Pattern Recognition Letters 36 (2014) 36–45

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Combining anisotropic diffusion, transport equation and texture synthesis for inpainting textured images

Wallace Casaca [a,b,*], Maurílio Boaventura [c], Marcos Proença de Almeida [b,d], Luis Gustavo Nonato [b]

[a] Brown University, School of Engineering, Providence, RI 02912, USA
[b] University of São Paulo, ICMC, Av. Trabalhador São-carlense 400, São Carlos, Brazil
[c] São Paulo State University – UNESP, IBILCE, Rua Cristóvão Colombo 2265, S. José do Rio Preto, Brazil
[d] University of Minho, Department of Polymer Engineering, 4800-058 Guimarães, Portugal

ARTICLE INFO

Article history:
Received 15 February 2013
Available online 6 September 2013

Communicated by Y. Liu

Keywords:
Image inpainting
Anisotropic diffusion
Transport equation
Texture synthesis

ABSTRACT

In this work we propose a new image inpainting technique that combines texture synthesis, anisotropic diffusion, transport equation and a new sampling mechanism designed to alleviate the computational burden of the inpainting process. Given an image to be inpainted, anisotropic diffusion is initially applied to generate a cartoon image. A block-based inpainting approach is then applied so that to combine the cartoon image and a measure based on transport equation that dictates the priority on which pixels are filled. A sampling region is then defined dynamically so as to hold the propagation of the edges towards image structures while avoiding unnecessary searches during the completion process. Finally, a cartoon-based metric is computed to measure likeness between target and candidate blocks. Experimental results and comparisons against existing techniques attest the good performance and flexibility of our technique when dealing with real and synthetic images.

1. Introduction

The problem of inpainting digital images has received great attention by the scientific community in the last decade, mainly due to the growth of important applications such as *image restoration* and *image editing*, which strongly rely on image inpainting to be effective. The basic idea of inpainting is to recover parts of an image that have been damaged or partially occluded by undesired objects. Techniques devoted to perform image inpainting can be organized in different ways. In this work we gather inpainting techniques in four main groups: *texture synthesis and exemplar-based methods*, *PDE (Partial Differential Equations) and variational modeling-based methods*, *techniques based on space transformation and sparse representation*, and *other methods and techniques that combine the previous approaches*. In the following we provide an overview of existing inpainting methods organized according to the proposed groups.

1.1. Texture synthesis and exemplar-based methods

Texture synthesis algorithms rely on the investigation of location and stationariness of texture patterns contained in the image (Efros and Leung, 1999; Efros and Freeman, 2001; Ashikhmin, 2001; Wei, 2002). The inpainting process is carried out by a pixel similarity-based copy-and-paste strategy. The core idea is to find out the group of pixels that best fits into the region to be filled and copy it to that location. This is done by measuring the similarity between group of pixels from the image and on the boundary of the region to be filled. Texture synthesis techniques are effective when the image is made up of a unique texture pattern, but they are prone to fail when multiple textures and homogeneous structures are present in the image simultaneously. The computational cost involved on texture synthesis-based algorithms is also an issue for practical applications.

Texture synthesis algorithms have been significatively improved by the so-called *exemplar/patch-based inpainting techniques*. This class of techniques applies the texture synthesis procedure to blocks of pixels, imposing a priority order for the filling process, as described in (Komodakis and Tziritas, 2007; Li and Zhao, 2011; Cao et al., 2011; Criminisi et al., 2004; Sun et al., 2005). The seminal work Criminisi et al. (2004) defines the filling order based on local image isophotes (lines of constant intensity). Given the inpainting domain Ω and the pixel $p \in \partial\Omega$ with higher priority, the algorithm performs a global search throughout the valid image extension Ω^c to select the most appropriate block of pixels (an exemplar) to fill the neighborhood of p. Improvements in Criminisi et al. (2004) have also been proposed such as (Cheng et al., 2005; Chen et al., 2007; Cai et al., 2008). Although techniques based on exemplar replication usually produce good results, they tend to lead to significant loss of visual congruence while still

[*] Corresponding author. Address: Guiomar A. Calil 236, V. Italia, S.J.R. Preto, Brazil. Tel.: +55 4015805001.
E-mail address: wallace@icmc.usp.br (W. Casaca).

0167-8655/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.patrec.2013.08.023

performing a global search that is computationally costly and prone to produce non-realistic results, as pixels far from the inpainting area are also considered in the process.

1.2. PDE-based inpainting methods

Unlike the approaches based on texture synthesis, *PDE-based methods* are effective when dealing with non-textured images, also presenting a less prohibitive computational cost. The use of PDE in the context of image inpainting was introduced by Bertalmío et al. (2000), proposing a third-order differential equation that simulates the manual inpainting process accomplished by professional restorators. The method transports information through image isophotes while applying an anisotropic diffusion filter to correct the evolution of the inpainting direction. In the spirit of Bertalmío et al. (2000), Shen and Chan (2002) derive a diffusive PDE using the total variation minimization principle. In order to tackle the issue related to the *Connectivity Principle* (Kanizsa, 1979) (this principle claims that broken lines tend to be connected in an unconscious way by the human mind) an improvement of Shen and Chan (2002) has been proposed in (Chan and Shen, 2001), where the authors make use of mean curvature flow to modify the conductivity coefficient of the governing PDE. Other variational/PDE-based approaches have also been proposed, as for example the method described in Tschumperlé and Deriche (2005), which produces good results when applied to small inpainting regions but that is very sensitive to the choice of parameters. Bornemann and März (2007) propose an effective and fast technique that combines a non-iterative transport equation and a fast marching algorithm, demanding however, the tuning of a large number of parameters. Burger et al. (2009) present a subgradient TV-based approach that leads to good image recovery, but its use is limited to grayscale images. Wen-Ze and Zhi-Hui (2008) present an interesting PDE that preserves quite well edges when restoring non-textured images. A common drawback of PDE-based methods is the smoothing effect introduced in the filled region. Moreover, those methods are more effective when inpainting small regions.

1.3. Methods based on sparse representations

Recently, the so-called sparse representation-based methods have been introduced with great acceptance in the context of image inpainting. This group of techniques does not operate in the cartesian image domain, but in transformed domains such as those obtained by DCT (*Discrete Cosine Transform*), *wavelets* (Mallat, 2008), *curvelets* (Candes et al., 2006) and *wave atoms* (Demanet and Ying, 2007). Sparse representation-based methods rely on the assumption that it is possible to represent an image by a sparse combination of a particular set of transforms in which unfilled pixels are hierarchically ordered and predicted by handling these transforms. The seminal work by Guleryuz (2006) adaptively estimates the missing data while updating the corresponding sparse representation of the image through DCT or wavelet transform. In Elad et al. (2005) the reconstruction task is performed by employing a decomposition scheme that splits the given image in two layered-images called cartoon and texture components. The inpainting is then performed in both layers by using a sparse representation technique. Sparse representation was also used in Xu and Sun (2010) to propagate patches according to sparse linear combination of candidate patches. Inpainting based on sparsity analysis can also be achieved under the perspective of statistical Bayesian modeling. Fadili et al. (2009) accomplish the inpainting by solving a missing data estimation problem based on a Bayesian approach combined with an EM (*Expectation Maximization*) algorithm. A Bayesian model that uses simultaneously local and nonlocal sparse representations was proposed in Li (2011), where a DA (*Deterministic Annealing*) optimization scheme is employed to reduce the computational burden. From a practical point of view, although methods based on sparse decomposition produce pleasant results (specially for missing block completion), they tend to introduce blurring effects when restoring large and non-regular regions. Moreover, computational cost is also a hurdle for those methods.

1.4. Hybrid and other methods

Aiming at preserving relevant structures of the image, hybrid approaches intend to exploit the properties of each of the three previous inpainting methodologies. One interesting example is the association between texture replication, PDE and variational models (Komodakis and Tziritas, 2007; Cao et al., 2011; Bugeau and Bertalmío, 2009; Bertalmío et al., 2003; Grossauer, 2004; Aujol et al., 2010; Bugeau et al., 2010). In Bertalmío et al. (2003), for example, the goal is to split a given image f into two components: the cartoon u and texture v, processing each component independently. The components u and v hold geometric structures and texture patterns of f respectively. The decomposition must, a priori, satisfy the relation

$$f = u + v, \tag{1}$$

according to the cartoon/texture theoretical decomposition model (Meyer, 2002), which was enhanced in Vese and Osher (2003, 2006) and later employed, with a numerical scheme, to ensure Eq. (1). After the decomposition, the inpainting method (Bertalmío et al., 2000) and the texture synthesis algorithm (Efros and Leung, 1999) are applied to u and v, respectively. Both outcomes are then combined using Eq. (1) so as to generate the final result. The computational cost of processing both processings is high, mainly when the gap to be recovered is large, and satisfactory results are not always guaranteed. In Aujol et al. (2010), the authors propose a formulation based on continuous variational models in an effort to adapt exemplar-based algorithms that deal with local geometric features while still reconstructing textures.

There are also methods that exploit the inpainting problem through global energy minimization (Wexler et al., 2007; Kawai et al., 2009; Komodakis and Tziritas, 2007; Liu and Caselles, 2013). Wexler et al. (2007) formulate the reconstruction procedure as an optimization problem which employs a combination of dynamic space–time and tree structures. Kawai et al. (2009) rely on modifications of the energy functional proposed in Wexler et al. (2007) improving the spatial localization of the similarity weights and brightness invariance. Komodakis and Tziritas (2007) employ variations of the sum-product algorithm (*loopy belief propagation*) for graphs with cycles associated to "priority-based message scheduling" during the filling process. Liu and Caselles (2013) reformulate the exemplar-based model described in Demanet et al. (2003) as a global optimization problem encoding texture and structure information where the minimizer is obtained by efficiently solving a graph partitioning problem.

Other interesting inpainting methods have been successfully proposed in the literature. Hays and Efros (2007) and Li et al. (2010) have used a huge image database created from the web to find out the best set of images that approximates to the damaged image. This set is then used to perform color, texture and matching-based operations inside the inpainting domain.

1.5. Contributions

Encouraged by the ideas presented in (Criminisi et al., 2004; Bertalmío et al., 2000; Bertalmío et al., 2003; Calvetti et al., 2006) while simultaneously dealing with the adverse effects raised above, we propose a new inpainting method that combines:

- a novel filling order mechanism that relies on transport equation (Bertalmío et al., 2000) and refinement by cartoon-image isophotes;
- a cartoon-driven metric based on isophote orientation to better evaluate the similarity between missing and candidate pixels;
- an innovative strategy (called *dynamic sampling*) to locally sample the candidate pixels, which drastically reduces the computational cost and increases the inpainting quality.

This paper is organized as follows. In Section 2–5 we describe all details about our approach and in Section 6 we perform experiments using our technique against some state-of-the-art methods. Discussion and limitations of the proposed technique are presented in Section 7. Finally, Section 8 brings the conclusions about the work.

2. Pipeline overview

Let $f: D\subset\mathbb{R}^2\to\mathbb{R}^l$ be a given image ($l = 1$: gray-scale image; $l = 3$: RGB color image), $\Omega\subset D$ the region to be inpainted and $\partial\Omega$ its boundary. Here, we assume that D is a rectangular region in $\mathbb{R}^2$.

We start extracting the cartoon component u from f by solving an anisotropic diffusion equation (Barcelos et al., 2003). Image u is a smoothed component of f which contains geometric structures and isophote information of f. An inner product-based metric derived from a transport equation (Bertalmío et al., 2000) is then applied to u in order to set the order on which blocks of pixels in the fill front $\partial\Omega$ will be traversed. The optimal block of pixels is then dynamically assigned to each patch of unfilled pixels in $\partial\Omega$ using the proposed sampling mechanism. The similarity between blocks of pixels is measured based on the proposed cartoon-driven metric. Fig. 1 shows the described pipeline.

3. Filling-in priority based on cartoon image and transport equation

According to Criminisi et al. (2004) and Harrison (2001), an incorrect pixel filling order can lead to unsatisfactory results. Classical approaches typically define the filling order based on image isophotes, which are computed directly from the input image f (e.g. Criminisi et al., 2004; Cheng et al., 2005; Chen et al., 2007). Image isophotes can also be provided by the user (Sun et al., 2005; Kwok and Wang, 2009) or directly inferred from the target image (Cao et al., 2011; Li and Zhao, 2011) to guide the inpainting process.

In contrast to classical methods, our technique computes the orientation of the isophotes (and the filling order) by handling an auxiliar image that encodes the image boundary information. More precisely, the cartoon component u derived from f is used to define isophote directions, since u contains no texture and image edges, as shown in Fig. 2. Fig. 2(a) shows the input image f, Fig. 2(b) the cartoon component u, Fig. 2(c) and (d) show cropped regions corresponding to Fig. 2(a) and (b), and Fig. 2(e)–(f) depict the respective orientation fields derived from 2(c) and (d). As one can observe, the field in Fig. 2(f) is less noisy than the one computed directly from image f (Fig. 2(e)).

To extract the cartoon component u from f, we employ the denoising diffusion equation proposed in Barcelos et al. (2003). Similar to Casaca and Boaventura (2010), where such equation was employed to decompose f into cartoon and texture components so as to satisfy Eq. (1), we compute u by numerically solving the following anisotropic diffusion equation:

$$\frac{\partial f^{(t)}}{\partial t} = g|\nabla f^{(t)}|div\left(\frac{\nabla f^{(t)}}{|\nabla f^{(t)}|}\right) - (1-g)(f^{(t)} - f),$$

$$f(\mathbf{x})^{(0)} = f(\mathbf{x}), \quad \frac{\partial f(\mathbf{x})^{(t)}}{\partial \vec{n}}\Big|_{\partial D\times\mathbb{R}_+} = 0, \quad \mathbf{x}\in D, \quad t\in\mathbb{R}_+, \qquad (2)$$

where $f^{(t)}$ is the scale version of f, $g = g(|\nabla G_\sigma * f^{(t)}|)$ is an edge detection function, G_σ represents the gaussian function and σ is a tuning parameter.

Once the component u has been computed, the next step is to iteratively assign a label to each pixel in the filling front. Before presenting the labeling scheme we first discuss the motivation and basic aspects of the proposed scheme.

It is well known that approaches based on the Laplacian operator

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}, \qquad (3)$$

are a good alternative to the classical first-order gradient filtering for capturing the underlying image structures. Since the Laplacian (3) is a second-order operator, the result of its application is a smoothed image where edges are typically preserved and oscillatory details are filtered out (Paragios et al., 2005). The Laplacian operator has been successfully exploited in Bertalmío et al. (2000) to derive a PDE-based inpainting method formulated from the transport equation:

$$\frac{\partial I}{\partial t} = \nabla(\Delta I)\cdot\nabla^{\perp} I, \qquad (4)$$

where the Laplacian operator Δ is interpreted as a smoothing measure applied to a given *non-textured image I*. According to the authors, Eq. (4) accomplishes the transport of color tonalities (through the smoothing estimator ΔI) towards the vector $\nabla^{\perp} I$. An interesting interpretation of (4) is given in Bornemann and März (2007), where the authors rewrite Eq. (4) as:

$$\frac{\partial I}{\partial t} = -\nabla^{\perp}(\Delta I)\cdot\nabla I, \qquad (5)$$

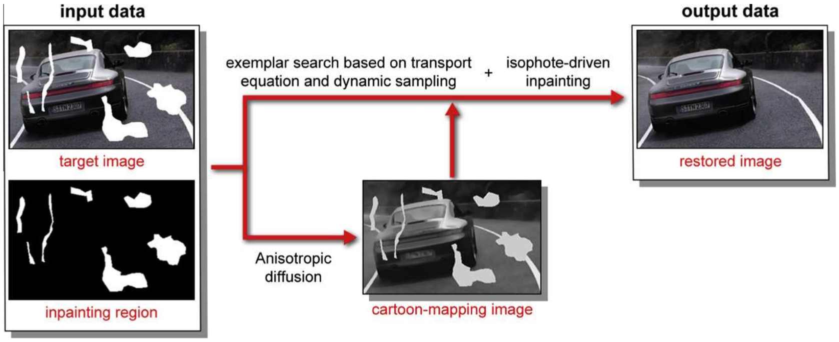

Fig. 1. Illustrative pipeline of our inpainting method.

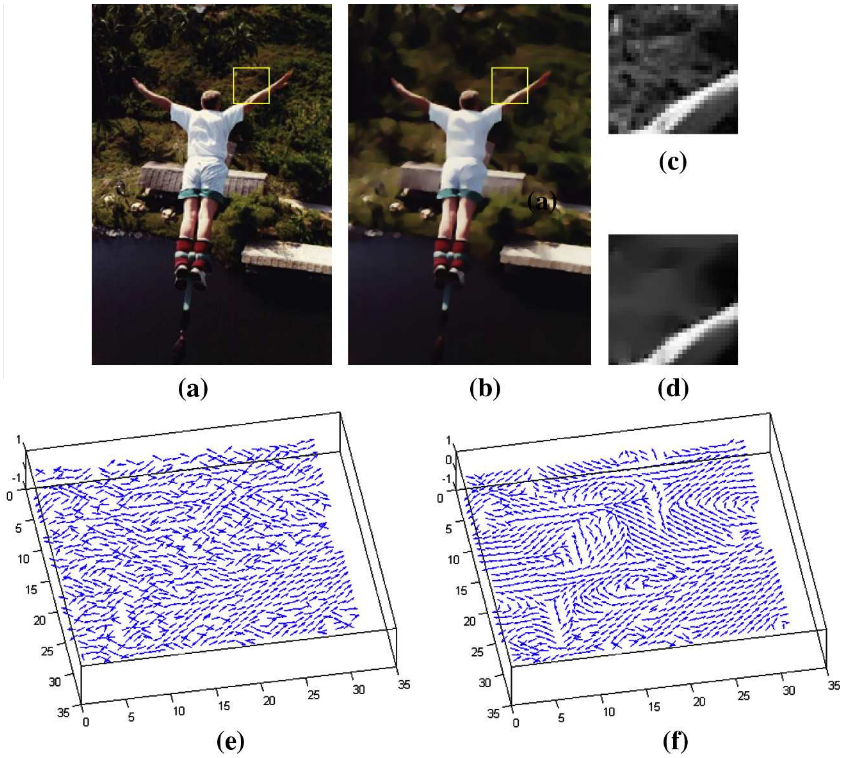

Fig. 2. Representation of the direction field in an illustrative image. (a) Original image, (b) cartoon (u component), (c)–(d) highlighted regions from (a) and (b), respectively, and (e)–(f) normalized field of directions of (c) (versor of $\nabla^{\perp}f$) and (d) (versor of $\nabla^{\perp}u$).

which makes clear the transport of I along the vector field $\nabla^{\perp}(\Delta I)$. This field provides the isophote orientation of ΔI and represents the well-known model of Marr and Hildreth (1980) for edge detection. Thus, the Laplacian operator Δ can be used for detecting isophotes as well as to transport information of interest.

Based on this discussion and the ideas presented in Bertalmío et al. (2000) and Bornemann and März (2007), we propose the following isophote detection measure:

$$\mathcal{R}(p) = |\nabla(\Delta u_p)\cdot \overrightarrow{d}_p|, \quad \overrightarrow{d_p} = \frac{\nabla^{\perp}u_p}{|\nabla^{\perp}u_p|}, \qquad (6)$$

where p is the pixel under analysis and $\overrightarrow{d_p}$ is given by the vector orthogonal to the gradient of u in p. $\mathcal{R}$ is called *relevance measure* and it aims to quantify the degree of relevance of geometric structures in the damaged image. The main difference between the expression defining (6) and Eq. (4) is that the direction field in (4) is not normalized. Therefore, it is reasonable to approximate the non-textured input image I from (4) by the cartoon component u in order to detect the isophotes of f. Furthermore, the choice of u instead of f in (6) can be justified by the following statements:

1. Regularity of the direction field $\frac{\nabla^{\perp}u}{|\nabla^{\perp}u|}$ when compared to $\frac{\nabla^{\perp}f}{|\nabla^{\perp}f|}$ (Fig. 2).
2. Application of the "smoothness propagation measure" Δu so that the geometry of interest (isophotes) is better elucidated due to lack of texture (Fig. 3(a)–(d)).

Fig. 3(e) shows a comparison between the relevance term R and the Criminisi et al. (2004) data term. Notice that Criminisi's data term introduces higher peaks on the graph, thus producing a jagged curve that reduces the accuracy of the inpainting priority. In contrast, our formulation presents a more stable behavior, that is, it is less affected by high-frequencies of the image. Moreover, our term is more effective in detecting only the isophote field, thus better preserving the evolution of linear structures.

Eq. (6) can be interpreted (except for the absolute value) as the variation of the smoothing measure Δu along $\overrightarrow{d}$,

$$\frac{\partial(\Delta u)}{\partial \overrightarrow{d}} = \nabla(\Delta u)\cdot \overrightarrow{d}, \quad \text{which implies } \mathcal{R}(p) = \left|\frac{\partial(\Delta u_p)}{\partial \overrightarrow{d_p}}\right|. \qquad (7)$$

Thus, the estimator (7) measures the variation of Δu_p towards the isophote field $\overrightarrow{d}$: if the derivative, in p, of Δu_p w.r.t. $\overrightarrow{d_p}$ is high, the relevance of p will also be high, otherwise, the pixel p will not play an important role in the restoration process.

Following Criminisi et al. (2004), we define a measure of balancing called *biased confidence term* $C(p)$:

$$\mathcal{C}(p) = \mathcal{C}_k(p) = \left(\frac{\sum_{q\in H_m(p)\cap(D-\Omega)} C(q)^{\frac{1}{k}}}{|H_m(p)|}\right)^{k}, \qquad (8)$$

where $|H_m(p)|$ denotes the size of a squared block $m \times m$ centered at pixel p, $C(q)$ is initialized as one and $k > 0$ is the bias parameter.

Finally, we propose the following measure to compute the filling priority:

$$\mathcal{P}(p) = \mathcal{R}(p)\cdot\mathcal{C}(p), \quad p \in \partial\Omega, \qquad (9)$$

where the terms $\mathcal{R}(p)$ and $\mathcal{C}(p)$ are given by Eqs. (6) and (8). Relevance term $\mathcal{R}$ computes the isophotes from the boundary $\partial\Omega$ while the biased confidence term $\mathcal{C}$ ensures the coherence of the image completion. Notice that for $k = 1$, Eq. (8) gives the same term proposed in Criminisi et al. (2004).

As pointed out in Cheng et al. (2005), regularizing the confidence term $\mathcal{C}$ is of paramount importance to guarantee the good quality of the inpainting. The regularized measure $\mathcal{C}$ allows us to tune the priority mechanism as follows: if $k \ll 1$ then $\mathcal{C}$ assigns more significance in balancing Eq. (9). Moderate significance

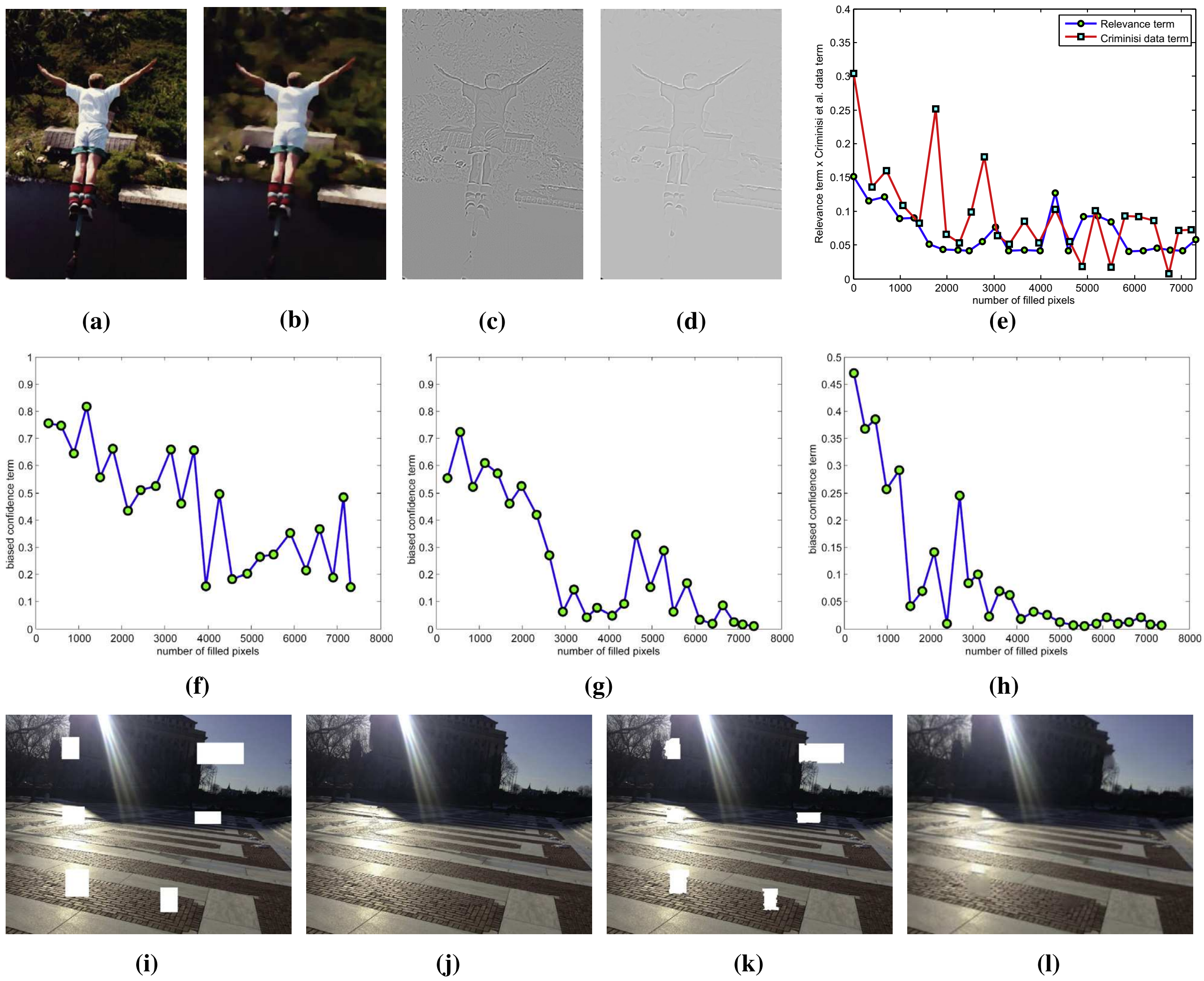

Fig. 3. (a)–(d) Original image, cartoon u and the result of the Laplacian operator Δ when applied to both images. (e) Comparison of the relevance term $\mathcal{R}$ against Criminisi et al. (2004) data term. (f)–(h) Plot of $\mathcal{C}(k = 0.5, 1.0, 1.5$, respectively) versus number of iterations when inpainting image (a). (i)–(l) Damaged image and the use of the dynamic sampling scheme (j) against methods Criminisi et al. (2004) (k) and Wexler et al. (2007) (l) in terms of computational effort when dealing with a huge photography (1000 × 1300).

occurs when $k \approx 1$ while $k \gg 1$ yields small significance. Fig. 3(f)–(h) illustrate the contribution of $\mathcal{C}$ when inpainting Fig. 3(a). In all cases, there is a decrease of intensities, however, the rate of decreasing changes significantly as k increases.

4. Dynamic processing of the sampling region

Classical approaches define the *inpainting source-sampling region* $\Lambda\Omega$ (region that provides the pixels to fill the damaged area) as the complement of the initial inpainting domain Ω, i.e., Ω^c (Efros and Leung, 1999; Cao et al., 2011; Criminisi et al., 2004; Cheng et al., 2005; Chen et al., 2007). However, instead of setting $\Lambda\Omega$ as a static region of the image, we employ a *dynamic sampling mechanism* that is tuned to each pixel to be filled. This methodology is less prone to fail, as it avoids to replicate pixels distant from the inpainting region in inappropriate places (e.g.: see Fig. 5 first row).

As in our approach the region $\Lambda\Omega$ is iteratively built for each pixel $p \in \partial\Omega$, we denote it by $\Lambda\Omega_p$. It is defined in terms of the valid region on $H_L(p)$ (an $L \times L$ block of pixels centered at pixel p). More specifically, for each pixel $p \in \partial\Omega$:

$$\Lambda\Omega_p = H_L(p) \cap \Omega^c. \tag{10}$$

Fig. 4(a) illustrates the construction. The advantage of this strategy is that it scans only the region $\Lambda\Omega_p$, whose dimensions are much smaller than those considered in previous approaches. Besides the computational gain, this strategy also prevents the transport of undesired information into the inpainting region. Notice that $\Lambda\Omega_p$ starts to be defined in a step-by-step manner for each pixel p. The filling process is indeed a recursive procedure. Fig. 3(i)–(l) compare the robustness of the proposed dynamic sampling against Criminisi et al. (2004) and Wexler et al. (2007) when dealing with a large photography (1000 × 1300, 51 k damaged pixels). Our technique (Fig. 3(j)) accomplishes the inpainting in a few minutes. In contrast, the algorithm proposed in Criminisi et al. (2004) filled only a small part of the image (Fig. 3(k)) by the time our approach completed the whole inpainting process. Considering the same amount of time, the technique proposed in Wexler et al. (2007) resulted in a coarse image (325 × 250) with bad quality inpainting (Fig. 3(l)).

5. Block-based pixel replication

After choosing the target pixel p using Eq. (9) and setting the corresponding sampling region $\Lambda\Omega_p$ by Eq. (10), the algorithm

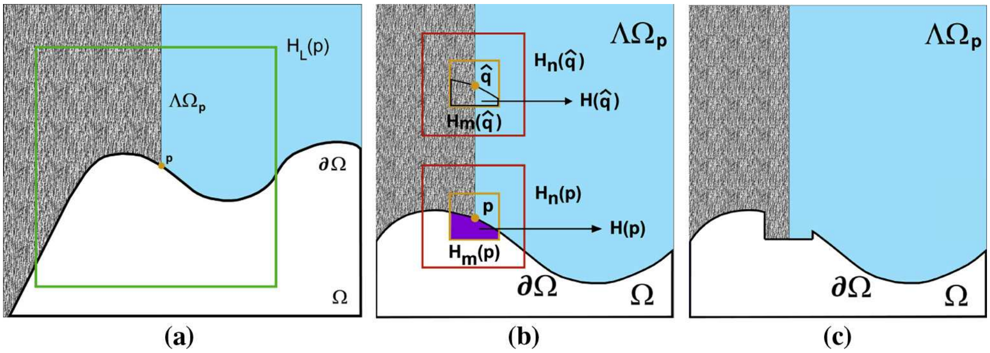

Fig. 4. Illustrative sketch of the dynamic sampling and the completion process. (a) $\Lambda\Omega_p$ (gray and blue parts) is the region inside $H_L(p)$ (green square) which provides candidate pixels. (b) Comparison between the content of patches $H_n(p)$ and $H_n(\widehat{q})$ (optimal patch) and (c) result after copying the information of interest. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

needs to locate the most suitable block of pixels (patch) inside $\Lambda\Omega_p$ to fill the neighborhood of p.

Given a pixel $p \in \partial\Omega$, the regions $H_L(p)$ and $\Lambda\Omega_p$ are initially settled (see Fig. 4(a)). Our algorithm uses a cartoon-based metric to compare the fixed patch $H_n(p)$ with all candidate patches $H_n(q)$ inside $\Lambda\Omega_p$. More precisely, the optimal patch $H_n(\widehat{q})$ is the one that minimizes the distance between $H_n(p)$ and $H_n(q)$ w.r.t. the metric. From $H_n(\widehat{q})$, a smaller patch $H_m(\widehat{q})$ is then selected and its valid pixels $H(\widehat{q})$ are used to fill up the neighborhood $H(p)$ of p (see Fig. 4(b) and (c)). Using the support regions $H_n(q)$ and $\Lambda\Omega_p$ to find appropriate pixels renders the search task faster and more robust.

In order to measure the distance (similarity) between $H_n(p)$ (target) and $H_n(q)$ (candidate) blocks, we use a weighted metric named *normalized root mean-square distance* (NRMSD). Let $\mathbf{p} = (f_{p_1}, f_{p_2}, \ldots, f_{p_k}), \mathbf{q} = (f_{q_1}, f_{q_2}, \ldots, f_{q_k})$ be column vectors in $\mathbb{R}^k, k < n^2$, containing the intensities of the pixels in $H_n(p)$ and the corresponding pixels in $H_n(q)$. The distance between $H_n(p)$ and $H_n(q)$ is measured as follows:

$$d(\mathbf{p},\mathbf{q}) = \frac{||\mathbf{p}-\mathbf{q}||_{\Delta U}}{\sqrt{||\mathbf{p}||^2_{\Delta U} + ||\mathbf{q}||^2_{\Delta U}}}, \tag{11}$$

where

$$\begin{aligned} ||\mathbf{p}-\mathbf{q}||_{\Delta U} &= \sqrt{(\mathbf{p}-\mathbf{q})^T \Delta U (\mathbf{p}-\mathbf{q})}, \\ ||\mathbf{p}||_{\Delta U} &= \sqrt{\mathbf{p}^T \Delta U \mathbf{p}}, \quad ||\mathbf{q}||_{\Delta U} = \sqrt{\mathbf{q}^T \Delta U \mathbf{q}}, \end{aligned} \tag{12}$$

with ΔU being a diagonal matrix defined by the Laplacian of the cartoon image u: $\Delta U_{ii} = \Delta u_{p_i}, p_i \in H_n(p) \cap \Lambda\Omega_p$. Metric (11) assigns higher weights for pixels located on the edges of the Laplacian of u. The weights of the pixels in (12) are defined from the Laplacian of u and build into the distance that compares blocks.

In mathematical terms, metric (11) holds many attractive properties. The term $||\cdot||_{\Delta U}$ is an Euclidian metric induced by an inner product that encodes data from ΔU. In fact, we can write $||\mathbf{x}||_{\Delta U} = \sqrt{\langle \mathbf{x}, \mathbf{x} \rangle_{\Delta U}} > 0, \; \forall \mathbf{x} \neq 0 \in \mathbb{R}^k$, so that the following mathematical properties are guaranteed:

1. $det(\Delta U) = \Delta U_{11}, \Delta U_{22}, \ldots, \Delta U_{kk} > 0$ (matrix ΔU is derived from an image);
2. ΔU is symmetric (diagonal matrix).

Since $||\cdot||_{\Delta U}$ is an Euclidian metric, Eq. (11) also defines a metric in $\mathbb{R}^k$. The advantages of using the metric (11) is that it can employ information provided from the Laplacian operator while still measuring the structural similarity between patches (see Brunet et al., 2012).

6. Experimental results

This section presents some results obtained by our technique. A comparative study with state-of-the-art methods is also provided. In our experiments we fix $N = 50$ and let σ to change in Eq. (2) (for further details, see Barcelos et al. (2003, 2005)). In Eq. (8), we also set $k = 0.5$. Dimensions m and n in (11) are free parameters while L in (10) has been defined as a function of $n: \; L = L(n) = 4n + 1$ (choice made after exhaustive tests) except in Fig. 7, where $L = 71$. Outputs were obtained in MATLAB on a 1.80 GHz AMD with 2 GB of RAM without any MATLAB-MEX optimization scheme.

6.1 (a) Pattern transfering and contribution of C, R and P The first row in Fig. 5 shows some inpainting results for a real-world image. Without using the dynamic sampling scheme one can notice that parts of the region shown in green in the left most image were transferred to the inpainted domain (highlighted in red in the second image from left). Although visually pleasant, the result is not realistic. Moreover, the computational cost was high (six times bigger than the proposed scheme). From the third to the fifth images in the first row of Fig. 5, one can see the results of our approach when considering only measures $\mathcal{C}$ (8), $\mathcal{R}$ (6) and the proposed filling order mechanism $\mathcal{P}$ (9) to compute the inpainting priority for a fixed number of iterations. In all cases, it is clear what is the contribution of each term into the pipeline of restoration. Finally, the fully restored image ($\sigma = 40, m = 7, n = 15$) is presented in the right most image.

6.1 (b) Effectiveness of the cartoon-driven filling order mechanism Second row in Fig. 5 depicts our method step-by-step when applied to an image with both textures and smooth regions, which have been intentionally damaged. As shown in the third and fourth images from left to right, the regions highlighted in green and red were first considered due to proposed priority mechanism, computed from the cartoon component (second image in second row). In addition, the texture was fully reconstructed and details were nicely preserved, as shown in the right most image ($\sigma = 30, m = 9, n = 9$).

6.1 (c) Highly-detailed texture inpainting We present an example of object removal where the target-object is inserted in a region containing regular texture patterns. The left most image on the bottom row of Fig. 5 shows the original image while the second image from left shows the object to be removed. The third image presents the result of our method ($\sigma = 15, m = 9, n = 11$) and right most image is a "zoom" in the inpainted region. One can clearly see that the resulting image was accurately inpainted. Notice that the dynamic sampling mechanism successfully contributed to reach the good result, since only pixels nearby the inpainting domain were used in the completion process.

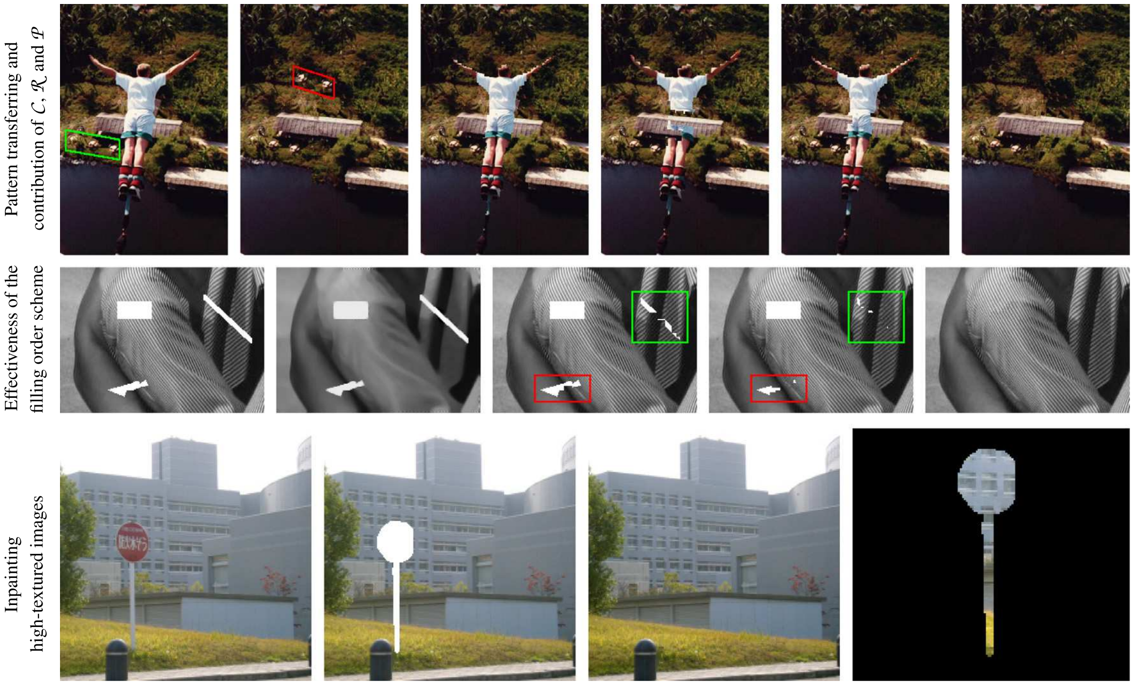
Pattern transferring and contribution of $\mathcal{C}$, $\mathcal{R}$ and $\mathcal{P}$
Effectiveness of the filling order scheme
Inpainting high-textured images

Fig. 5. Particular properties of the proposed method. First row: pattern transferring and individual contribution of each term, second row: step-by-step of our method and third row: exploiting high-textured images.

6.2 Comparative results In this section we provide comparisons against the state-of-the-art methods describe in Efros and Leung (1999), Criminisi et al. (2004), Wexler et al. (2007), Guleryuz (2006), Elad et al. (2005), Fadili et al. (2009), Xu and Sun (2010), and Li (2011). Parameters of each method we compare against were tuned according to original papers author's implementation, their personal web sites, and exhaustive tests towards obtaining the best results for each technique.

6.2 (a) Qualitative comparison From left-to-right in Fig. 6 we present the input images and the results obtained by Efros and Leung (1999),[1] Criminisi et al. (2004),[2] Wexler et al. (2007)[3] and our method,[4] respectively. The input image in the first row of Fig. 6 is a synthetic image made up of six different groups of textures with a large amount of unfilled pixels. It is clear that our technique outperforms other methods in terms of properly filling the inpainting regions. Left most image in the second row is an image where structural objects are missing. Efros's, Criminisi's and Wexler's methods introduce artifacts in the inpainted image. Our method, however, generates a more pleasant result (although not perfect) when reconstructing edges and structural objects. Third row in Fig. 6 brings an example of a photograph where important details are damaged. One can see that the proposed approach outperforms other methods, as no artifact are introduced in the "iris" and above the eyebrow. The monkey image in the fourth row of Fig. 6 depicts a photograph with large inpainting regions placed on complex structures of the image. Wexler's method blurs the image while Efros's and Criminisi's methods introduce artifacts. Our technique, in contrast, has restored the details quite well (compare "the coat of the Monkey"). In the fifth row we investigate the problem of object removal. Notice that Criminisi's method introduces many artifacts, creating unpleasant visual effect. Efros's and Wexler's techniques blur the inpainted region. In contrast, our approach performs well avoiding artifacts and blurring effects. The row before the last one is another example of object removal in a non-textured image. Efros's and Criminisi's methods present unpleasant results while Wexler's and our technique produce the better outcomes. In fact, Wexler's method performed quite well due to its intrinsic smoothing transition of the colors. The challenge in inpainting the image shown on the bottom row of Fig. 6 is to recover parts of the fence hidden by the statue while maintaining the natural aspect of the photography. Criminisi's and Wexler's methods result in non-realistic images. Ours and Efros's method present a more convincing reconstruction, but, it is easy to see that our method introduced less artifacts.

6.2 (b) Quantitative comparison with sparsity-based inpainting Our last comparative evaluation deals with missing block completion. We perform both qualitative and quantitative comparison against various methods that rely on sparsity properties of the image. From left-to-right in Fig. 7 we show the input images and the results produced by the Guleryuz (2006), Elad et al. (2005), Fadili et al. (2009), Xu and Sun (2010) and Li (2011) methods. All examples are tricky to handle due to the predominance of structures and textures around the region to be filled. Notice that the Guleryuz and Fadili et al. algorithms produce blurring effect on the images while the Elad et al. method produces some artifacts in the outputs. The results reached by Xu and Sun as well as by the Li method are visually better than those two, but they still suffer from smoothing effect. In contrast, our technique leads to non-blurred completion and accurately recovers isophotes and pure texture regions, producing a pleasant and more realistic result.

For sake of quantitative comparison, PSNR (Peak Signal-to-Noise Ratio) between the recovered and original images from Fig. 7 were computed (see Table 1). Notice from the average PSNR in the last row of Table 1 that our approach clearly outperforms others.

Computational cost The computation cost of our methodology is considerable lower than the other techniques we compared against. For instance, for the first experiment in Fig. 6 (18186

[1] Window size: $(5,5,7,5,5,5,5)$.

[2] Window size: 9.

[3] L: $(4,4,2,3,3,3,3)$, window size: $(3,5,5,5,7,7,7)$ and n.iterations:15.

[4] $\sigma:\ (50,30,30,15,15,35,35)$, m: 5 and n: $(9,9,7,15,11,9,9)$.

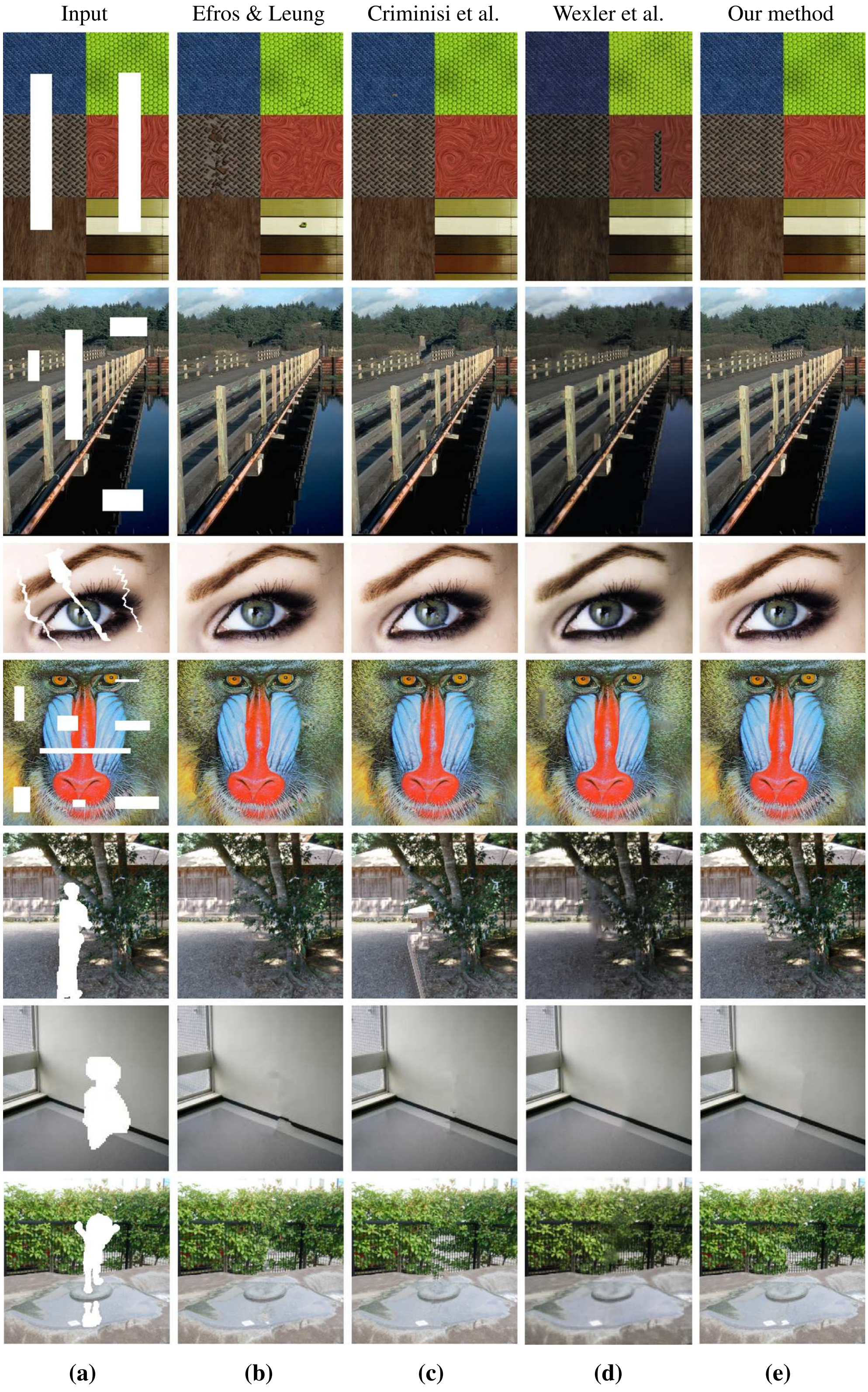

Fig. 6. Comparison with other inpainting methods. (a) Input images, (b) inpainted by Efros and Leung (1999), (c) Criminisi et al. (2004), (d) Wexler et al. (2007) and (e) our method.

damaged pixels), our approach took 2 min while Efros's algorithm spent 30 min to complete the processing. Considering the average time to recover all images in Fig. 7, our technique took 44 s versus 187 min of the Li's method. The difference of our technique and the evaluated methods is even greater when we compare the algorithms taking large inpainting domains such as in the fourth row of Fig. 6 (25,025 damaged pixels): our technique took 7 min against 36 min of the Criminisi et al. method and a few hours of the Wexler

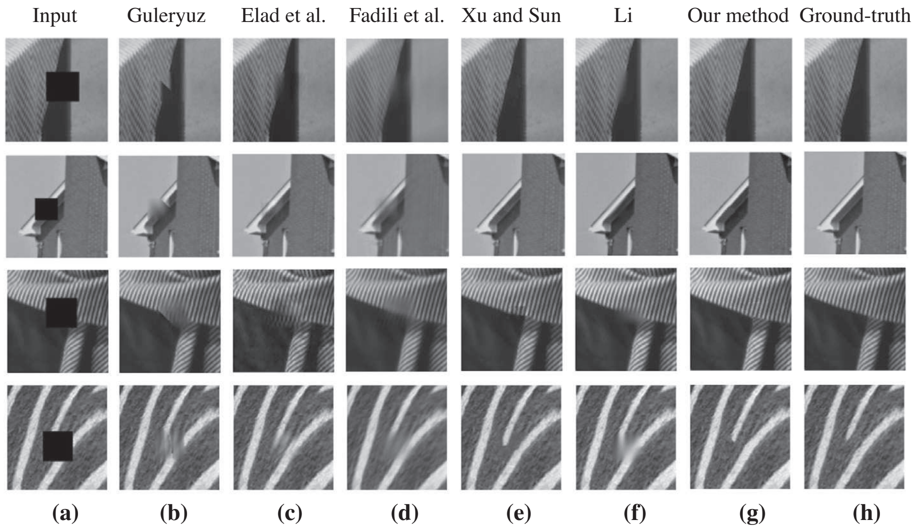

Fig. 7. Comparison with sparse representation-based inpainting methods. (a) Input images (“Tissue”, “Eaves”, “Barbara’s part” and “Fur”), (b) inpainted by Guleryuz (2006) (with DCT of 32×32), (c)–(d) Elad et al. (2005) and Fadili et al. (2009) (with dictionary of texture/cartoon layers using DCT of 32×32 and 5-resolution curvelet transform (Candes et al., 2006), respectively), (e) Xu and Sun (2010) (with patch size of $7 \times 7, N(p)$ size of 51×51 and $\epsilon = N = 25$), (f) Li (2011) ($B = 31, k = 30, d = 4, n_{max} = 50, N_{max} = 3$), (g) our method ($(\sigma, m, n) = (15, 13, 15), (10, 3, 7), (15, 7, 11), (15, 5, 7)$ from top to down) and (h) the ground-truth images.

Table 1
Quantitative evaluation using PSNR (in dB) for all comparative images from Fig. 7.

Image	Guleryuz (2006)	Elad et al. (2005)	Fadili et al. (2009)	Xu and Sun (2010)	Li (2011)	Our method
Tissue	20.41	22.43	22.16	23.53	22.21	25.02
Eaves	16.15	22.85	17.86	28.45	26.77	29.30
Barbara’s part	18.39	19.20	17.85	23.07	23.61	24.43
Fur	16.46	19.49	20.67	20.87	18.43	21.55
Average	17.85	20.99	19.64	23.98	22.75	25.08

et al. technique. In fact, the gain in performance of our method is due to the use of dynamic sampling strategy as depicted in Fig. 3(i)–(l).

7. Discussion and limitation

The proposed cartoon-driven filling order and dynamic sampling mechanisms turn out to be quite efficient for image inpainting. Moreover, the proposed cartoon-based metric used to compute the filling order and pixel similarity have resulted in accurate results, outperforming metrics purely based on the target image. The comparisons presented in Section 6 attest the effectiveness of our approach in terms of accuracy as well as computational cost. The good performance when dealing with highly textured images (e.g. Figs. 5 and 7), fine details (e.g. Fig. 5 second row, Fig. 6 third row) and large inpainting domains (e.g. Figs. 3(i), 5 first row, Fig. 6 first and fourth rows) show the robustness and flexibility of the proposed method.

Despite the good properties and results, our method also has limitations. For example, it demands several parameters that have to be tuned. Optimizing those parameters is an issue we will investigate in future work. Unsatisfactory results can be obtained when inpainting non-textured images with color variation, being this another limitation of our method.

8. Conclusion and future work

This work presents a new inpainting technique based on copy-and-paste blocks to recover real and synthetic images containing a large variety of textures and structural objects. The concepts of image decomposition and transport equation were revisited so as to provide a robust mechanism to define the filling order and determine the similarity between blocks of pixels during the inpainting process. The proposed sampling mechanism also turned out to be quite effective to reduce computational costs.

Real and synthetic images of different complexity levels were evaluated with the purpose of assessing the efficiency of the proposed approach. Our approach reaches a good trade-off between visual quality and low computational cost.

Acknowledgments

The authors would like to thank the anonymous reviewers for their useful and constructive comments. This research has been supported by FAPESP-Brazil, CNPq-Brazil and CAPES-Brazil.

References

Ashikhmin, M., 2001. Synthesizing natural textures. In: ACM Symposium on Interactive 3D Graphics (I3D), pp. 217–226.
Aujol, J.-F., Ladjal, S., Masnou, S., 2010. Exemplar-based inpainting from a variational point of view. SIAM J. Math. Anal. 42, 1246–1285.
Barcelos, C.A.Z., Boaventura, M., Silva Jr., E.C., 2003. A well balanced flow equation for noise removal and edge detection. IEEE Trans. Image Process. 12, 751–763.
Barcelos, C.A.Z., Boaventura, M., Silva Jr., E.C., 2005. Edge detection and noise removal by use of a PDE with automatic selection of parameters. Comput. Appl. Math. 24, 131–150.
Bertalmío, M., Sapiro, G., Caselles, V., Ballester, C., 2000. Image inpainting. In: Annual Conference on Computer Graphics (SIGGRAPH), pp. 217–226.
Bertalmío, M., Vese, L.A., Sapiro, G., Osher, S., 2003. Simultaneous structure and texture image inpainting. IEEE Trans. Image Process. 12, 882–889.
Bornemann, F., März, T., 2007. Fast image inpainting based on coherence transport. J. Math. Imaging Vis. 28, 259–278.
Brunet, D., Vrscay, E.R., Wang, Z., 2012. On the mathematical properties of the structural similarity index. IEEE Trans. Image Process., 1–10.
Bugeau, A., Bertalmío, M., 2009. Combining Texture Synthesis and Diffusion for Image Inpainting. In: VISAPP (1), pp. 26–33.

Bugeau, A., Bertalmío, M., Caselles, V., Sapiro, G., 2010. A comprehensive framework for image inpainting. IEEE Trans. Image Proces. 19, 2634–2645.

Burger, M., He, L., Schönlieb, C.-B., 2009. Cahn-hilliard inpainting and a generalization for grayvalue images. SIAM J. Imaging Sci. 2, 1129–1167.

Cai, J.-F., Chan, R., Shen, Z., 2008. A framelet-based image inpainting algorithm. Appl. Comput. Harmon. Anal. 24, 131–149.

Calvetti, D., Sgallari, F., Somersalo, E., 2006. Image inpainting with structural bootstrap priors. Image Vision Comput. 24, 782–793.

Candes, E., Demanet, L., Donoho, D., Ying, L., 2006. Fast discrete curvelet transforms. SIAM J. Multiscale Model. Simul. 5, 861–899.

Cao, F., Gousseau, Y., Masnou, S., Pérez, P., 2011. Geometrically guided exemplar-based inpainting. SIAM J. Imaging Sci. 4, 1143–1179.

Casaca, W.C.O., Boaventura, M., 2010. A decomposition and noise removal method combining diffusion equation and wave atoms for textured images. Math. Problems Eng. 2010, 1–21.

Chan, T., Shen, J., 2001. Nontexture inpainting by curvature-driven diffusion (CDD). J. Visual Commun. Image Represent. 12, 436–449.

Chen, Q., Zhang, Y., Liu, Y., 2007. Image inpainting with improved exemplar-based approach. In: Proc. of the Int. Conf. on Multimedia Content Analysis and Mining, MCAM'07, pp. 242–251.

Cheng, W.-H., Hsieh, C.-W., Lin, S.-K., Wang, C.-W., Wu, J.-L., 2005. Robust algorithm for exemplar-based image inpainting. In: Proc. of the International Conference on Computer Graphics, Imaging and Visualization, pp. 64–69.

Criminisi, A., Peréz, P., Toyama, K., 2004. Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Image Process. 13, 1200–1212.

Demanet, L., Ying, L., 2007. Wave atoms and sparsity of oscillatory patterns. Appl. Comput. Harmon. Anal. 23, 368–387.

Demanet, L., Song, B., Chan, T., 2003. Image inpainting by correspondence maps: a deterministic approach. In: Proc. VLSM, pp. 1–8.

Efros, A.A., Freeman, W.T., 2001. Image quilting for texture synthesis and transfer. In: ACM SIGGRAPH '01, pp. 341–346.

Efros, A.A., Leung, T.K., 1999. Texture synthesis by non-parametric sampling. In: IEEE Int. Conf. Computer Vision, pp. 1033–1038.

Elad, M., Starck, J.-L., Querre, P., Donoho, D., 2005. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). Appl. Comput. Harmon. Anal. 19, 340–358.

Fadili, M., Starck, J.-L., Murtagh, F., 2009. Inpainting and zooming using sparse representations. Comput. J. 52 (1), 64–79.

Grossauer, H., 2004. A combined PDE and texture synthesis approach to inpainting. Euro. Conf. Comput. Vision 4, 214–224.

Guleryuz, O.G., 2006. Nonlinear approximation based image recovery using adaptive sparse reconstructions and iterated denoising-part ii: adaptive algorithms. IEEE Trans. Image Process. 15 (3), 555–571.

Harrison, P., 2001. A non-hierarchical procedure for re-synthesis of complex textures. In: WSCG '2001 Conference Proceedings, pp. 190–197.

Hays, J., Efros, A.A., 2007. Scene completion using millions of photographs. ACM Trans. on Graph. 26(3).

Kanizsa, G., 1979. Organization in Vision: Essays on Gestalt Perception, Praeger.

Kawai, N., Sato, T., Yokoya, N., 2009. Image inpainting considering brightness change and spatial locality of textures and its evaluation. In: Proceedings of the 3rd Pacific Rim Symp. on Adv. in Img. and Video Technology, PSIVT '09, pp. 271–282.

Komodakis, N., Tziritas, G., 2007. Image completion using efficient belief propagation via priority scheduling and dynamic pruning. IEEE Trans. Image Process. 16, 2649–2661.

Kwok, T.-H., Wang, C.C., 2009. Interactive image inpainting using dct based exemplar matching, in: Proc. of the 5th International Symposium on Advances in Visual Computing: Part II, ISVC '09. Springer-Verlag, pp. 709–718.

Li, X., 2011. Image recovery via hybrid sparse representations: a deterministic annealing approach. IEEE J. Sel. Top. Signal Process. 5 (5), 953–962.

Li, S., Zhao, M., 2011. Image inpainting with salient structure completion and texture propagation. Pattern Recogn. Lett. 32 (9), 1256–1266.

Li, H., Wang, S., Zhang, W., Wu, M., 2010. Image inpainting based on scene transform and color transfer. Pattern Recogn. Lett. 31 (7), 582–592.

Liu, Y., Caselles, V., 2013. Exemplar based image inapinting using multiscale graph cuts. IEEE Trans. Image Process. 22, 1699–1711.

Mallat, S., 2008. A Wavelet Tour of Signal Processing: The Sparse Way, third ed. Academic Press.

Marr, D., Hildreth, E., 1980. Theory of edge detection. In: Proc. R. Soc. Lond. B, pp. 187–217.

Meyer, Y., 2002. Oscillating Patterns in Image Processing and Nonlinear Evolution Equations, vol 22. 1st ed., University Lectures Series, American Mathematical Society.

Paragios, N., Chen, Y., Faugeras, O., 2005. Handbook of Mathematical Models in Computer Vision. Springer-Verlag Inc., New York.

Shen, J., Chan, T.F., 2002. Mathematical models for local nontexture inpainting. SIAM J. Appl. Math. 62, 1019–1043.

Sun, J., Yuan, L., Jia, J., Shum, H.-Y., 2005. Image completion with structure propagation. In: ACM SIGGRAPH 2005, SIGGRAPH '05, pp. 861–868.

Tschumperlé, D., Deriche, R., 2005. Vector-valued image regularization with PDEs: a common framework for different applications. IEEE Trans. Pattern Anal. Mach. Intell. 27, 506–517.

Vese, L., Osher, S., 2003. Modeling textures with total variation minimization and oscillating patters in image processing. J. Sci. Comput. 19, 553–572.

Vese, L., Osher, S., 2006. Color texture modeling and color image decomposition in a variational-PDE approach. In: Proc. of the Eighth Int. Symp. on Symb. and Numeric Algorithms for Sci. Computing. IEEE Computer Society, pp. 103–110.

Wei, L.-Y., 2002. Texture Synthesis by Fixed Neighborhood Searching. Ph.D. Thesis. Stanford University, Stanford, CA, USA.

Wen-Ze, S., Zhi-Hui, W., 2008. Edge-and-corner preserving regularization for image interpolation and reconstruction. Image Vision Comput. 26, 1591–1606.

Wexler, Y., Shechtman, E., Irani, M., 2007. Space-time completion of video. IEEE Trans. Pattern Anal. Mach. Intel. 29, 463–476.

Xu, Z., Sun, J., 2010. Image inpainting by patch propagation using patch sparsity. IEEE Trans. Image Process. 19 (5), 1153–1165.