Spatial patch blending for artefact reduction in pattern-based inpainting techniques

Maxime Daisy, David Tschumperlé, and Olivier Lézoray

GREYC Laboratory (CNRS UMR 6072), Image Team, 6 Bd Marechal Juin, 14050 Caen/France,
{maxime.daisy,david.tschumperle,olivier.lezoray}@ensicaen.fr

Abstract. Patch-based (or "pattern-based") inpainting, is a popular processing technique aiming at reconstructing missing regions in images, by iteratively duplicating blocks of known image data (patches) inside the area to fill in. This kind of method is particularly effective to process wide image areas, thanks to its ability to reconstruct textured data. Nevertheless, "pathological" geometric configurations often happen, leading to visible reconstruction artifacts on inpainted images, whathever the chosen pattern-based inpainting algorithm. In this paper we focus on these problematic cases and propose a generic *spatial-blending technique* that can be adapted to any type of patch-based inpainting methods in order to reduce theses problematic artefacts.

Keywords: inpainting, pattern, patch, blending

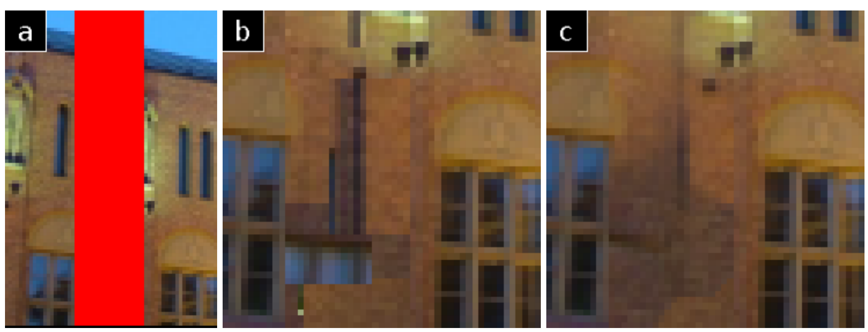


Fig. 1: Illustration of our contribution : (a) masked image to be reconstructed, (b) zoom on inpainted result by classical copy-paste of patches, (c) zoom on inpainted result with our patch blending algorithm, proposed in this paper.

1 Context and current issues

Image inppainting is the technique that is used to reconstruct the missing parts in an image while keeping the geometrical consistency in the reconstructed portion as much as possible. A good review of inpainting methods can be found in [1] where two main kinds of techniques are described:

– "Geometry-based" methods, introduced by Masnou and Morel [2] in their seminal work on area disocclusion by level-line completion, solve the inpainting problem by

applying partial derivative equations and finiding a function that minimizes an energy that expresses the inpainting problem. These methods allow to locally extend the geometry of the image structures at the boundaries of the inpainting domain. Nevertheless, no texture regeneration is possible with these techniques, and spatial inconsistencies appear between known parts and reconstructed ones, especially for real images.
– "Pattern-oriented" methods, introduced by Efros and Leung [3] in their work on texture extrapolation, are based on copy and paste of patches from known parts of the image into the area to be recovered. These algorithms generate a kind of patchwork of image pieces by iteratively choosing the most similar patches to those living on the boundaries of the inpainting domain. Results given by these methods are particularly interesting for wide and textured area reconstructed.

Some variations intend to mix these two approaches to either create hybrid methods [4,5,6], or to to average several patches [7,8] to compute the final patch to use for the reconstruction. In any case, there are always local configurations that do not satisfy the selected similarity criteria. Copying and pasting, or averaging such kind of patches produces some visual artifacts appearing as block effects or discontinuities in the image. These configurations are actually not so uncommon: an inpainting mask often covers regions where transitions in texture or luminosity should occur and there are no ways a single patch taken around can fit perfectly for the reconstruction.

In this paper we introduce an original approach based on *spatial patch blending*, to reduce these kinds of artifacts. The advantages of our proposed algorithm are twofold: First, it can be used to post-process any type of pattern-based inpainting result, and second, it is able to reconstruct patches that are *smart combinations* of those located in the known area of the image. Moreover, we propose a way to perform this patch blending only on locations where inpainting artifacts occur by automatically detecting the most visible artefact points. The simultaneous application of these two contributions (spatial patch blending and inpainting artifacts detection) is illustrated by several inpainting results where the visual artifacts are clearly reduced compared to a classical patch-based inpainting approach.

This paper is organized as follows. Second section exposes the principle of our patch blending technique for patch-based image inpainting post processing. Third section shows how artifacts are detected from patch-based image inpainting results. Last section concludes with some commented results.

2 A patch blending algorithm for inpainting

This section shows how our spatial patch blending algorithm works, along with the modifications that have to be performed for patch-based inpainting methods to provide the data used in our blending algorithm.

Let $I : \Gamma \mapsto \mathbb{R}_+^3$ be a color image of which the missing pixels are defined over a domain $\Omega \subset \Gamma$ (inpainting mask). Let ψ_p denote a square-sized ($N \times N$ with N odd) image patch centered at $p \in \Gamma$. The ith color component of a patch ψ_p will be denoted by $\psi_p^i \in \mathbb{R}^{N \times N}$. Most of the pattern-oriented inpainting algorithms are based on two steps:

1. the research, for each $p \in \mathit{\Omega}$, of the "best" patch ψ_q (with $\psi_q \subset \mathit{\Gamma} \setminus \mathit{\Omega}$),
2. the copy of the patch ψ_q on $\psi_p \cap \mathit{\Omega}$ inside the image I to be reconstructed.

Based on this copy/paste principle, the content of $\mathit{\Omega}$ is filled in an iterative and globally concentric way (however, some points of interest can be processed in priority).

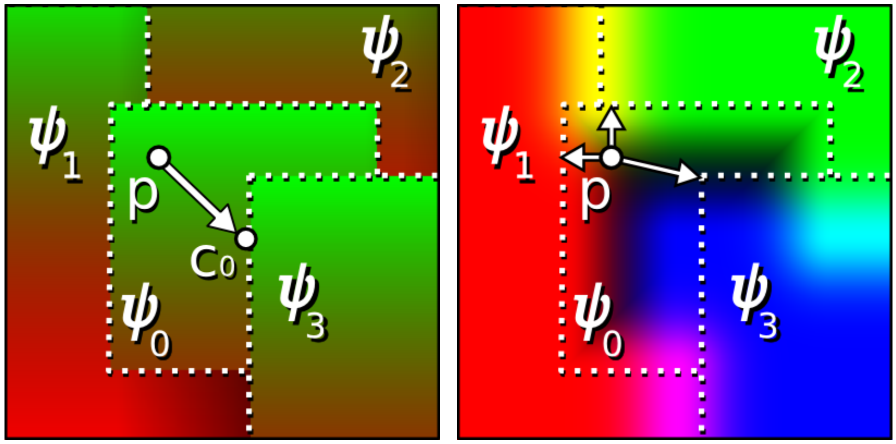
Our technique consists in modifying this kind of inpainting algorithm to be able to post-process the inpainting result with patch blending method in order to visually improve the quality of the recovered image I. This modification requires to save both the map $\mathcal{U} : \mathit{\Gamma} \mapsto \mathit{\Gamma}$ of the locations of the centers of the original patches, and the map $\mathcal{S} : \mathit{\Omega} \mapsto \mathbb{R}^2$ of the offsets between the reconstructed points and the centers of the partially copied patches. The connected component labelization $\mathcal{L}$ of $\mathcal{U}$ defines the partition of the patches pieces in the known parts of the image that were stuck back in $\mathit{\Omega}$. Using the following data sets : $I, \mathcal{U}$, $\mathcal{S}$ and $\mathcal{L}$, we are able to generate an image $J : \mathit{\Gamma} \mapsto \mathbb{R}^3_+$ of which the patches of I, used to fill $\mathit{\Omega}$, spatially blend one to another. In practice, for each point $p \in \mathit{\Omega}$, this blending is realized with a set $\mathit{\Psi}_p$ of patches $\{\psi_{p_1}, \ldots, \psi_{p_n}\}$, obtained from a neighborhood $\mathcal{V}(p)$ around p, and that contains p. This set is built from a full search of all the centers of the different patches that were copied in $\mathcal{V}(p)$. Then, the blending is computed for the pixel p by the spatial merging of all the patches contained in $\mathit{\Psi}_p$ as follows for each color component i:

$$J^i(p) = \frac{\sum\limits_{\psi_q \in \mathit{\Psi}_p} w(q,p)\ \psi_q^i(p-q)}{\varepsilon + \sum\limits_{\psi_q \in \mathit{\Psi}_p} w(q,p)} \tag{1}$$

with $\varepsilon \in \mathbb{R}$ close to 0 used to prevent a potential division by zero. The quantity $\psi_q^i(p-q)$ represents the pixel value at the coordinates $((N/2, N/2)^T + (p-q))$ in ψ_q^i. The gaussian weights $w(q,p) = e^{-\frac{d(q,p)^2}{\sigma^2}}$ give more importance in the blending computation to the patches of $\mathit{\Psi}_p$, that are the spatially closest to the point p. The variance σ is a parameter related to the amplitude of the spatial merging. The function $d(p,q)$ defines the minimal spatial distance between the point p and the sub-domain of the patch that was used for the reconstruction of q.

$$d(q,\ p) = \min_{q' \in \mathcal{V}(p)} \|q' - p\| \quad \text{where} \quad \mathcal{L}(q') = \mathcal{L}(q) \tag{2}$$

The usage of the distance $d(p,q)$ allows to generate a spatial blending orthogonally oriented according to the boundaries of the patches in the neighborhood of p. Fig. 2 illustrates a synthetical situation where we want to apply this blending technique. In the latter figure, p is the point where the blending has to be computed, and $\{\psi_0, \ldots, \psi_3\}$ are the patches contained in $\mathit{\Psi}_p$. The x and y component of the vectors of $\mathcal{S}$ are respectively represented by the red and green components in the first figure. The point c_0 is the center of the patch ψ_0 and one has $\mathcal{S}(p) = c_0 - p$. In Fig. 2(b), the arrows represent the minimal distances from p to each neighbor patch. The weights w related to ψ_1, ψ_2, ψ_3 are respectively represented by the lighting intensities in the red, green and blue channels (with $w(p,q) = 0$ for $q \in \psi_0$). Algorithm. 1. summarizes the main steps of our patch blending process.



(a) Representation of the map $\mathcal{S}$ of the offsets with the patch centers.

(b) Representation of the weights w used to compute (1) for every point of ψ_0.

Fig. 2: Illustration of both $\mathcal{S}$ and w quantities computed for the patch blending.

Our spatial patch blending technique uses all the patches from which a point $p \in \Omega$ could have been reconstructed during the inpainting process if the filling order had been different from those really used. Intuitively this comes to keep a list of candidate patches (rather than the only one of the basic copy/paste process) for each $p \in \Omega$ reconstructed during a patch-based inpainting. It is very different from an inpainting process that would average the K best patches (like in [7]) to reconstruct each point p since here the blending uses the geometry of the piece of patch copied at p and also the spatial configuration of the neighbour patches. One can notice that the blending is computed with a constant amplitude σ. Actually, all the areas of the inpainted image do not contain breaks of the same strength, so it would be naturally desirable to be able to locally vary σ. Hence, the third section explains our second contribution, the adaptation of σ to the local artifacts strength.

Algorithm 1: Spatial patch blending for inpainting

```
for p ∈ Ω do
    var centers = list(∅);
    var distances = list(∅);
    for q ∈ V(p) do
        var c = q + S(q) ;
        if (c ∉ centers) ∧ (p ∈ ψ_c) then
            insert(centers, c) ;
            insert(distances, d(c, p)) ;            // see (2)
    J(p) = blend(p, centers, distances) ;           // see (1)
```

3 Automatic detection of inpainting artifacts

In this section, we explain a method to automatically detect artifacts produced by patch-based inpainting algorithms. This method uses additional data provided by modifying patch-based inpainting methods.

Most of the visual artifacts due to a reconstruction mainly appear when two patches with not enough similarity were copied close to each other inside Ω. In practice, the point $p \in \Omega$ where artifacts appear are inside areas verifying these two conditions :

1. $\|\nabla I(p)\|$ is high, and so indicates strong spatial discontinuities in the image values.
2. Patches pasted inside the neighborhood of p come from different and far locations producing discontinuities inside $\mathcal{U}$.

Therefore, our proposition is to track down the set $\mathcal{E}$ of these artifact points (also named *break points*) as the set of points verifying the condition $\mathcal{R}(p) > \tau$ where :

$$\forall p \in \Omega, \quad \mathcal{R}(p) = \frac{\|\nabla I(p)\| \; \cdot \; |\operatorname{div}(\mathcal{U})(p)|}{\alpha} \tag{3}$$

and $\alpha = (\max_{q \in I} \|\nabla I(q)\|) \times (\max_{q \in I} |\operatorname{div}(\mathcal{U}(q))|)$ is a normalization factor allowing the user parameter τ, related to the detected break points density, to be chosen within $[0, 1]$. $\mathcal{R}(p)$ is used to locally estimate the strength of the geometrical break due to the reconstruction since $|\operatorname{div}(\mathcal{U})|$ determines whether two patches stuck next to each other come from close locations (low $|\operatorname{div}(\mathcal{U})|$), or far locations (high $|\operatorname{div}(\mathcal{U})|$). Consequently, if $\mathcal{R}(p)$ is high, the point p fulfills both 1. and 2. and is more likely to locate an artifact inside the image I.

By associating to each point $p \in \mathcal{E}$ a Gaussian function the variance of which depends on $\mathcal{R}(p)$, a spatial map $\sigma : \Gamma \mapsto \mathbb{R}_+$ of the local blending amplitude can be defined as follows:

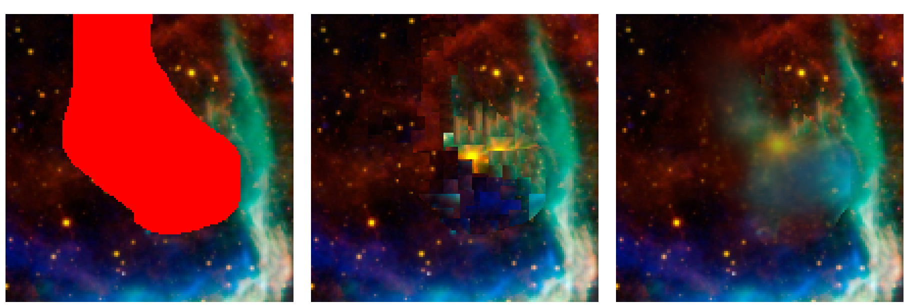
$$\forall p \in \Gamma, \quad \sigma(p) = \rho \times \frac{\sum_{r \in \mathcal{E}} w_b(p, r)}{\max_{q \in \Gamma} \sum_{r \in \mathcal{E}} w_b(q, r)} \tag{4}$$

where $w_b(p, r) = e^{-\frac{\|p-r\|^2}{(3\rho\mathcal{R}(r))^2}}$ and ρ is a user parameter defining the maximal width of the spatial patch blending inside J. Using $\sigma = \sigma(p)$ in the patch blending equation (1), we are able to locally change the blending amplitude according to the presence (also the strength) or not of the artifacts. For the sake of performance, the blending is made only inside a subset $\Omega' \subset \Gamma$ where $\sigma(p)$ is high enough. It should be noted that in the case of strongly textured areas, it may happen that 1. and 2. are simultaneously verified for some points where no visual artifact exist. The experiments we made suggest that using spatial patch blending in these areas does generally not damage the quality of the reconstruction.

4 Results and Conclusions

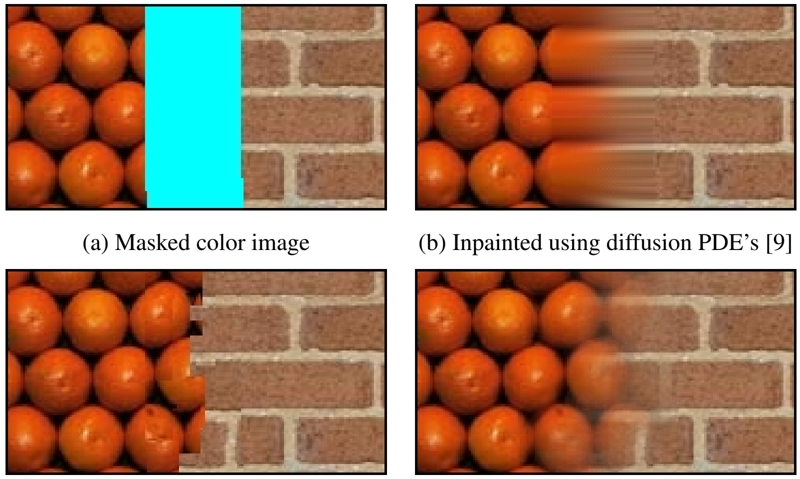
Fig. 4, 3 and 5 illustrate different results of blended inpainting on synthetic and real color images, as well as comparisons with other classical inpainting algorithms presented in [7,9,10,11,12]. The top row images of Fig.3 were post-processed with the same contrast enhancement filter for printing quality purpose. Through these examples, it is particularly interesting to see that our proposed blending method creates results which have all the good properties of other approaches at the same time, something between pure diffusion techniques (Fig.4b) where colors are smoothly interpolated but without reconstructed textures, and pure patch-based techniques (Fig.4c) where repetitive patterns are fully reconstructed. This way, the visual artefacts due to incoherent patch collage are strongly reduced, while the computation time overhead is negligible. The effect is close to of the results in [4,13], but takes less time to compute. Moreover, our approach only requires to set two additional parameters τ et ρ which are intuitive and easy to adjust. This makes our contribution very generic, and relevant for the improvement of any kind of patch-based inpainting algorithms.

(a) Masked colored image. (b) Inpainting result from [12]. (c) Inpainting result + patch blending (proposed algorithm).



(d) Masked colored image. (e) Inpainting obtained with the algorithm from [7]. (f) Inpainting result + patch blending (proposed algorithm).

Fig. 3: Comparison with the resynthetizer algorithm [7,12].



(a) Masked color image

(b) Inpainted using diffusion PDE's [9]

(c) Inpainted using a standard patch-based method [10]

(d) Inpainted using our proposed patch blending algorithm

Fig. 4: Illustrating the advantage of patch blending on a synthetic case.

References

1. Caselles, V.: Examplar-based image inpainting and applications. SIAM News **44**(10) (December 2011)
2. Masnou, S., Morel, J.M.: Level lines based disocclusion. In: ICIP (3). (1998) 259–263
3. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: Proc. of ICCV-Volume 2. ICCV '99, Washington, DC, USA, IEEE Computer Society (1999) 1033–1038
4. Arias, P., Facciolo, G., Caselles, V., Sapiro, G.: A variational framework for exemplar-based image inpainting. Int. J. Comput. Vision **93**(3) (July 2011) 319–347
5. Kawai, N., Sato, T., Yokoya, N.: Image inpainting considering brightness change and spatial locality of textures and its evaluation. In: Proc. of the 3rd PSIVT. PSIVT '09, Berlin, Heidelberg (2009) 271–282
6. Sun, J., Yuan, L., Jia, J., Shum, H.Y.: Image completion with structure propagation. ACM Trans. Graph. **24**(3) (July 2005) 861–868
7. Le Meur, O., Gautier, J., Guillemot, C.: Examplar-based inpainting based on local geometry. In: ICIP, Brussel, Belgium (2011) 3401–3404
8. Türkan, M., Guillemot, C.: Image prediction based on neighbor-embedding methods. IEEE Trans. Im. Proc. **21**(4) (2012) 1885–1898
9. Tschumperlé, D., Deriche, R.: Vector-valued image regularization with pdes: A common framework for different applications. IEEE Trans. PAMI **27**(4) (2005) 506–517
10. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. IEEE Trans. Im. Proc. **13**(9) (September 2004) 1200–1212
11. Harrison, P.: Image Texture Tools. PhD thesis, Monash University (2005)
12. Pérez, P., Gangnet, M., Blake, A.: Patchworks: Example-based region tiling for image editing. Technical report, Microsoft Research, MSR-TR-2004-04 (2004)
13. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. ACM Transactions on Graphics (SIGGRAPH'03) **22**(3) (2003) 313–318

(a) Masked colored image (used in [7]). (b) Inpainting result from [7]. (c) Inpainting result + patch blending (proposed algorithm).

(d) Masked colored image. (e) Inpainting result in [10]. (f) Inpainting result + patch blending (proposed algorithm).

(g) Masked image. (h) Inpainting obtained with [11].

(i) Inpainting obtained with [10]. (j) Inpainting result + patch blending (proposed algorithm).

Fig. 5: Comparison of the patch blending results with the state of the art inpainting methods [7,10,11].