A Comprehensive Framework for Image Inpainting

Aurélie Bugeau, Marcelo Bertalmío, Vicent Caselles, *Member, IEEE*, and Guillermo Sapiro

***Abstract*—Inpainting is the art of modifying an image in a form that is not detectable by an ordinary observer. There are numerous and very different approaches to tackle the inpainting problem, though as explained in this paper, the most successful algorithms are based upon one or two of the following three basic techniques: copy-and-paste texture synthesis, geometric partial differential equations (PDEs), and coherence among neighboring pixels. We combine these three building blocks in a variational model, and provide a working algorithm for image inpainting trying to approximate the minimum of the proposed energy functional. Our experiments show that the combination of all three terms of the proposed energy works better than taking each term separately, and the results obtained are within the state-of-the-art.**

***Index Terms*—Image inpainting, partial differential equations (PDEs), texture synthesis, variational models.**

I. Introduction

INPAINTING is the art of modifying an image or video in a form that is not (easily) detectable by an ordinary observer, and has become a fundamental area of research in image processing. Formally we could express the inpainting problem in this way: given an image u with a masked region Ω, fill-in each pixel inside Ω with a value (or combination of values) taken from Ω^c. This interest in image inpainting is born in part in the great practical importance of restoring and modifying images and videos, but is also a result of the value of using image inpainting to understand the validity of different image models. Considering image models then, image inpainting techniques can be divided in three basic groups.

One such group considers the self-similarity principle, meaning that an image has a lot of repetitions of local information. In the seminal paper [1], the texture is modeled assuming that the probability distribution of brightness values for one pixel given the brightness values of its spatial neighborhood is independent from the rest of the image. The neighborhood is a square window (patch) around the pixel and its size is a global parameter of the algorithm. The input of the algorithm is a set of model image patches (all patches entirely belonging to Ω^c) and the task is to select an appropriate patch by computing a difference between patches for each of the unknown pixels (pixels in Ω), so as to predict its value. See Fig. 1. This approach is a one-pass greedy algorithm: once a pixel is filled-in, its value remains unchanged. It then suffers from common problems with greedy algorithms, being the filling order (as well as the patch size) very critical. While staying in the greedy one-pass arena, the authors of [2] designed a clever ordering procedure with priorities depending upon edge strength, further improving on [1].

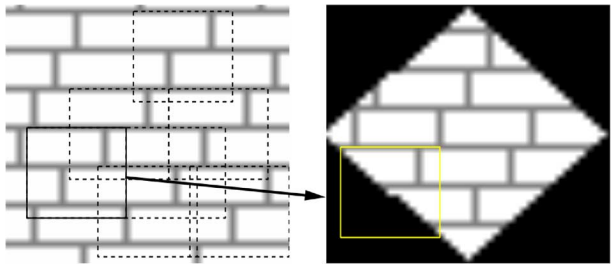


Fig. 1. Efros and Leung's algorithm overview (figure taken from [1]). Given a sample texture image (left), a new image is being synthesized one pixel at a time (right). To synthesize a pixel, the algorithm first finds all neighborhoods in the sample image (boxes on the left) that are similar to the pixels neighborhood (box on the right) and then randomly chooses one neighborhood and takes its center to be the newly synthesized pixel.

The texture synthesis problem as just described is akin to finding the *correspondence map* [3], $\varphi : \mathcal{P} \to \Omega^c$, that associates each pixel of the image to an original pixel such that

$$\varphi(p) = \begin{cases} p & \text{if } p \in \Omega^c \\ q \in \Omega^c & \text{if } p \in \Omega \end{cases} \tag{1}$$

where q is a pixel such that the difference between the patches centered at q and p is minimal. We have denoted $\mathcal{P} = \Omega \cup \Omega^c$ as the full image domain.

Following this line of research of copying patches to the regions to be filled-in, a number of authors proposed to add spatial (and temporal for video) coherence in the texture synthesis process, giving rise to another group of inpainting techniques. The first such work was by Ashikhmin [4], who proposed to perform exemplar-based texture synthesis looking for the best match for pixel $p \in \Omega$ not in all Ω^c but only among the set $\{\varphi(p+k)-k\}$ of shifted candidates from the correspondents of the neighbors of p. In practice, it imposes a certain *coherence* in the mapping function φ which clearly improves the visual quality of the synthesis results. This idea was later formalized

Manuscript received March 01, 2010; revised April 05, 2010; accepted April 05, 2010. Date of publication April 29, 2010; date of current version September 17, 2010. This work was developed for Mediapro inside the i3Media project which is supported in part by the Centro para el Desarollo Tecnológico Industrial (CDTI), within the Ingenio 2010 initiative. The work of A. Bugeau was supported by the Torres Quevedo program of the Ministerio de Educación y Ciencia in Spain. The work of M. Bertalmío and V. Caselles was supported in part by the PNPGC project, reference MTM2006-14836. V. Caselles was also supported in part by "ICREA Acadèmia" for excellence in research funded by the Generalitat de Catalunya. The work of G. Sapiro was supported in part by NSF, NGA, ONR, ARO, and DARPA. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Eric Kolaczyk.

A. Bugeau is with Barcelona Media—Centre d'Innovació, 08018 Barcelona, Spain (e-mail: aurelie.bugeau@barcelonamedia.org).

M. Bertalmío and V. Caselles are with Departament de Tecnologies de la Informació i les Comunicacions, Universitat Pompeu-Fabra, 08002 Barcelona, Spain (e-mail: marcelo.bertalmio@upf.edu; vicent.caselles@upf.edu).

G. Sapiro is with Department of Electrical and Computer Engineering University of Minnesota, Minneapolis, MN 55455 USA (e-mail: guille@ece.umn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2010.2049240

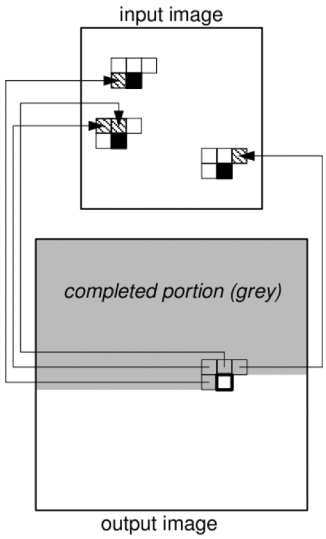


Fig. 2. Ashikhmin's texture synthesis method imposes coherence in the mapping function (figure taken from [4]). Each pixel in the current L-shaped neighborhood generates a shifted candidate pixel (black) according to its original position (hatched) in the input texture. The best pixel is chosen among these candidates only. Several different pixels in the current neighborhood can generate the same candidate.

a bit more and the name "coherence" introduced in this context in [5]. In [6], both spatial and temporal coherence for video inpainting are imposed via a variational formulation globally optimized for. Coherence appears also in the inpainting formulation of [7] by favoring the similarity of the overlapping region of patches corresponding to neighboring pixels. There are also a number of works (see [8]–[11] and references therein) which perform inpainting in a context of sparse representation: patches inside Ω are synthesized as a sparse linear combination of elements from an image dictionary. The sparsity of the linear combination, the size of the dictionary and the use of overlapping patches favor mapping functions φ which are *coherent* (see Fig. 2).

A third group of inpainting algorithms is based upon diffusion with partial differential equations (PDEs) and variational formulations, which have been very successfully used, in particular for piecewise smooth images or when the gap Ω is thinner than the surrounding objects. The first work in this area was by Masnou and Morel [12], [13], whose algorithm performs inpainting by joining with geodesic curves the points of the level lines (iso-value contours) arriving at the boundary of Ω (see Fig. 3). They were inspired by the work of Nitzberg *et al.* [14] who, in the context of image segmentation, were looking for completion curves in a missing region Ω by minimizing an energy functional called Euler's elastica. Many other functionals for inpainting have been proposed, as well as methods that directly introduce a PDE to perform inpainting, although these PDEs do not minimize any known functional and therefore these methods are not variational, [15]–[18]. The underlying idea of these type of approaches is to smoothly transport the contours of the image into the region being inpainted. They are thereby very efficient at smoothly inpainting boundaries, but for the same reason fail in textured parts of the image.

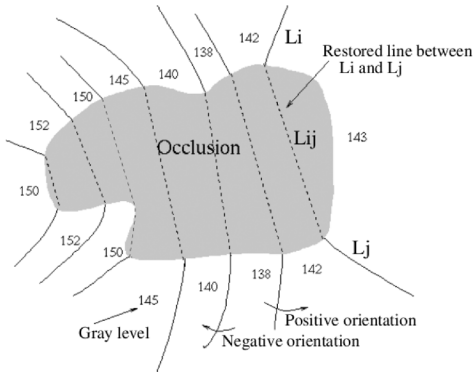


Fig. 3. Masnou and Morel's disocclusion algorithm (figure taken from [12]). An occlusion (the mask Ω in our context) and a possible connection of level lines two by two.

Pairwise combinations of the previously mentioned basic building blocks are common in the literature (e.g., copy-and-paste texture synthesis with coherence [4], [7], copy-and-paste texture synthesis with diffusion PDEs [2], [19], PDEs and variational formulations with a setting on patches [20], [21], etc.). Without intending to build an exhaustive taxonomy of all the inpainting algorithms that have been proposed in the literature, we have found that all the best of these algorithms are based upon one or two, but not all three, of the aforementioned blocks. Our formulation goes one step further and combines all three models.

In this paper, we propose an inpainting algorithm based upon the three main ideas discussed previously. The first term is a variational formulation of the texture synthesis method of Efros and Leung [1], also introduced in [3], [22]. The second and third terms combine both ideas of diffusion and coherence applied in the patch space. We have found only two works in the literature with related ideas.

1) Demanet *et al.* [3] perform iterative texture synthesis, and suggest as a possible improvement to impose restrictions on the mapping function by penalizing its Total Variation and favoring it to behave locally as a translation, but the authors did not later pursue this line of thought.
2) Aujol *et al.* [22] proposed and studied several energy functionals for inpainting with terms related to texture synthesis, coherence (by favoring that the mapping function is locally a roto-translation), and geometry (by minimizing the Total Variation of the "geometry" part of the image, after a cartoon+texture decomposition), but the authors did not actually implement a minimization algorithm for them.

Therefore, to the best of our knowledge, the approach introduced in this paper is novel. We propose a working algorithm for image inpainting trying to approximate the global minimum of an energy functional that combines the three fundamental concepts of self-similarity, coherence, and propagation. This would be a *comprehensive* approach in the sense that, as we have pointed in this section, in the extensive literature of the problem all successful algorithms deal with only one or two of the aforementioned fundamental concepts.

Let us describe the plan of the paper. First, in Section II, we explain how the three types of inpainting methods are combined

into one energy and we describe the main structure of our algorithmic approach. In Section III, we give further details of the implementation. Finally, the experimental results are presented in Section IV.

II. Combining Existing Approaches Into One Energy Function

Throughout the paper $u : \mathcal{P} \to \mathbb{R}^N$ ($N = 1$ for gray level images, $N = 3$ for color images) denotes the given image, where $\mathcal{P} = \Omega \cup \Omega^c$ is the image domain, Ω being the hole or region with missing data and Ω^c the known part of the image. In practice, gray level images or each color channel take integer values in $\{0, \ldots, 255\}$.

The proposed energy has the form

$$\mathcal{E}(\varphi) := \alpha_1 \mathcal{E}_1(\varphi) + \alpha_2 \mathcal{E}_2(\varphi) + \alpha_3 \mathcal{E}_3(\varphi)$$

where φ is the correspondence map, $\alpha_1, \alpha_2, \alpha_3 \geq 0$ represent weights, $\mathcal{E}_1(\varphi)$ is the energy corresponding to texture synthesis (self-similarity), $\mathcal{E}_2(\varphi)$ is the term corresponding to propagation/diffusion (with some coherence), and $\mathcal{E}_3(\varphi)$ reinforces coherence. Working with patches, we use the notation Ψ_s to denote a squared patch domain of size $(2s+1) \times (2s+1)$ centered at the pixel coordinates (0, 0), $\Psi_s(p) = p + \Psi_s$ to denote a $(2s+1) \times (2s+1)$ patch centered at pixel p, and $P(p) = \{u(q) : q \in \Psi_s(p)\}$ to denote the image restricted to the patch $\Psi_s(p)$. Then we express the energy in the following way:

$$\mathcal{E}(\varphi) = \sum_{p \in \Omega} d\left(p, \varphi(p), \varphi|_{\Psi_s(p)\setminus\{p\}}\right) \tag{2}$$

where

$$d\left(p, \varphi(p), \varphi|_{\Psi_s(p)\setminus\{p\}}\right) := \sum_{i=1,2,3} \alpha_i d_i(p, \varphi(p), \varphi|_{\Psi_s(p)\setminus\{p\}}) \tag{3}$$

is the contribution of the pixel p to the energy (sum of the texture synthesis, diffusion, and coherence terms) which depends upon $p, \varphi(p)$ and on φ restricted to the rest of the patch centered at $p(\varphi|_{\Psi_s(p)\setminus\{p\}})$. The weights α_i and the patch difference measures d_i will be explained later.

The solution to the inpainting problem would be given by the map φ minimizing (2). Due to the difficulties in minimizing such a general form of the energy, we propose to proceed in an iterative way, minimizing at each iteration n the relaxed energy

$$\mathcal{E}(\varphi, \varphi^{n-1}) = \sum_{p \in \Omega} d\left(p, \varphi(p), \varphi^{n-1}|_{\Psi_s(p)\setminus\{p\}}\right) \tag{4}$$

where φ^{n-1} is the current estimate of φ.

Then we may use that

$$\min_{\varphi} \mathcal{E}(\varphi, \varphi^{n-1}) = \sum_{p \in \Omega} \min_{\varphi} d\left(p, \varphi(p), \varphi^{n-1}|_{\Psi_s(p)\setminus\{p\}}\right) \tag{5}$$

which permits to devise an effective search algorithm (the one we use in practice) although this may not correspond to the actual global minimum or to a stationary point of (2). Indeed, on one hand the (2) is not convex and there is clear numerical strategy which guarantees that one is able to reach its global minimum. On the other hand there is no proof that the iterations given by the sequence of problems (5) converges to a minimum of (2) be either local or global. Finally, let us observe that the iterative scheme given by (5) does not guarantee that boundary conditions are necessarily satisfied. This can be easily seen by considering the functional $F(\varphi) = \sum_{i=1}^{N} |\varphi(i) - \varphi(i-1)|^2$ with the boundary conditions $\varphi(0) = 0$ and $\varphi(N) = N$. The minimum of F is $\varphi(i) = i$. If we replace F by $F(\varphi, \varphi^{n-1}) = \sum_{i=1}^{N} |\varphi(i) - \varphi^{n-1}(i-1)|^2$ and we take $\varphi^{n-1}(i) = i$ (the actual minimum of F), then $\min_{\varphi} F(\varphi, \varphi^{n-1})$ would produce $\varphi(i) = i - 1$ for any $i = 1, \ldots, N-1$.

In spite of these comments we shall adopt the scheme given by (5) since, at each iteration n and for each pixel $p \in \Omega$, we find its best candidate matching pixel $\varphi(p)$ as the pixel $q \in \Omega^c$ which minimizes $\sum_{i=1}^{3} \alpha_i d_i(p, q, \varphi^{n-1})$. If there are several pixels minimizing this quantity we just choose one randomly. The iterative process should end when the map remains unchanged during two consecutive iterations: $\varphi^n(p) = \varphi^{n-1}(p), \forall p \in \Omega$. As there is no assurance that the iterative process leads to a stationary point, we set by hand the maximum number of iterations. In practice, we have observed that the energy shows a decreasing behavior: we have obtained good results using 10 iterations and observed that the inpainted image does not evolve much after more iterations.

A. First Energy Term: Self-Similarity and Texture Synthesis

Iterative texture synthesis has already been proposed in many papers [4], [23], [3], [24], [25]. The idea is to use the output image at the previous iteration as initialization for the next iteration. The main motivation for such an approach is texture refinement. In particular, during the initialization (i.e., the first iteration) the inpainting is done from the boundary inwards which often creates edge discontinuities. With the iterative process, these discontinuities as well as the "garbage growing" [1] (propagation of errors) are reduced.

A variational formulation of Efros and Leung's texture synthesis method [1] (which compares square patches using a Sum of Squared Differences (SSD)) in terms of the correspondence map was introduced in [3], [22]. As in those works, we propose to compute the correspondence map φ that minimizes the energy

$$\mathcal{E}_{E-L}(\varphi) = \sum_{p \in \Omega} \sum_{k \in \Psi_s} \|u(\varphi(p+k)) - u(\varphi(p)+k)\|^2.$$

To fix ideas we assume that we are working with a color image and we denote by $\|v\|$ the Euclidean norm of vector $v \in \mathbb{R}^3$.

By iterating Efros and Leung's algorithm [1], we are effectively looking for a minimum of the following energy at each iteration n:

$$\mathcal{E}_{E-L}(\varphi, \varphi^{n-1})$$

Fig. 4. Result after two iterations of the iterative algorithm on a grass image with 9×9 patches. The last two images display the value of the correspondence map for several pixels of the mask. (a) Masked image; (b) initialization; (c) iteration 2; (d) φ at initialization; and (e) φ at iteration 2.

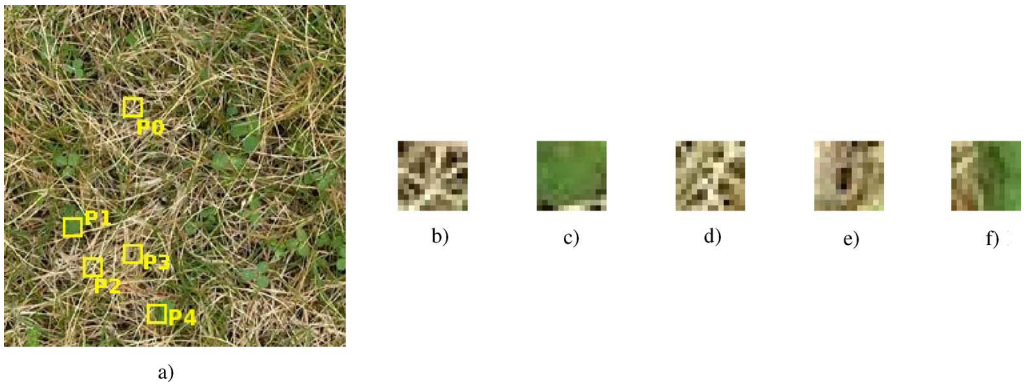


Fig. 5. Sum of squared differences between texture and smooth patches: $d_{\mathrm{SSD}}(\Pi_0, \Pi_1) = 5290$; $d_{\mathrm{SSD}}(\Pi_0, \Pi_2) = 5443$; $d_{\mathrm{SSD}}(\Pi_0, \Pi_3) = 5416$; $d_{\mathrm{SSD}}(\Pi_0, \Pi_4) = 5070$. The patch with minimum SSD to Π_0 is Π_4, but visually Π_2 is much closer to Π_0. (a) Original image; (b) patch Π_0; (c) patch Π_1; (d) patch Π_2; (e) patch Π_3; and (f) patch Π_4.

$$= \sum_{p \in \Omega} \sum_{k \in \Psi_s} \|u(\varphi^{n-1}(p+k)) - u(\varphi(p)+k)\|^2 \quad (6)$$

where φ^{n-1} is the correspondence map at iteration $n-1$. Equivalently, we could rewrite (6) as

$$\mathcal{E}_{E-L}(\varphi, \varphi^{n-1}) = \sum_{p \in \Omega} d_{\mathrm{SSD}}\left(P_1^{(n-1)}(p), P(\varphi(p))\right) \quad (7)$$

where $P_1^{(n-1)}(p) := \{u(\varphi^{n-1}(q)) : q \in \Psi_s(p)\}$ denotes the current patch of image values centered at p, $P(\varphi(p)) := \{u(\varphi(p)+k) : k \in \Psi_s\}$ is the patch centered at the new correspondent of p, and d_{SSD} computes the SSD among these patches.

This is a nonconvex energy and we cannot guarantee that this iterative process will find its global minimum. Let us also notice that for each value of patch size s, the global minimum may be different. We have experimentally confirmed that the energy in (6) does indeed decrease with the iterations, often uniformly after the first one.[1]

We have found that the SSD may be problematic for a wide class of images, especially when we have to iterate in order to obtain the solution, as it is in our case. Fig. 4 shows that, as early as at the second step of the iterative process, the bias of the SSD for uniform patches makes a flat region take over most of the inpainting region Ω. The figure also shows that while at the first iteration the mapping function φ is rather "dispersive," already at the second iteration it starts to behave locally as a translation toward flat regions. Fig. 5 explains what is the problem with the SSD. We have a textured patch such as Π_0 which is, *in its overall appearance*, very similar to Π_2. But, pixel-wise, as the SSD is computed, the difference between pixels at the same location in both patches is greater than the difference between the pixels in Π_0 and the *average value* of Π_0. Think of the difference between two sinusoids of average μ which are 180° out of phase: although they look exactly the same, their SSD is greater than the SSD between either sinusoid and the constant function of value μ. This is why patch Π_4, which is rather uniform and does not resemble at all Π_0, is more similar *in terms of SSD* to Π_0 than Π_2 is to Π_0. More formally, we could argue that the patches extracted from an image lie on a manifold of lower dimension, therefore the SSD should be replaced by an appropriate geodesic distance which reflects the structure of the manifold (see [26] and references therein). Conversely, we could impose certain restrictions on the mapping function φ that prevents it to become locally a translation towards uniform patches. There are several works in the literature which introduce some assumptions on φ: in [3] the authors propose to encourage φ to locally look like the identity function, in [22] the solutions are searched within a set of piecewise roto-translations.

As we will mention in the conclusion, the introduction of restrictions on φ to mitigate its "dispersive" nature is subject of further research; in this paper we have opted to define a new similarity measure which is more suitable than the SSD for iterative texture synthesis. Our proposal for the texture synthesis term d_1 of (3) is

[1]We have always observed an *asymptotic* monotonic decrease, although not necessarily after the second iteration.

a) b) c) d)

Fig. 6. Result after two iterations of texture synthesis using the d_1 measure instead of the SSD, with 9×9 patches. (a) Initialization; (b) iteration 2; (c) φ at initialization}; and (d) φ at iteration 2.

$$d_1(p, \varphi(p), \varphi^{n-1}|_{\Psi_s(p)\setminus\{p\}}) := \frac{1}{|\Psi_s|} d_{\mathrm{SSD}}(P_1^{(n-1)}(p), P(\varphi(p))) \cdot d_B\left(P_1^{(n-1)}(p), P(\varphi(p))\right) \quad (8)$$

where $|\Psi_s|$ is the number of pixels of the patches being compared, d_{SSD} is the SSD metric discussed previously and for a given pair of patches $P(p), P(q)$, the function $d_B(P(p), P(q))$ measures the difference between patches $P(p)$ and $P(q)$

$$d_B(P(p), P(q)) = \sqrt{1 - \sum_{i=1}^{B} \sqrt{\rho_p(i)\rho_q(i)}} \quad (9)$$

where ρ_p, ρ_q are the histograms of u computed on the patches $P(p), P(q)$, respectively, and B is the number of bins of the histograms.[2] Let us comment on the rationale behind this choice for d_1, which combines the SSD and the distance d_B. While d_B permits to distinguish a smooth patch from a textured one because they have different distributions, it is rotation invariant and therefore we still need the SSD to be able to distinguish that case. In Fig. 6, we display the result obtained using the d_1 measure. Note how the blur has disappeared when we compare it to the result displayed in Fig. 4(c) and how the arrows keep pointing in different directions after two iterations.

B. Second Energy Term: Diffusion and Propagation

Let us denote $u^{\varphi}(p) = u(\varphi(p)), p \in \Omega$. If we choose Laplacian diffusion for the second term of our proposed energy, then it could be written as

$$\mathcal{E}_2(\varphi) = \sum_{p\in\Omega} \sum_{k\in\Psi_s} ||\Delta u^{\varphi}(p+k)||^2$$

where $\Delta u^{\varphi}(p)$ denotes the discrete Laplacian of u^{φ} at p.

As we did in Section II-A, here we use an iterative algorithm updating the correspondence map step by step. For that, assume that we already know φ^{n-1} and we intend to compute φ^n. Given a pixel $p = (x, y)$, let

$$v(p, \varphi) = \frac{1}{4}[u^{\varphi}(x+1, y) + u^{\varphi}(x-1, y) + u^{\varphi}(x, y+1) + u^{\varphi}(x, y-1)]. \quad (10)$$

Then we may write the (iterative) diffusion term $\mathcal{E}_2$ as

$$\sum_{p\in\Omega} \sum_{k\in\Psi_s} ||v(p+k, \varphi^{n-1}) - u^{\varphi}(p+k))||^2. \quad (11)$$

In order to reinforce the coherence we would like to impose that

$$u(\varphi(p+k)) \approx u(\varphi(p) + k) \quad (12)$$

and we redefine (11) as

$$\mathcal{E}_2(\varphi, \varphi^{n-1}) = \sum_{p\in\Omega} d_2(p, \varphi(p), \varphi^{n-1}) \quad (13)$$

where for given pixels $p \in \Omega, q \in \Omega^c$ we denote

$$d_2(p, q, \varphi^{n-1}) = \sum_{k\in\Psi_s} ||v(p+k, \varphi^{n-1}) - u(q+k)||^2. \quad (14)$$

Minimizing (13) amounts then to find the correspondence map that best matches the image u with the image v defined in (10). We call v the *diffusion image*. In the case just mentioned, with Laplacian diffusion, the definition of v comes from the definition of the energy. But we may choose instead to specify the diffusion image directly, so that the image information is transported in the direction of the level lines of the image, e.g.,

$$v(p, \varphi^{n-1}) = \frac{\sum_{l\in\Psi_r, l\neq 0} \pi(p, l) u(\varphi^{n-1}(p+l))}{\sum_{l\in\Psi_r, l\neq 0} \pi(p, l)} \quad (15)$$

where p is any pixel and $\pi(p, l)$ are the weights proposed in the coherence transport diffusion method of Bornemann and März [18] ($\pi(p, l)$ is higher when the pixel $(p + l)$ is in the direction of the level line at p). Notice that we have considered a patch Ψ_r of size $(2r + 1) \times (2r + 1)$ where r may be different from s.

C. Third Energy Term: Coherence

Again, we proceed in an iterative way and we assume that we have φ^{n-1}. As we already mentioned in Section II-B, coherence is represented by the relation (12). In this formulation, it

[2]This function is a modified version of the Bhattacharyya probability density distance so that it satisfies the triangle inequality. In practice, a histogram is computed on each patch by using 8 bins for each color dimension (i.e., 512 bins in the case of color images).

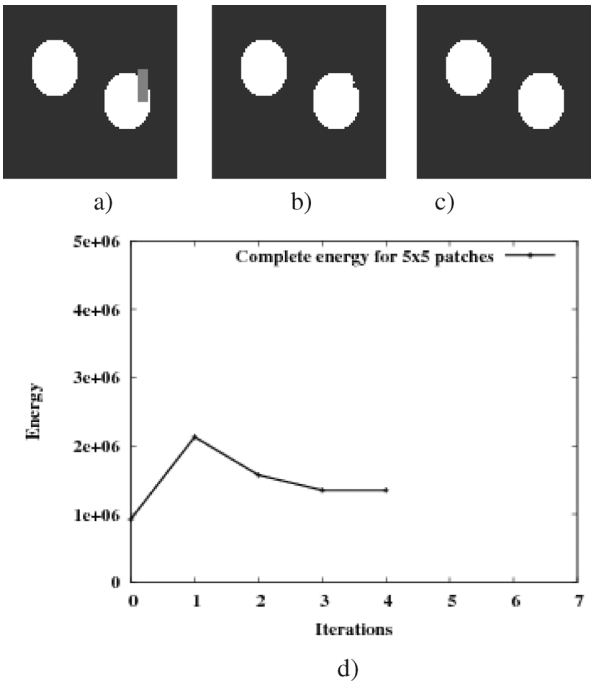


Fig. 7. Result of proposed algorithm for an image containing repetitive patterns. The mask to be inpainted is the gray rectangle and the size of the patches is 5×5. (a) Masked image; (b) initialization; (c) iteration 4; and (d) evolution of the energy value, (19)

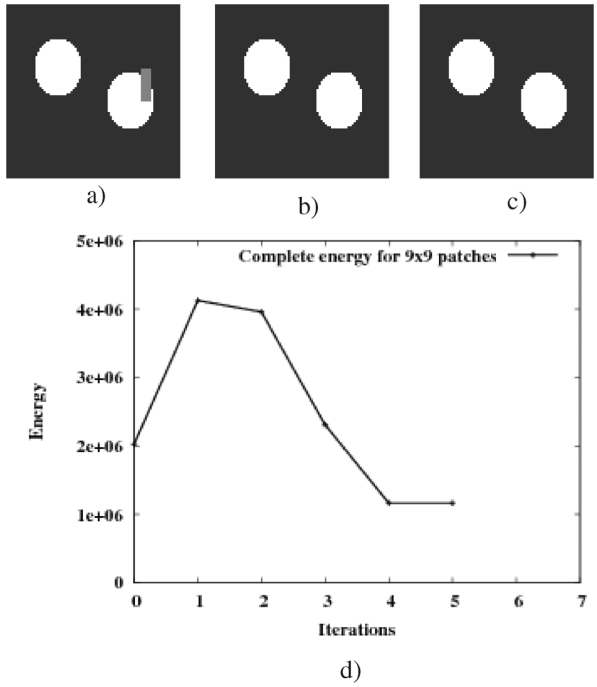


Fig. 8. Result of proposed algorithm for an image containing repetitive patterns. The mask to be inpainted is the gray rectangle and the size of the patches is 9×9. (a) Masked image; (b) initialization; (c) iteration 4; and (d) evolution of the energy value, (19).

is implicit that the correspondence map has to be rigid in some sense: it should prefer to assign nearby corresponding points to nearby pixels. We could ask even more and consider that for any $k \in \Psi_s$, the family of values $\{u(\varphi^{n-1}(p+l)+k-l) : l \in \Psi_s\}$ is representative of the value of $u(\varphi(p) + k))$ and permits to give an estimate of it. To that end, we add a last term to our energy that encourages the estimation to be close to this set of values, by favoring the similarity of the patches corresponding to neighboring pixels. To make this estimate less sensitive to the possibly aberrant values in the neighborhood of p, we propose to use median average instead of mean average (as in [7]) and compute

$$w(p,k,\varphi^{n-1}) = \text{median}_{l\in\Psi_s}\{u(\varphi^{n-1}(p+l)-l+k)\}. \quad (16)$$

Then we compute $w(p,k,\varphi^{n-1})$ and define the reinforcement coherence term as

$$\mathcal{E}_3(\varphi,\varphi^{n-1}) = \sum_{p\in\Omega} d_3(p,\varphi(p),\varphi^{n-1}) \quad (17)$$

where

$$d_3(p,q,\varphi^{n-1}) = \sum_{k\in\Psi_s} [w(p,k,\varphi^{n-1}) - u(q+k)]^2 \quad (18)$$

is the coherence similarity measure. In other words, the candidate $\varphi(p)$ is chosen so that the intensity of its neighborhood is close to the intensity of the "shifted" candidates of the neighbors of p. In consonance with the previous section, we call w the *coherence image*.

Notice that the underlying energy could be written as

$$\mathcal{E}_3(\varphi) = \sum_{p\in\Omega}\sum_{k\in\Psi_s} [w(p,k,\varphi) - u(\varphi(p)+k)]^2.$$

D. Toy Example

We have now defined three different similarity measures, one for texture synthesis (8), one for diffusion (14) and one for coherence (18). These measures are combined into the energy

$$\mathcal{E}(\varphi,\varphi^{n-1}) = \sum_{p\in\Omega}\sum_{i=1}^{3} \alpha_i d_i(p,\varphi(p),\varphi^{n-1}) \quad (19)$$

where $\alpha_i \geq 0$ are some weights. In Section III, we shall describe an improved choice of the weights α_i, but for the time being we just choose $\alpha_i = (1/3), i = 1,2,3$. We test our formulation on a toy example and compute the evolution of the full, three-terms energy value of (19). The results are in Figs. 7 and 8. As we can see, the result obtained with 5×5 patches is not satisfactory while with 9×9 patches the image is inpainted correctly. This should be no surprise since the global minimum may depend upon the size of the patch. Notice that the minimum value of the energy associated with the correct inpainting is not zero. Indeed, as the diffusion and the coherence images are smoothed versions of the original image, they are no longer binary black and white images, and therefore the energy value for these two terms can not be zero. As stated before, the proposed iterative algorithm does not assure the convergence to the global or a local minimum of the energy but we have observed an asymptotic monotonic decrease of the energy.

III. Implementation Details

1) Initialization Step: The algorithm must be applied to an image where there are no valid values inside the mask Ω: while this is standard procedure in texture synthesis, for our diffusion and coherence terms it would be much easier to compute the measures d_2 and d_3 if the whole image was already estimated. Therefore, we perform an initialization step where we fill-in Ω using only the texture synthesis term.

2) Reducing the Computational Cost: Searching for the best patch among all the patches of the image is extremely time-consuming. Recently, an algorithm to speed-up patch searches has been proposed in [27], using randomized sampling and exploiting the coherence in natural images. In order to get a faster algorithm, we propose to use a method that we name *pruning*, inspired by the label pruning of [7]. We reduce the number of possible candidates for each pixel of Ω after the initialization step has been performed. The number N of candidates kept for each pixel is a fixed parameter set by the user. These candidates are simply the ones that had the highest similarity (smallest measure d_1) to the pixel during the initialization step. In the sequel, we will denote as $C(p)$ the *pruning set* (with N candidates) for the pixel p. While the initialization still has an important computational cost, the subsequent iterations are now much faster, and the visual quality does not deteriorate when N is high enough (higher than 200 in practice).

Despite the pruning, the processing of each iteration still requires quite some time (even without taking into account the initialization step), in particular due to the fact that a histogram still has to be computed for each patch. We have therefore run some experiments where we apply the texture similarity measure d_1 only at the initialization, use it to perform the pruning, and then in all subsequent iterations use just the SSD to compare patches. The results are very similar to those obtained when using d_1 at every iteration, but at a much lower computational cost. This makes sense since d_1 has been introduced in order to reduce the appearance of flat regions in the results, and by applying d_1 for the pruning we are already discarding from the pruning set all pixels that could produce flatness.

3) Improved Choice of the Weights and Practical Computation of the Correspondence Map: The final inpainting algorithm is an iterative method that at each iteration n and for each pixel p finds the best corresponding pixel $\varphi^n(p)$ by solving:[3]

$$\varphi^n(p) \in \arg\min_{q\in C(p)} \sum_{i=1,2,3} \alpha_i d_i(p,q,\varphi^{n-1}). \tag{20}$$

Let us discuss an improved choice of weights. The weights α_i define the influence of each of the three terms on the result. They have to be defined depending upon the properties of each pixel of the image. For example, for a pixel situated in a textured area, the parameter α_1 should be larger than the two other ones, while if it is located on a sharp contour, α_2 should be the largest one. It is more difficult to interpret the influence of α_3 and its value will be given by a formula similar to the ones used for α_1 and α_2.

[3]Note that the minimum is not always unique so we used the symbol $\in$ instead of the equality. If there are several solutions one is chosen randomly.

First the diffusion and coherence images are computed from (10) or (15) and (16). We then extract, for each pixel p, the texture patch $P_1^{(n-1)}(p)$ from the inpainted image u^{n-1} at previous iteration $n-1$ as the $(2s+1)\times(2s+1)$ image patch centered at p. The structure patch $P_2^{(n-1)}(p)$ extracted from the diffusion image and the coherence patch $P_3^{(n-1)}(p)$ from the coherence image are defined in the same way. Similarly, we define the candidate patches $Q_i(p), i=1,2,3$, as the patches centered at some points q_i realizing the minimum

$$m_i(p) := \min_{q_i\in\Omega^c} d_i(p,q_i,\varphi). \tag{21}$$

In order to define the weights, we assume that their values should depend upon the "validity" of the best candidate patch $Q_i(p)$. That is, if no good candidate patch is present in the image for the patch $P_i^{(n-1)}(p), i=1,2,3$, the value of the corresponding weight should be small. More specifically, the values of the weights must be a decreasing function of the values of $m_i(p)$. Because of all the considerations mentioned previously, we propose to automatically compute the weights with these functions

$$\alpha_i(p) = e^{-\frac{m_i}{\sigma}}, \quad i=1,2,3 \tag{22}$$

where σ is set as

$$\sigma = \frac{m_1+m_2+m_3}{3}. \tag{23}$$

Now, the corresponding pixel $\varphi^n(p)$ is computed using (20) and the inpainted image at iteration n is finally given by

$$u^n(p) = u(\varphi^n(p)) \quad \forall p.$$

A. Multiresolution Scheme

There are several reasons for introducing a multiresolution scheme in our algorithm. First, it permits to accelerate the initialization step, as it is applied on a smaller image. For all other iterations and scales, the full algorithm (with three terms) is applied, with the computational cost savings given by the pruning technique. Second, it allows to capture different details of the image, something which could only be achieved without multiresolution by locally changing the size of the patches: if we use the same size for all the scales of the pyramid, different patterns of the image are captured. The larger structures can be captured correctly at the coarsest scales.

Multiresolution schemes have already been used for image inpainting. In particular, in [25], they have been used for enlarging the patches (different types of structural components are captured at different scales) and to guide the search of patches with the already computed neighbors of each pixel. Also, multiresolution may be implicit depending upon the choice of the dictionary in sparsity-based inpainting. Here we use a slightly different method as the size of the patches is kept the same at all scales and no constraint between scales is imposed (we do not keep or use the values of the inpainted pixels at the previous scale except when initializing the next level). Hence, the algorithm works as follows. First, a Gaussian pyramid is built from the original image. Next, the iterative inpainting algorithm is applied on the smallest image (the highest one in the pyramid).

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

Fig. 9. Influence of the three terms. (a) Original image. (b) Result of our algorithm. (c) Result using only the texture synthesis term $(\alpha_1 = 1, \alpha_2 = \alpha_3 = 0)$. (d) Result using only the diffusion term $(\alpha_2 = 1, \alpha_1 = \alpha_3 = 0)$. (e) Result using only the coherence term $(\alpha_3 = 1, \alpha_1 = \alpha_2 = 0)$. Notice how the text in the middle lantern at the bottom row is only successfully inpainted when the three terms are combined.

(a) (b) (c)

Fig. 10. Effect of the choice of geometric term. (a) Original image, from [31]. (b) Result with Laplacian diffusion, (10). (c) Result with coherence transport diffusion, (15). The particular choice of geometric diffusion method does not seem to affect the result.

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

Fig. 11. Effect of the choice of patch size. (a) Original image; (b) result with 3×3 patches; (c) result with 5×5 patches; (d) result with 7×7 patches; and (e) result with 11×11 patches. With small patches, the middle-sized blob at the left of the image is not correctly inpainted. Big patches also give poor results because the mask Ω is spread all over the image, so there are not enough big patches completely inside Ω^c.

The result obtained on this image is then used as the initialization for the second level of the pyramid, and so on.

B. Poisson Editing

Despite all the great advantages of the multiresolution scheme, we have observed that it may create some illumination discontinuities at the border of the mask. At the coarsest scale, it appears sometimes that there are no more candidate patches of the right color while these patches exist at a finest scale. To deal with this defect, we add a postprocessing step of Poisson image editing [28]. A small modification is applied to the standard method: for each $p \in \Omega$ we compute $\nabla u(\varphi(p))$ instead of $\nabla u(p)$.

IV. Experimental Results

In this section, we test our algorithm with respect to variables such as patch size or diffusion method used, and compare it with state-of-the-art inpainting methods. Hereafter, each time the parameters are not mentioned, we set the following values: the patch size $s = 4$, the size of the pruning set $N = 200$, the number of scales in the pyramid is 3 and a Laplacian diffusion was used. Also we used dyadic Gaussian pyramids, built with standard deviations of 1 and nearest neighbor interpolations.

Fig. 9 compares the results obtained when we consider the three terms of the energy (the proposed algorithm) and when we only take into account one of the terms. Notice how the text in the middle lantern at the bottom row is only successfully inpainted when the three terms are combined. For this image of size 256×163, the computational cost of our non optimized algorithm is around 5 minutes with a dual-core PC (RAM 2 GB and CPU 2.13 GHz).

Fig. 10 compares the results obtained when we use different diffusion images v, either Laplacian diffusion with (10) or coherence transport diffusion with (15): we can see that the differences are minimal. For this image of size 750×563, the computational cost with the Laplacian diffusion is about 15 minutes while it takes about 25 minutes with the coherence transport diffusion. As the particular choice of geometric diffusion method does not seem to affect the result, it appeared preferable to work with the Laplacian diffusion.

Fig. 11 compares the results obtained when we use different patch sizes. For this figure, the Gaussian pyramid only had two scales. With small patches the algorithm is unable to inpaint

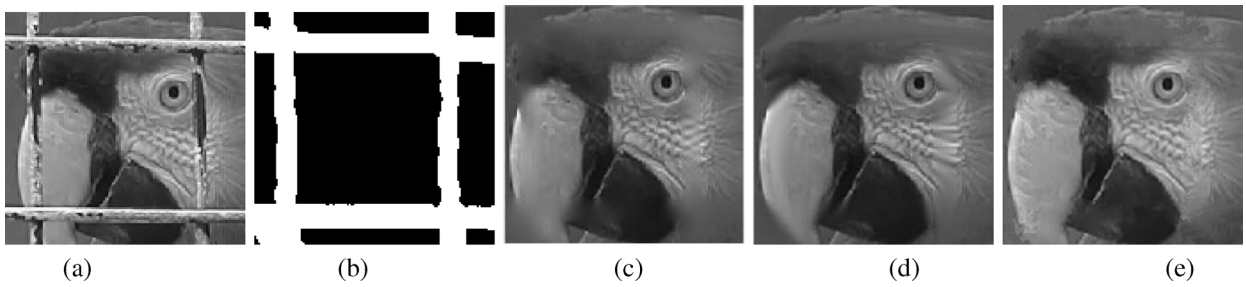


Fig. 12. Comparisons with other algorithms. (a) Original image; (b) mask Ω in white; (c) result from Fadili *et al.* [11]; (d) result from Tschumperle *et al.* [16]; and (e) result from our algorithm. The differences are most noticeable at the tip of the beak, the feathers on the forehead and on the vertical strip to the side of the eye.

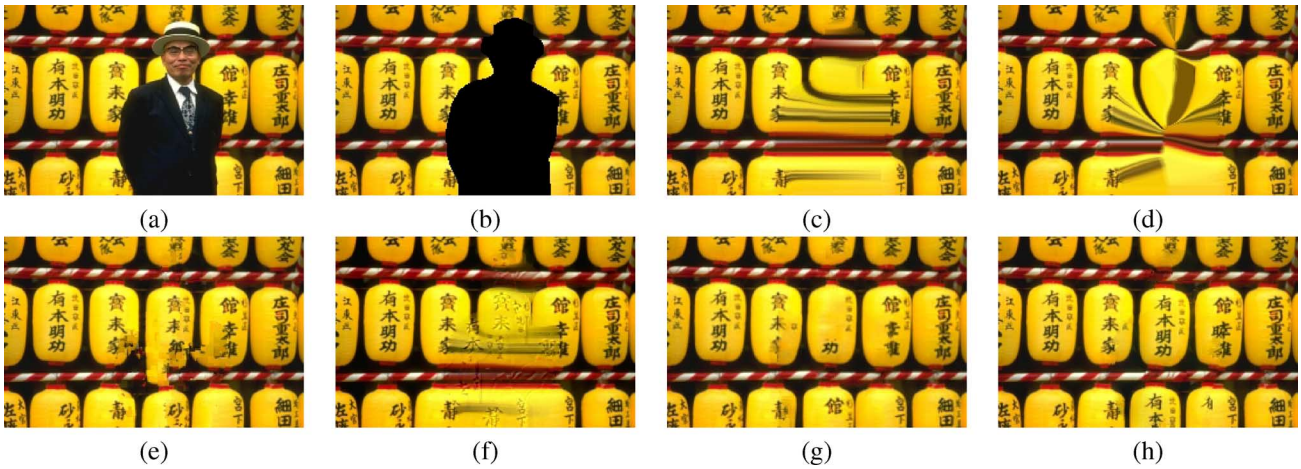


Fig. 13. Comparison of several inpainting algorithms. (a) Original image; (b) in black the mask to be inpainted; (c) result from Tschumperle [29]; (d) result from Bornemann [18]; (e) result from Criminisi *et al.* [2]; (f) result from Bugeau *et al.* [30]; (g) result from Komodakis *et al.* [7]—taken directly from the paper; and (h) result obtained with our algorithm. Our method is able to correctly inpaint the covered lanterns, the text on these lanterns and the red and white striped bars, while the other techniques fail in some of these aspects.

Fig. 14. Comparison with method by Hays *et al.* [31]. (a) Original image; (b) result from [31]; and (c) result from our algorithm. The reconstruction of the building facade is acceptable (though not perfect), but the inpainting of the road part of the image has failed.

correctly the middle-sized blob at the left of the image. But with big patches another problem arises: the mask Ω is spread all over the image, so there are not enough big patches completely inside Ω^c and the final result is poor. We should incorporate into our algorithm the possibility that the candidate patches are not completely included in Ω^c, and how to do this properly is the subject of further research.

Fig. 12 compares our algorithm with the results obtained with the state of the art techniques of Fadili *et al.* [11] and Tschumperle *et al.* [16], as taken from these papers. The differences are most noticeable at the beak, the feathers in the forehead, and the vertical strip to the side of the eye. The result obtained with our method contains less blur, in particular at the side of the eye, but the bottom of the beak is more fuzzy.

On Fig. 13, some additional comparisons with existing inpainting methods are presented. We have compared with five other methods already mentioned in the paper: the diffusion methods of [29][4] and [18], the texture synthesis algorithm from [2], the texture and coherence method from [7], and finally the earlier method we proposed in [30]. All of these methods require the tuning of some parameters. For the method from [29] and [30], as suggested in the original papers, we set the con-

[4]For the results with Tschumperlé's method we used the code available on-line.

Fig. 15. Application to image editing. (a) Original image, from [32]; (b) our inpainting result; (c) compositing the kid over our result; (d) result from [32]; and (e) result from [27].

Fig. 16. Application to image resizing. (a) Original image by Zanettco, sized 500 × 375; (b) in white the mask to be inpainted; and (c) Resized 600 × 375.

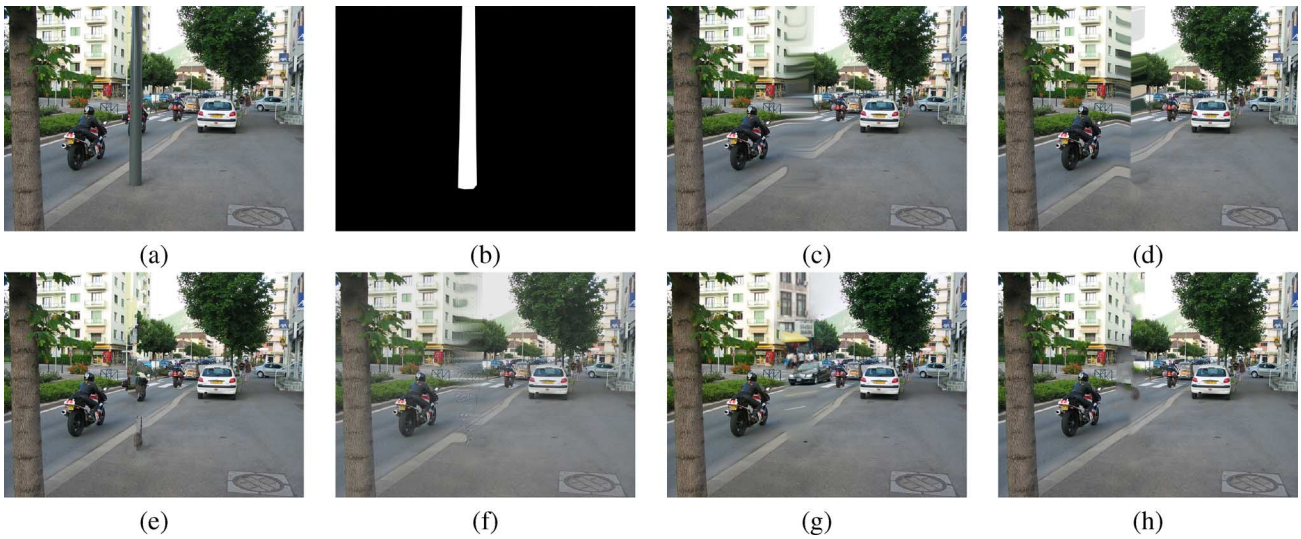


Fig. 17. Example where none of these methods seems to work. (a) Original image.;(b) in white the mask to be inpainted; (c) result from Tschumperle [29]; (d) result from Bornemann [18]; (e) result from Criminisi *et al.* [2]; (f) result from Bugeau *et al.* [30]; (g) result from Hays *et al.* [31]; and (h) result obtained with our algorithm.

tour preservation parameter p_1 to 0.001, the structure anisotropy $p_2 = 100$, the time step $dt = 150$, and the number of iterations nb equal to 100. For the technique from [18], we set the averaging radius $\epsilon = 5$, the sharpness parameter $\kappa = 25$, the scale parameter for presmoothing $\sigma = 1.4$, and for postsmoothing $\rho = 4$. For both our method and the ones from [2] and [30], we used 9×9 patches. We can see that our method is able to correctly inpaint the covered lanterns, the text on these lanterns and the red and white striped bars, while the other techniques fail in some of these aspects.

Fig. 14(c) shows a 800×600 image inpainted with 11×11 patches and four scales. The mask Ω (not shown) covers the kiosk in the center of the image. The reconstruction of the building facade is acceptable, but the inpainting of the road part of the image has failed. The reason is that this part is not present in the rest of the image, so there are not enough patches to copy from. This limitation is shared by all methods based upon patches. For comparison we show in Fig. 14(b) the result obtained by Hays *et al.* [31], where they search a database of millions of images, but in this case the result is not good either.

Fig. 15 shows an application of our proposed method (with 11×11 patches and four scales) to image editing. Here the mask Ω corresponds to a region that we want to relocate in the image, so first we inpaint Ω and then we superimpose the original region at the desired location. We compare with the state of the art techniques of Cho *et al.* [32] and Barnes *et al.* [27].

Fig. 16 shows an application of the proposed method to image resizing, where we have enlarged the original 500×375 image by adding 100 columns to the right. Four scales were used.

Finally, Fig. 17 shows an image where none of the methods discussed previously seems to work. The problem here is that diffusion methods [Fig. 17(c) and (d)] are able to synthesize new information but they are not able to deal with texture, while patch-based methods [Fig. 17(e)-(h)] treat texture correctly but are not good at creating new information. Even if our algorithm combines these different approaches, it is still a patch-based method that may not give good results if the information needed is not already present in the image. For most of the mask, no good candidate patches exist. This is even more true with the multiresolution scheme (which is unavoidable, given the size of the image, 800×600). For example, for the tree, there are not so many areas from where to copy and the method had to make a compromise between the tree and the façade.

V. Conclusion

In this article we have started by stating explicitly the three models or building blocks which are common to all the most successful inpainting algorithms. We then combine them into one energy functional. We also provided a working algorithm for image inpainting trying to approximate the minimum of this energy functional. Our experiments show that the combination of all three terms in the proposed energy works better than taking each term separately, something which remains consistent when comparing our results with other techniques, none of which is based upon all the three, but at most two, of the stated building blocks. The particular choice of geometry/diffusion method does not affect noticeably the results, but the size of the patches used is a relevant parameter. When the image has not enough patches to copy from, either because the mask is too spread and the patch size is large, or because the mask is placed on a *singular* location on the image (with surroundings that can't be found anywhere else in the image) then the results are poor; this problem is common to all patch-based inpainting methods, and although the presence of a geometry term seems to help, it is clearly not enough. We are currently trying to deal with these problems, as well as working on the speeding-up of our algorithm by optimizing the search in the patch space.

References

[1] A. Efros and T. Leung, "Texture synthesis by non-parametric sampling," in *Proc. IEEE Int. Conf. Computer Vision*, 1999, vol. 2, pp. 1033–1038.

[2] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based inpainting," *IEEE Trans. Image Process.*, vol. 13, no. 9, pp. 1200–1212, Sep. 2004.

[3] L. Demanet, B. Song, and T. Chan, "Image inpainting by correspondence maps: a deterministic approach," UCLA CAM R, Tech. Rep. 03-04, Aug. 2003.

[4] M. Ashikhmin, "Synthesizing natural textures," in *Proc. ACM Symp. Interactive 3D Graphics*, 2001, pp. 217–226.

[5] Y. Wexler, E. Shechtman, and M. Irani, "Space-time video completion," in *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, 2004, vol. 1, pp. 120–127.

[6] Y. Wexler, E. Shechtman, and M. Irani, "Space-time video completion," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 1463–476, Mar. 2007.

[7] N. Komodakis and G. Tziritas, "Image completion using global optimization," in *Proc. IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition*, 2006, pp. 442–452.

[8] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[9] M. Elad, J. Starck, P. Querre, and D. Donoho, "Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA)," *Appl. Comput. Harmon. Anal.*, vol. 19, no. 3, pp. 340–358, 2005.

[10] G. Peyre, J. Fadili, and J. Starck, "Learning adapted dictionaries for geometry and texture separation [6701-66]," *Proc. SPIE – Int. Soc. Opt. Eng.*, vol. 6701, no. 2, pp. 6701–, 2007.

[11] M. Fadili, J.-L. Starck, and F. Murtagh, "Inpainting and zooming using sparse representations," *Comput. J.*, vol. 52, no. 1, pp. 64–79, 2009.

[12] S. Masnou and J. M. Morel, "Level lines based disocclusion," in *Proc. IEEE Int. Conf. Image Processing*, 1998, pp. 259–263.

[13] S. Masnou, "Disocclusion: A variational approach using level lines," *IEEE Trans. Image Process.*, vol. 11, no. 2, pp. 68–76, Feb. 2002.

[14] M. Nitzberg, D. Mumford, and T. Shiota, *Filtering, Segmentation, and Depth*. Berlin, Germany: Springer-Verlag, 1993.

[15] M. Bertalmío, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. SIGGRAPH: ACM Special Interest Group on Computer Graphics and Interactive Techniques*, New York, 2000, pp. 417–424.

[16] D. Tschumperle and R. Deriche, "Vector-valued image regularization with pdes: A common framework for different applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 4, pp. 506–517, Apr. 2005.

[17] M. Bertalmío, "Strong-continuation, contrast-invariant inpainting with a third-order optimal pde," *IEEE Trans. Image Process.*, vol. 15, no. 7, pp. 1934–1938, Jul. 2006.

[18] F. Bornemann and T. März, "Fast image inpainting based on coherence transport," *J. Math. Imag. Vis.*, vol. 28, no. 3, pp. 259–278, 2007.

[19] M. Bertalmío, L. Vese, G. Sapiro, and S. Osher, "Simultaneous structure and texture image inpainting," *IEEE Trans. Image Process.*, vol. 12, no. 8, pp. 882–889, Aug. 2003.

[20] G. Gilboa and S. Osher, "Nonlocal linear image regularization and supervised segmentation," *SIAM Multiscale Model. Sim.*, vol. 6, pp. 595–630, 2007.

[21] G. Peyré, S. Bougleux, and L. Cohen, "Non-local regularization of inverse problems," in *Proc. Eur. Conf. Computer Vision*, 2008, pp. 57–68.

[22] J.-F. Aujol, S. Ladjal, and S. Masnou, "Exemplar-based inpainting from a variational point of view," 2008.

[23] A. Efros and W. Freeman, "Image quilting for texture synthesis and transfer," in *Proc. SIGGRAPH: ACM Special Interest Group on Computer Graphics and Interactive Techniques*, 2001, pp. 341–346.

[24] S. Shin, T. Nishita, and S. Shin, "On pixel-based texture synthesis by non-parametric sampling," *Comput. Graph.*, vol. 30, no. 5, pp. 767–778, Oct. 2006.

[25] L. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," in *PROC. SIGGRAPH: ACM Special Interest Group on Computer Graphics and Interactive Techniques*, 2000, pp. 479–488.

[26] G. Peyré, "Manifold models for signals and images," *Comput. Vis. Image Understand.*, vol. 113, pp. 249–260, 2009.

[27] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 1–11, 2009.

[28] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," *ACM Trans. Graph.*, vol. 22, no. 3, pp. 313–318, 2003.

[29] D. Tschumperlé, "Fast anisotropic smoothing of multi-valued images using curvature-preserving pde's," *Int. J. Comput. Vis.*, vol. 68, no. 1, pp. 65–82, 2006.

[30] A. Bugeau and M. Bertalmío, "Combining texture synthesis and diffusion for image inpainting," in *Proc. Int. Conf. Computer Vision Theory and Applications*, 2009, pp. 26–33.

[31] J. Hays and A. Efros, "Scene completion using millions of photographs," *Commun. ACM*, vol. 51, no. 10, pp. 87–94, 2008.

[32] T. S. Cho, M. Butman, S. Avidan, and W. T. Freeman, "The patch transform and its applications to image editing," in *Proc. IEEE Computer Soc. Conf. Computer Vision and Pattern Recognition*, 2008, p. 1–8.

Aurélie Bugeau received the Ph.D. degree in signal processing from the University of Rennes, France, in 2007.

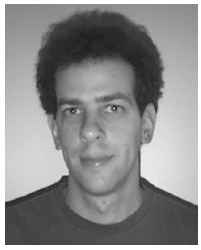
Since November 2007, she holds a Post-Doctoral position in the foundation Barcelona Media, Barcelona, Spain. Her main research interests include objects detection and tracking, data clustering, image and video inpainting.

Marcelo Bertalmío received the B.Sc. and M.Sc. degrees from Universidad de la República, Uruguay, in 1996 and 1998 respectively, and the Ph.D. from the University of Minnesota in 2001.

Since 2006, he has been an Associate Professor at the Universitat Pompeu Fabra, Spain. His research include image processing and computer vision and their applications to the movie industry.

Vicent Caselles (M'10) received the Licenciatura and Ph.D. degrees in mathematics from Valencia University, Spain, in 1982 and 1985, respectively.

Currently, he is a Professor at the Pompeu Fabra University, Barcelona, Spain. His research interests include image processing, computer vision, and the applications of geometry and partial differential equations to both previous fields.



Guillermo Sapiro (M'94–SM'03) was born in Montevideo, Uruguay, on April 3, 1966. He received the B.Sc. (summa cum laude), M.Sc., and Ph.D. degrees from the Department of Electrical Engineering, The Technion—Israel Institute of Technology, Haifa, in 1989, 1991, and 1993, respectively.

After postdoctoral research at the Massachusetts Institute of Technology, he became Member of Technical Staff at the research facilities of HP Labs in Palo Alto, CA. He is currently with the Department of Electrical and Computer Engineering, University of Minnesota, where he holds the position of Distinguished McKnight University Professor and Vincentine Hermes-Luh Chair in Electrical and Computer Engineering. He works on differential geometry and geometric partial differential equations, both in theory and applications in computer vision, computer graphics, medical imaging, and image analysis. He has authored and coauthored numerous papers in this area and has written a book published by Cambridge University Press, January 2001.

Dr. Sapiro co-edited a special issue of the IEEE TRANSACTIONS ON IMAGE PROCESSING and the *Journal of Visual Communication and Image Representation*. He was awarded the Gutwirth Scholarship for Special Excellence in Graduate Studies in 1991, the Ollendorff Fellowship for Excellence in Vision and Image Understanding Work in 1992, the Rothschild Fellowship for Post-Doctoral Studies in 1993, the Office of Naval Research Young Investigator Award in 1998, the Presidential Early Career Awards for Scientist and Engineers (PECASE) in 1998, and the National Science Foundation Career Award in 1999. He is a member of SIAM and the founding Editor-in-Chief of the *SIAM Journal on Imaging Sciences.*