008)
44. Masnou, S.: Disocclusion: a variational approach using level lines. *IEEE Trans. on IP* **11**(2), 68–76 (2002)
45. Masnou, S., Morel, J.M.: Level lines based disocclusion. In: Proc. of IEEE ICIP (1998)
46. Morel, J.M., Yu, G.: On the consistency of the SIFT Method. Preprint, CMLA **26** (2008)
47. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. In: Proc. of SIGGRAPH (2003)
48. Pérez, P., Gangnet, M., Blake, A.: PatchWorks: Example-based region tiling for image editing. Tech. rep., Microsoft Research (2004)
49. Peyré, G.: Manifold models for signals and images. *Comp. Vis. and Im. Unders.* **113**(2), 249–260 (2009)
50. Peyré, G., Boughleux, S., Cohen, L.: Non-local regularization of inverse problems. In: *ECCV '08*, pp. 57–68. Springer-Verlag, Berlin, Heidelberg (2008)
51. Peyré, G., Boughleux, S., Cohen, L.D.: Non-local regularization of inverse problems. Preprint Hal-00419791 (2009)
52. Pizarro, L., Mrázek, P., Didas, S., Grewenig, S., Weickert, J.: Generalised nonlocal image smoothing. Technical Report No. 248, Department of Mathematics, Saarland University, Saarbrücken, Germany (2009)
53. Protter, M., Elad, M., Takeda, H., Milanfar, P.: Generalizing the non-local-means to super-resolution reconstruction. *IEEE Trans. on IP* **18**(1), 36–51 (2009)



- 
54. Tschumperlé, D., Deriche, R.: Vector-valued image regularization with PDE's: a common framework for different applications. *IEEE Trans. on PAMI* **27**(4) (2005)
  55. Wei, L.Y., Levoy, M.: Fast texture synthesis using tree-structured vector quantization. In: *Proc. of the SIGGRAPH* (2000)
  56. Wexler, Y., Shechtman, E., Irani, M.: Space-time completion of video. *IEEE Trans. on PAMI* **29**(3), 463–476 (2007)
  57. Zhou, D., Schölkopf, B.: Regularization on discrete spaces. In: *Proceedings of the 27th DAGM Symposium*, pp. 361–368. Springer (2005)