A smarter exemplar-based inpainting algorithm using local and global heuristics for more geometric coherence

Maxime Daisy, Pierre Buyssens, David Tschumperlé, Olivier Lézoray

A SMARTER EXEMPLAR-BASED INPAINTING ALGORITHM USING LOCAL AND GLOBAL HEURISTICS FOR MORE GEOMETRIC COHERENCE

M. Daisy, P. Buyssens, D. Tschumperlé, O. Lézoray

GREYC (UMR 6072) - CNRS, Université de Caen Basse-Normandie, ENSICAEN - Image Team
6, Bd. Maréchal Juin - 14000 Caen, FRANCE

ABSTRACT

In this paper, we propose two major improvements to the exemplar-based image inpainting algorithm, initially formulated by Criminisi et al. [1]. First, we introduce a structure-tensor-based data-term for a better selection of pixel candidates to fill in based on priority. Then, we propose a new lookup heuristic in order to locate the best source patches to copy/paste to these targeted points. These two contributions clearly make the inpainting algorithm reconstruct more geometrically coherent images, as well as speed up the process drastically. We illustrate the great performances of our approach compared to existing state-of-the-art methods.

***Index Terms*—** Examplar-based image inpainting, Structure tensor analysis, Patch lookup strategy.

1. INTRODUCTION

Inpainting is the process of reconstructing unknown (masked) image regions such that the resulting image looks as natural as possible. Since 2000, this kind of algorithm has raised a wide interest in the image processing community due to the various applications it is able to deal with, such as object removal, or image repairing and interpolation. The literature on inpainting algorithms, reviewed in [2], exhibits mainly two kinds of approaches: geometry-based, and patch-based methods.

• *Geometry-based* methods [3, 4, 5] aim at reconstructing image regions using various *semi-local* geometry-aware interpolations of the known (noncorrupted) image data. These methods usually provide interesting results in terms of local geometry consistency. However, they mostly fail at reconstructing wide-scaled textures and often lead to flat-looking images particularly when large holes have to be filled.

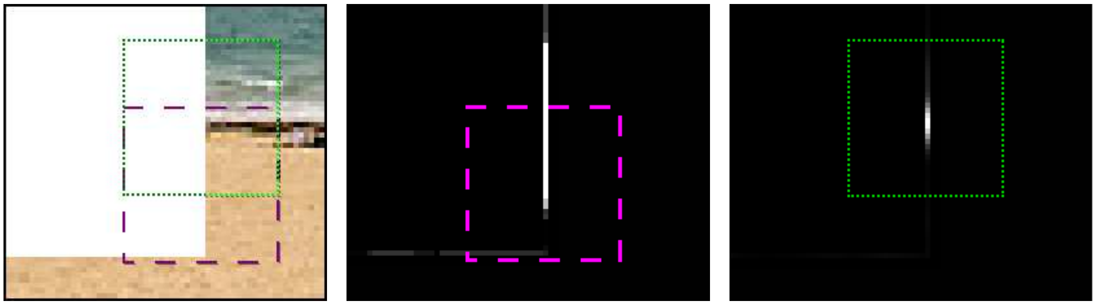
• On the other hand, *patch-based* methods [1, 6, 7, 8, 9] are the logical sequel of previous algorithms originally proposed for texture synthesis [10]. They focus on finding patterns which are present in the known part of the image and fit visually well with the local data lying on the boundaries of the missing regions. Such similar local pieces of images (i.e. patches) are then copied inside the masked region to fill in, iteratively. This kind of methods has proven to work remarkably well in case of quite large holes to fill, due to its ability to synthesize large-scale textures respectful of the image content. Nonetheless, such methods often lack a good global geometry analysis of the image. This leads to some unrealistic reconstruction of geometrical structures (often with block artefacts for instance). Although hybrid methods have been more recently proposed in the literature [11, 12] to take advantages of both geometric and texture approaches, most of these visual artefacts remain. In this paper we focus on the so-called *examplar-based inpainting* algorithm initiated by Criminisi et al. in [1], which has been quickly considered as a reference method in the field of patch-based inpainting. This paper proposes two important adjustments to the original method and show that they both clearly improves the geometric coherence of the inpainted results: we first reformulate the so-called data-term, in order to better select the pixel candidates to fill in based on priority. Then, we introduce a new lookup heuristic that is able to locate source patches to be copied back to the masked hole, in a very efficient way (both in terms of visual quality and execution speed). These two *geometry-guided* contributions greatly improve the visual quality of inpainted results when complex geometric structures are involved. We illustrate the effectiveness of our approach with difficult reconstruction examples of real images.

2. CONTEXT AND CONTRIBUTIONS

Examplar-based inpainting: Back in 2004, Criminisi et al. proposed a greedy inpainting algorithm [1] aiming at reconstructing a missing region Ω of an image $\mathcal{I}$ piece by piece, with a priority scheme. The order of pixel reconstruction was defined by a priority function P_p estimated at each $p \in \delta\Omega$ ($\delta\Omega$ being the boundary of Ω). The priorities $P_p = C_p \times D_p$ are defined over $[0, 1]$ s.t. the *confidence* term C_p that only depends on the local topology of Ω (the more pixels of Ω in the neighborhood of p means a lower confidence), while the *data* term D_p also depends on local image structures:

$$D_p = \frac{|\overrightarrow{\nabla I_p}^{\perp} \,.\, \overrightarrow{n_p}|}{\alpha} \text{ with } \overrightarrow{\nabla I}_p^{\perp} = \{\overrightarrow{\nabla I}_q^{\perp} \mid \underset{q \in ((\mathcal{I}-\Omega)\cap\Psi_p)}{\arg\max} \|\overrightarrow{\nabla I}_q\|\} \quad (1)$$

This research was supported by French national grant *Action 3DS*.



(a) Example of a local inpainting configuration, with Ω (in white) crossed by an horizontal contour (zoom).

(b) Map of the original data terms D_p (computed with (1)), and corresponding highest-priority patch (in hatched purple).

(c) Map of our modified data terms D_p (computed with (2)), and corresponding highest-priority patch (in dotted green).

Fig. 1. Illustration of the impact of the data terms (1) (original) or (2) (ours) for selecting the priority points to inpaint.

where Ψ_p is the patch with a fixed size, centered at p, $\overrightarrow{n}_p$ is the normal vector to the mask at p, and α a constant normalization factor (that can be in fact ignored). D_p is intended to favor the reconstruction of local linear structures orthogonally crossing Ω at p. Once all priorities P_p have been computed, the pixel p with the highest priority is chosen as the target pixel. For this p, the patch $\psi_{\hat{p}} = \{\psi_q \mid \underset{q|\psi_q \in (\mathcal{I}-\Omega)}{\arg\min} \|\Psi_p - \psi_q\|^2\}$ is sought *all over the image* (or generally into a square lookup region of fixed size centered at p) and is drawn at p in $\Psi_p \cap \Omega$. The confidences are then updated with $\forall\, q \in \Psi_p \cap \Omega, C_q = C_p$ as well as the boundary $\delta\Omega$. The whole process is repeated until $\delta\Omega = \varnothing$. This algorithm has been actually praised for the overall quality of the image reconstructions, and the acceptable execution time it provides. The heuristics proposed for defining the priorities P_p were actually good and quite hard to improve (some recent attempts in that direction has been tried in [2, 13]). Anyway, the algorithm has some drawbacks that we propose to manage hereafter. Note that, as the inpainting process is iterative, and as each patch Ψ_p copy-pasted in Ω at one iteration depends mostly on the local image geometry (i.e. on what happened during all previous iterations), it is not so surprising improving even slightly one or two aspects of the initial algorithm leads finally to great visual ameliorations in the results.

A better priority term: The data-term D_p must tell about whether linear structures are present or not at a point p and with which angle they eventually cross the region Ω. Unfortunately, as $p \in \delta\Omega$ on an point having no data, the gradient $\overrightarrow{\nabla I}_p$ cannot be precisely estimated at this point. In [1], it is suggested to approximate $\overrightarrow{\nabla I}_p$ as the maximum value of the image gradient in the known neighborhood $\Psi_p \cap \mathcal{I}$ of p (1). This approximation is actually a bit too coarse to get a locally accurate geometry analysis: when the patch size N is large, D_p will be high not only on the exact location of an eventual image contour crossing Ω, but also *for every target candidates no more distant than* N to this contour. This *dilation* effect of D_p gives too much importance to these neighboring points and one of them can be unluckily selected as having the highest priority (instead of the contour point itself), particularly if its confidence term C_p is higher. Fig.1(a)-(b) illustrates this assertion: here, D_p is high for all target points around the sand-sea frontier, and the final selected target patch to reconstruct (in hatched purple) will not be centered at the contour. A patch containing only sand will be likely pasted there, breaking seriously this local image structure.

To overcome this issue, we propose a more geometry-aware approach by using structure tensors [14] to modelize the image variations inside a candidate patch Ψ_p. First, this has the interest of better managing local image structures using a channel-correlated approach (R,G,B channels will be taken into account simultaneously). Second, we take advantage of the algebraic properties of the tensor sum, to allow patches containing structures with multiple orientations (typically textures) to score favorably whatever the local normal vector $\overrightarrow{n}_p$ to $\delta\Omega$ is. We propose to define $\tilde{D}_p$ as:

$$\tilde{D}_p = \|\mathbf{G}_p \overrightarrow{n}_p\| \;\text{ with }\; \mathbf{G}_p = \sum_{q \in \Psi_p \cap (\mathcal{I}-\Omega)} w_q \, \overrightarrow{\nabla I}_q \overrightarrow{\nabla I}_q^T \quad (2)$$

$\mathbf{G}$ is a weighted average of structure tensors estimated on non-masked parts $(\mathcal{I} - \Omega)$ of the target patch Ψ_p and w_p is a normalized 2d gaussian function centered at p, with a given constant standard deviation. Thus, these properties hold:

• When $\mathbf{G}$ is anisotropic ($\mathbf{G} \approx \overrightarrow{u}\overrightarrow{u}^T$), one main contour oriented along $\overrightarrow{u}^\perp$ lies inside the patch Ψ_p. (2) is almost equal to $| < \overrightarrow{u}, \overrightarrow{n}_p > |$ and will be indeed high when the contour crosses Ω orthogonally. Note that D_p will be higher for target points p that are *precisely* on the image contour (thanks to the weights w which will be maximal at the center of Ψ_p). No more dilation effects occur, and the contour point itself will have the highest data-term priority (Fig.1c).

• When $\mathbf{G}$ is isotropic (i.e. $\mathbf{G} \approx \lambda\ \mathbf{Id}$), (2) is almost equal to λ. As expected, it is low for homogeneous regions. But, it remains high when Ψ_p contains a lot of variations with

different directions (λ high) whatever the orientation of the mask normal $\overrightarrow{n}_p$ is (on the contrary, the original formulation (1) can be low there if the maximum gradient and the normal vector $\overrightarrow{n}_p$ are orthogonal by misfortune).
By using (2), we improve the finding of the point to reconstruct in priority. This has actually a tremendous incidence on the overall quality of the reconstructed images, as the iterative nature of the inpainting algorithm propagates reconstruction errors (Fig.3(b)-(c) show an example of differences).

A better patch lookup procedure: Our second contribution consists in the proposal of an improved patch search technique for the examplar-based inpainting algorithm [1]. We actually borrow ideas from the famous *Patchmatch* algorithm [15] and its recent extensions [16, 9]. These algorithms actually propose to efficiently compute an approximate Nearest Neighbor Field (NNF) for each point p of the image, which quickly returns the most similar patch to the one centered at p. This has been successfully used within multiscale pixel-based inpainting approaches, but it becomes far less efficient when dealing with purely patch-based methods, as it requires an update of the NNF each time a new patch is pasted, which is time-consuming in practice.
Our proposed extended search scheme relies on a similar use of offsets to reduce the patch lookup space. Let p_N be the center of the partially unknown target patch Ψ_{p_N} to fill in, $\mathcal{A} = \{p_0, \ldots, p_{N-1}\}$ the set of centers of the patches $\{\Psi_{p_0}, \ldots, \Psi_{p_{N-1}}\}$ already pasted during the previous iterations of the inpainting algorithm, $NN(p_i) = \hat{p}_i$ the center of the best patch $\psi_{\hat{p}_i}$ found for Ψ_{p_i}, and $NNF = \{\hat{p}_0, \ldots, \hat{p}_{N-1}\}$ the set centers of the best patches.
We first define the offset vector between p_N and the points of $\mathcal{A}$ as $\mathbf{\Delta}_{p_N}^{p_i} = p_N - p_i, \forall i \in \{0, \ldots, N-1\}$. Let S be the set of points p_i of $\mathcal{A}$ for which $\|\mathbf{\Delta}_{p_N}^{p_i}\| < T$. Our proposed search scheme consists in extending the search space by considering several *lookup sub-windows* centered around each point $NN(q) - \mathbf{\Delta}_{p_N}^{q}, \forall q \in S$. This scheme allows to search for the best patch candidate in potentially more interesting windows w.r.t. the already pasted patches. Note that a search window centered on p_N is still considered for the lookup. In order to reduce the number of overall lookup candidates, we set the size sz_N of all these sub-windows as $sz_N = sz/\sqrt{Card(S)}$, where sz is the size of the initial seach window (fixed parameter, used when no patches have been already reconstructed in a neighbor of the target point $p \in \delta\Omega$). The rationale behind this size is to keep a similar complexity regardless the number of visited sites. A schematic view of the process is illustrated on Fig. 2, where the target patch Ψ_{p_3} is the dotted square inside Ω, and the two search windows are depicted by the two large dotted squares centered on the initial guesses (obtained by the offsets $\mathbf{\Delta}_{p_1}^{p_3}$ and $\mathbf{\Delta}_{p_2}^{p_3}$ w.r.t. $\hat{p}_1$ and $\hat{p}_2$). For clarity purposes, the search window centered on the p_3 is not shown. Note that to inpaint a patch such as Ψ_0, the search window is bigger and centered

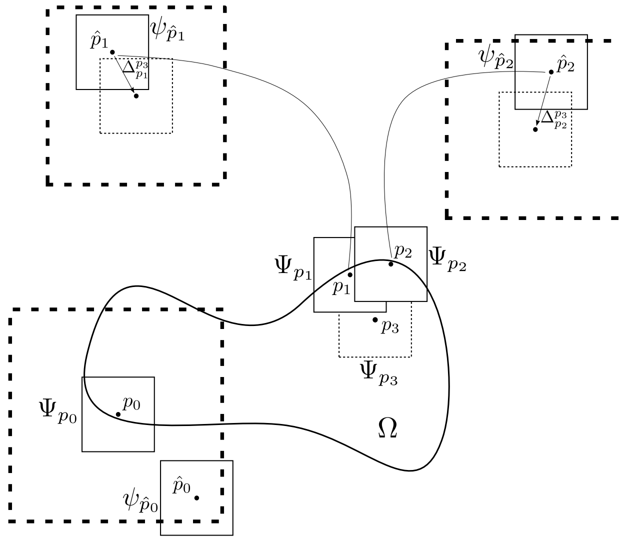


Fig. 2. Summarized view of our proposed patch lookup technique.

on its center p_0 since it has no direct already reconstructed neighbors (i.e. $S = \varnothing$). The advantages of our scheme are:
• As lookup sub-windows may overlap (they do it often in practice), the number of lookup candidates is always lower than the one defined by the initial searching scheme [1], hence we accelerate the whole process by a non-neglictible order of magnitude (see Fig.3(a)-(d)).
• As the size of the search space is reduced, the approximate nearest neighbor patch may be not optimal (in the sense of the SSD), but in fact it is often visually better. Lookup sub-windows are indeed centered at geometrically coherent locations w.r.t. the previously pasted patches, As a consequence, our modified inpainting algorithm becomes less sensitive to the patch size: in practice, one can recover larger coherent image structures and textures while considering smaller patches sizes than the original examplar-based algorithm [1].

3. RESULTS AND CONCLUSION

Fig. 3(a)-(d) illustrates the successive improvements of our two contributions to the original examplar-based inpainting algorithm [1]. The large size of the mask as well as the very complex structures involved in this image make it a very difficult case for an image inpainting algorithm. It clearly appears that our two improvements make the inpainted image more respectful of the initial image content. Fig. 3 shows that our final inpainting algorithm clearly competes with other state-of-the-art methods [17] (modified examplar-based inpainting) and [8, 16] (base of the Photoshop content-aware filling algorithm). For each result, an additional *spatial patch blending* step has been used to slightly improve the visual coherence (this is one of our previous work published in [18]). To ease the reproductibility of our research work, we have released the C++ sources and binaries of this inpainting algorithm within the *G'MIC* open-source framework [19].

(a) Masked image.

(b) Inpainting result using Criminisi et al. [1] *(1m48s)*.

(c) Inpainting result using [1] with our new data term (2) *(1m36s)*.

(d) Inpainting result using [1] with all improvements from this paper *(0m01s)*.

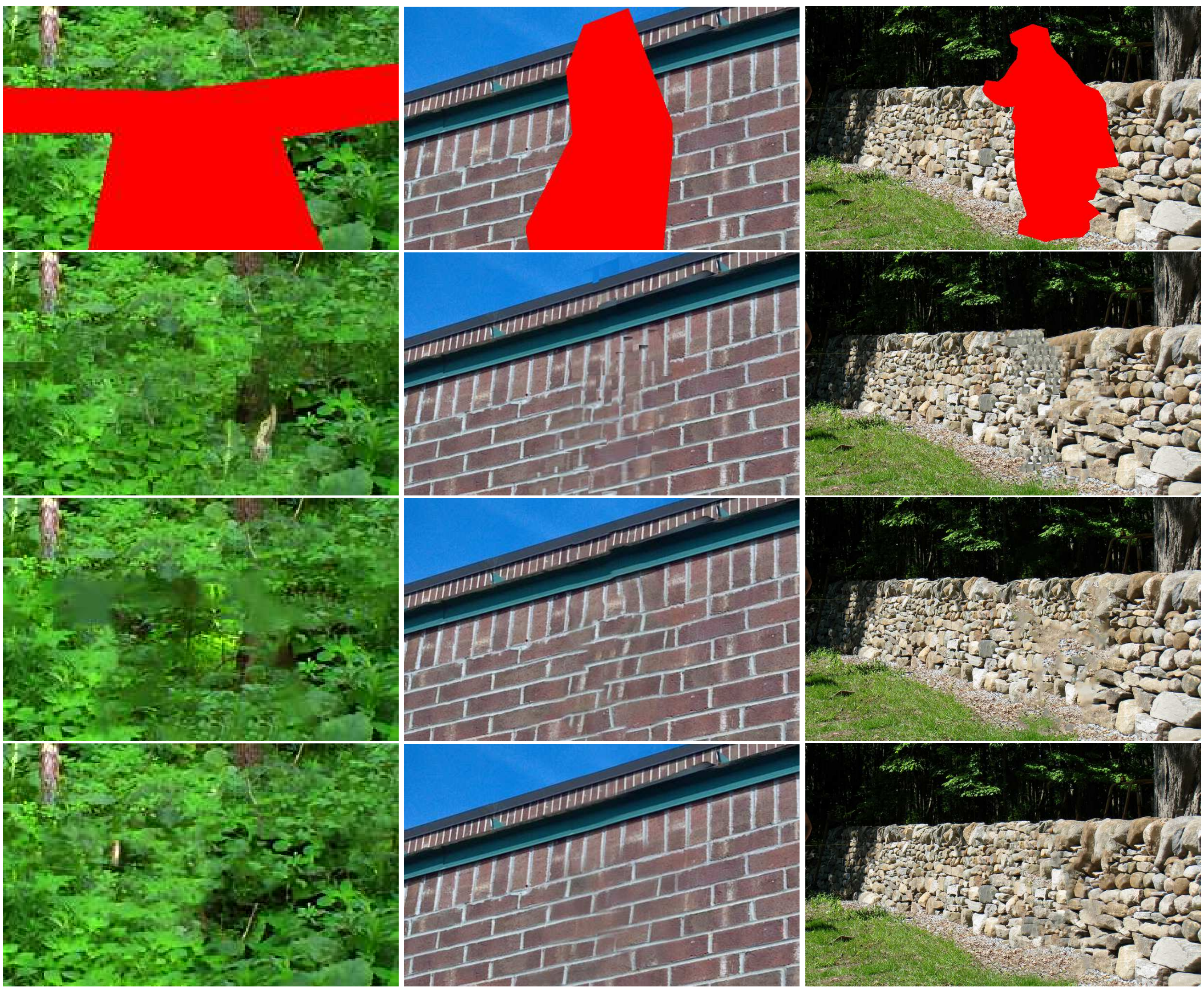


Fig. 3. First row: Illustration of the successive contributions presented in this paper. Then from top to bottom, comparisons with state-of-the-art inpainting algorithms: Masked images, results from Lemeur et al. [17], results from Wexler et al. [8] (using Patchmatch [16]), and results from our inpainting algorithm. We have tried to find optimal parameters to generate each of these results.

4. REFERENCES

[1] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Im. Proc.*, vol. 13, no. 9, pp. 1200–1212, Sept. 2004.

[2] Christine Guillemot and Olivier Le Meur, "Image inpainting: Overview and recent advances," *IEEE Signal Processing Magazine*, vol. 31, no. 1, pp. 127–144, Jan. 2014.

[3] S. Masnou and J-M. Morel, "Level lines based disocclusion," in *ICIP (3)*, 1998, pp. 259–263.

[4] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Proc. of the 27th annual SIGGRAPH conference*, New York, USA, 2000, pp. 417–424.

[5] D. Tschumperlé and R. Deriche, "Vector-valued image regularization with pdes: A common framework for different applications," *IEEE Trans. PAMI*, vol. 27, no. 4, pp. 506–517, 2005.

[6] P. Pérez, M. Gangnet, and A. Blake, "Patchworks: Example-based region tiling for image editing," Tech. Rep., Microsoft Research, MSR-TR-2004-04, 2004.

[7] J. Sun, L. Yuan, J. Jia, and H-Y. Shum, "Image completion with structure propagation," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 861–868, July 2005.

[8] Y. Wexler, E. Shechtman, and M. Irani, "Space-time completion of video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 463–476, Mar. 2007.

[9] K. He and J. Sun, "Statistics of patch offsets for image completion," in *Proceedings of the 12th European Conference on Computer Vision*, Berlin, 2012, vol. LNCS 7573, Part II, pp. 16–29, Springer-Verlag.

[10] A.A. Efros and T.K. Leung, "Texture synthesis by non-parametric sampling," in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, 1999, vol. 2, pp. 1033–1038.

[11] N. Kawai, T. Sato, and N. Yokoya, "Image inpainting considering brightness change and spatial locality of textures and its evaluation," in *Proc. of the 3rd PSIVT*, Berlin, Heidelberg, 2009, pp. 271–282.

[12] P. Arias, G. Facciolo, V. Caselles, and G. Sapiro, "A variational framework for exemplar-based image inpainting," *Int. J. Comput. Vision*, vol. 93, no. 3, pp. 319–347, July 2011.

[13] R. Martinez-Noriega, A. Roumy, and G. Blanchard, "Exemplar-based image inpainting: Fast priority and coherent nearest neighbor search," in *Machine Learning for Signal Processing (MLSP), 2012 IEEE International Workshop on*, Sept 2012, pp. 1–6.

[14] S.Di Zenzo, "A note on the gradient of a multi-image," *Computer Vision, Graphics, and Image Processing*, vol. 66, pp. 116–125, 1986.

[15] C. Barnes, E. Shechtman, A. Finkelstein, and Dan B. Goldman, "PatchMatch: A randomized correspondence algorithm for structural image editing," *ACM Transactions on Graphics (Proc. SIGGRAPH)*, vol. 28, no. 3, Aug. 2009.

[16] C. Barnes, E. Shechtman, Dan B. Goldman, and A. Finkelstein, "The generalized PatchMatch correspondence algorithm," in *Proceedings of the European Conference on Computer Vision*, Sept. 2010, vol. LNCS 6313, Part III, pp. 29–43.

[17] O. Le Meur, J. Gautier, and C. Guillemot, "Examplar-based inpainting based on local geometry," in *ICIP*, Brussel, Belgium, 2011, pp. 3401–3404.

[18] M. Daisy, D. Tschumperlé, and O. Lézoray, "A fast spatial patch blending algorithm for artefact reduction in pattern-based image inpainting," in *SIGGRAPH Asia Technical Briefs*, Hong Kong, 2013, pp. 8:1–8:4, ACM.

[19] D. Tschumperlé, "G'MIC, open-source framework for image processing," *http://gmic.sourceforge.net*, 2014.