



**HAL**  
open science

## Tensor-directed Spatial Patch Blending for Pattern-based Inpainting Methods

Maxime Daisy, Pierre Buysens, David Tschumperlé, Olivier Lézoray

► **To cite this version:**

Maxime Daisy, Pierre Buysens, David Tschumperlé, Olivier Lézoray. Tensor-directed Spatial Patch Blending for Pattern-based Inpainting Methods. *Computer Analysis of Images and Patterns*, Sep 2015, Valetta, Malta. hal-01164064

**HAL Id: hal-01164064**

**<https://hal.science/hal-01164064>**

Submitted on 16 Jun 2015

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Tensor-directed Spatial Patch Blending for Pattern-based Inpainting Methods

Maxime Daisy\*, Pierre Buysens, David Tschumperlé, Olivier Lézoray

GREYC CNRS UMR6072, Image Team, 6 Bd Maréchal Juin, 14000 Caen/France



(a) Inpainting without blending. (b) With isotropic blending. (c) With tensor-based blending.

**Fig. 1.** Effect of our anisotropic blending where the inpainted region is highlighted.

**Abstract.** Despite the tremendous advances made in recent years, in the field of patch-based image inpainting algorithms, it is not uncommon to still get visible artefacts in the parts of the images that have been resynthesized using this kind of methods. Mostly, these artifacts take the form of discontinuities between synthesized patches which have been copied/pasted in nearby regions, but from very different source locations. In this paper, we propose a generic patch blending formalism which aims at strongly reducing this kind of artifacts. To achieve this, we define a tensor-directed anisotropic blending algorithm for neighboring patches, inspired somehow from what is done by anisotropic smoothing PDE's for the classical image regularization problem. Our method has the advantage of blending/removing incoherent patch data while preserving the significant structures and textures as much as possible. It is really fast to compute, and adaptable to most patch-based inpainting algorithms in order to visually enhance the quality of the synthesized results.

**Keywords:** patch, blending, tensor-directed, geometry-aware, anisotropy

## 1 Introduction and Context

Image inpainting is an image processing task aiming at completing missing, corrupted, and/or undesired data inside an image. It is commonly used in a professional way to remove scratches or microphones in a video, or in a more amateur way, to remove undesired peoples or objects from photographs for example. As wisely described in [15], plenty of inpainting methods exist in the literature. They can be mainly categorized in two kinds of methods:

- Geometry-based methods [19,4,3,21,6] try to extend the local geometry inside the area to complete in order to reconstruct image structures as better as possible. These methods show impressive results in term of geometry synthesis, but mainly fail at building complex textures.

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

- Pattern-based methods have their origins in works on texture synthesis [14,2]. Based on the *self-similarity principle*, they use known parts of the image as potential sources to reconstruct the missing part. These methods can be clustered into three main groups: 1) The *greedy* methods [5,10,9,17] that copy/paste patch chunks in a greedy manner until the hole is filled. Efficient for texture synthesis, these methods can fail at reconstructing structures, and exhibit typical block-effect artifacts. 2) The *hybrid* methods [16,8] that incorporate geometry-based methods to first continue the main structures before inpainting the textures. These methods are quite slow in practice, and require a segmentation of the image to separate structures from textures, which is an ill-posed problem. 3) The *energy-based* methods try to minimize coherence energy function via a multi-resolution [22] or variational framework [1]. While the solution in [22] tends to produce blurry textures due to the mix of patches used for the reconstruction, [1] incorporates a Poisson variant derived from [20] to smooth the transitions between pasted chunks.

In this paper we focus on the greedy *pattern-based* inpainting methods for their ability at reconstructing large portions of textures. Especially, we are interested in enhancing the perceptual quality of the results they provide. Our main **contribution** is the proposal of a novel tensor-guided spatial blending algorithm that strongly reduces the typical block-effect artifacts, while preserving the sharpness of synthesized structures and textures.

**Notations:** In the following, we define by  $I : p \in \mathcal{I} \mapsto I(p) \in \mathbb{R}^3$  a multi-valued image. The known domain of this image is denoted  $\Omega$  while unknown one (often called *mask* or *hole*) is denoted  $\bar{\Omega}$ , with  $\Omega \cup \bar{\Omega} = \mathcal{I}$ . We define by  $\mathcal{N}_p$  a square neighbourhood domain of size  $N \times N$  centered at  $p$ , and by,  $\psi_p : q \in \mathcal{N}_p \subset \mathcal{I} \mapsto \psi_p(q) \in \mathbb{R}^3$  a patch value function. In the sequel, uppercase bold letters will stand for matrices while those in lowercase are for vectors.

**Structure Tensors:** Introduced in [23], *structure tensors* are a natural extension of gradient notion for multi-valued images. They represent the local image color changes: the value of the variations, and their directions. In this paper, we deal with two-dimensional RGB images, so structure tensors reduce to  $2 \times 2$  matrices defined as follows:

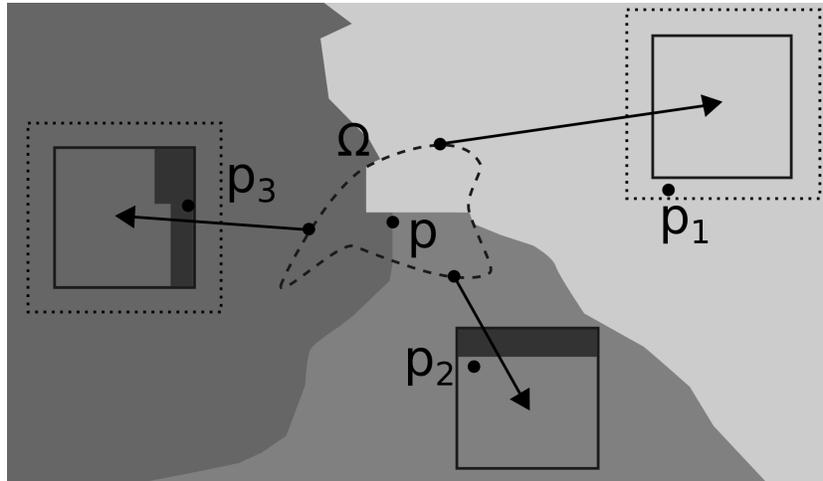
$$\begin{aligned} \mathbf{S} &= \sum_{c \in \{R,G,B\}} \overrightarrow{\nabla I^c} \cdot \overrightarrow{\nabla I^c}^T \\ &= \lambda_1 \cdot \mathbf{u} \cdot \mathbf{u}^T + \lambda_2 \cdot \mathbf{v} \cdot \mathbf{v}^T \quad \text{with} \quad \lambda_1 > \lambda_2 \end{aligned}$$

with  $\lambda_{\{1,2\}}$  the eigen values of  $\mathbf{S}$  and  $\mathbf{u}, \mathbf{v}$  the eigen vectors associated to  $\lambda_1$  and  $\lambda_2$  respectively. The eigen vector associated to the biggest eigen value is oriented along the major image color change direction while the one associated to the smaller eigen value is oriented along the smallest image variation direction. In the following, tensors are represented with ellipses whose diameters and orientation are given by the eigen values and eigen vectors respectively.

Since they define a robust and accurate local geometry model, structure tensors have been used for years in image processing, as for image regularization [21], geometry-based [21,7] or pattern-based [11,18] image inpainting.

**Spatial Patch Blending:** In our previous works [13,12], we proposed a patch blending method that tries to reduce block effect artifact in pattern-based image inpainting results. The core of the method is to mix the data of several patches in a way that the reconstruction seems more continuous than with blocky patch chunks. This algorithm is mainly composed of two main consecutive steps:

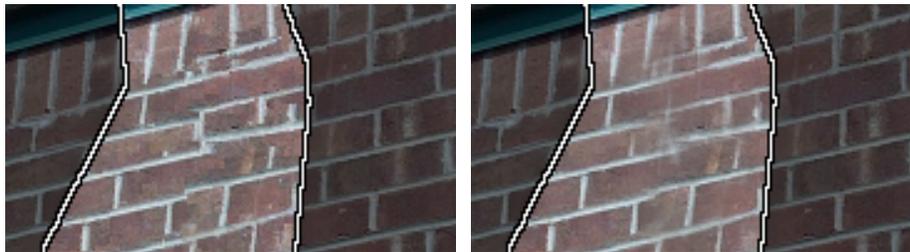
1. **Artifact detection** tries to locate and estimate the strength of the possible artifacts present in the image. To locate artifacts, two hypotheses are made a) there are sharp variations of image intensity, and b) the source patches used for the reconstruction come from very different locations. The output of this artifact detection pipeline is a blending amplitude map that indicates, for each pixel, the bandwidth of blending to be applied locally.
2. **Patch blending** step uses the previously computed blending amplitude map as a model. Figure 2 illustrates the principle: for each  $p \in \Omega$ , the spatial blending performs a linear combination of all the pixels  $\{p_1, \dots, p_n\}$  overlapping  $p$  from the set of reconstructed patches  $\{\psi_{p_1}, \dots, \psi_{p_n}\}$ .



**Fig. 2.** Illustration of the spatial patch blending principle.

Despite great improvements to the final visual quality of the inpainted image, this method suffers from a significant flaw: the spatial blending is performed through isotropic Gaussian weights, that do not respect enough the image structures. Figure 3 shows an inpainting result of a textured image (left) and its blended result (right) with this algorithm. Since the blending is applied in all directions (isotropic), some joints between bricks are too blurry. Such a flaw in fact damages both structures and textures, and that is the point we try to solve in this paper.

In the following of this paper, we propose a spatial patch blending method that is much more careful about the local image structures and textures. We first define a geometric model for patch blending, and then explain how we use it to apply a patch blending aware of local image geometry.



(a) Pure patch-based in-painting result without patch spatial (b) Result (a) with the application of isotropic patch blending blending. [12].

**Fig. 3.** Example of damaged textures with isotropic blending.

## 2 Tensor-directed Spatial Patch Blending

In this section, we describe the pluralist contribution of this paper: a spatial patch blending method respecting the local image geometry. First, we define a geometric model for the spatial patch blending. Then we explain how to use this model to perform spatial patch blending, either with the elemental formulation, or with a faster one.

### 2.1 Tensor Model for Spatial Patch Blending

The geometric model we propose reflects the strength and the orientation of the patch blending to be applied locally. As all these information can be represented by eigen values/vectors of tensors, this kind model is one of the most adapted to create our geometric blending model. The local amount of the color intensity variations and directions are encoded inside a structure tensor  $\mathbf{S}$  by their eigen values  $\lambda_{\mathbf{S}\{1,2\}}$  and vectors  $\mathbf{e}_{\mathbf{S}\{1,2\}}$  respectively. In the same way, local blending properties, i.e. strength and direction, can be put inside tensors. The eigen values  $\lambda_{\mathbf{B}\{1,2\}}$  and eigen vectors  $\mathbf{e}_{\mathbf{B}\{1,2\}}$  of blending tensors  $\lambda_{\mathbf{B}}$  represent respectively the bandwidth and the direction of the spatial patch blending to be applied locally. As structure tensors already provide a good local geometry analysis, we propose to use them as a basis for building our *blending tensor* model. Transformation on eigen values and vectors of structure tensors is performed in a way that the final tensor fit our model. For flat area, patch blending must be omnidirectional, strong enough to remove small reconstruction artifacts, and highly smooth sharp variations. In this case, blending tensors keep the isotropic shape of structure tensors, but are much bigger. The textures are very important inside an image and should

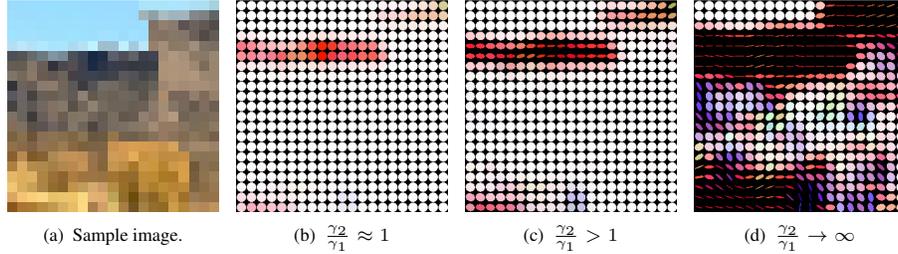
be preserved. In this case, blending tensors have the same properties of structure tensor, namely a small and isotropic shape. Finally, for sharp image structures, blending is able to preserve them while smoothing small breaks. Hence, they are oriented along the smallest variation direction, orthogonally to structure tensors. From this description of our tensorial blending model, we propose the following steps to build it. As computed using image gradient, structure tensors have eigen values that are fully dependent of the image value range, which is not be the case of blending tensors. Hence, the first step of the blending tensor construction is to normalize structure tensor eigen values:

$$\hat{\lambda}_{\mathbf{S}(p)i} = \frac{\lambda_{\mathbf{S}(p)}}{\max_{p \in \mathcal{I}} \lambda_{\mathbf{S}(p)i}}$$

The local blending bandwidth, and so the eigen values of blending tensors, highly depends on the local image geometry which is given by the ratio between the smallest and the biggest eigen values of structure tensors. The less the ratio, the more anisotropic the tensor. At the contrary, the more the ratio, the more isotropic the tensor. Therefore, the next step is to modify eigen values  $\hat{\lambda}_{\mathbf{S}(p)i}$  depending on this ratio in order to have new eigen values  $\lambda_{\mathbf{B}i}$ . The proposed function is inspired by partial differential equations for diffusion [21] and is defined as the following:

$$\lambda_{\mathbf{B}i} = \frac{1}{(1 + \hat{\lambda}_{\mathbf{S}1} + \hat{\lambda}_{\mathbf{S}2})^{\gamma_i}} \quad (1)$$

where  $\gamma_i$  ( $i \in \{1, 2\}$ ) are parameters controlling overall tensor isotropy. Examples of the effect of different configurations of  $\gamma_i$  are provided in Fig. 4. The final step is to



**Fig. 4.** Illustration of the effect of the values of parameters  $\gamma_i$ .

build the blending tensor itself. As a symmetric and positive definite matrix, it can be expressed using the following composition:

$$\mathbf{B} = \lambda_{\sigma \mathbf{B}1} \mathbf{e}_{\mathbf{S}1}^{\perp \cdot T} \mathbf{e}_{\mathbf{S}1}^{\perp} + \lambda_{\sigma \mathbf{B}2} \mathbf{e}_{\mathbf{S}2}^{\perp \cdot T} \mathbf{e}_{\mathbf{S}2}^{\perp} \quad (2)$$

where  $\lambda_{\sigma \mathbf{B}i} = \sigma_{\mathbf{B}} \lambda_{\mathbf{B}i}$ , are the eigen values of the blending tensors.

Type of image area	Blending strength	Direction
flat area	strong	all
textured area	small	all
structured area	weak	structure

**Table 1.** Correspondence between types of image areas and how patch blending should apply on these areas.

The above description gives a blending tensor configurations depending on the local geometry (see Table 1). The tensor model proposed in this section provides all local information needed to apply a spatial patch blending aware of image structure and texture. In the next section, we describe how to use this model to perform spatial patch blending on patch-based inpainting results.

## 2.2 Elemental Tensor-directed patch blending

The first proposed scheme is the elemental formulation of our geometry-guided patch blending process. This scheme aims at applying patch blending in a pixel-wise fashion using the proposed tensor model  $\mathbf{B}(p)$ . Our approach is to compute a linear combination of pixels coming from several patches overlapping one another, using the local geometry. For each  $p \in \Omega$  in each image channel  $k$ , the following formula is applied:

$$J^k(p) = \frac{\sum_{\psi_q \in \Psi_p} w_{\mathbf{B}}(p, q) \psi_q^k(p - q)}{\varepsilon + \sum_{\psi_q \in \Psi_p} w_{\mathbf{B}}(p, q)} \quad (3)$$

where  $\Psi_p = \{\psi_0, \dots, \psi_n\}$  is the set of source patches overlapping  $p$ , and  $w_{\mathbf{B}}$  is an anisotropic Gaussian weight function defined as follows:

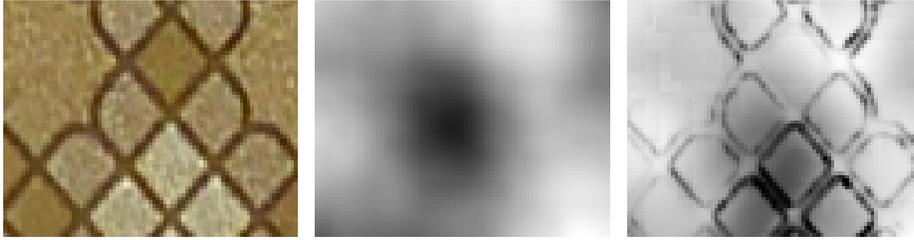
$$w_{\mathbf{B}}(p, q) = e^{-\frac{X^T \mathbf{B}(p)^{-1} X}{2\sigma_{\mathbf{B}}^2}} \quad \text{with } X = q - p \quad (4)$$

The effect is to blend very strongly flat areas such that no seams appear, while blending slightly along image contours. This allows to keep the strong image structures reconstructed during inpainting and also the complex textures.

## 2.3 Faster Tensor-directed patch blending

The elemental formulation of the spatial patch blending, as pointed out in [12], is slow to compute and does not allow to use it easily. In this section, we propose an adaptation of the work of [12] to the geometric model proposed in this paper.

The proposed algorithm remains quite the same that those in [12]. The first difference lies in the Gaussian weights used to apply patch blending at each scale of the blending amplitude map. While these weights are isotropic in [12], in this version we use anisotropic Gaussian weights based on the geometric model described in 2.1



**Fig. 5.** Illustration of the difference between summed patch blending weights using isotropic weights (middle), and anisotropic geometry-aware weights (right) on a sample image (left).

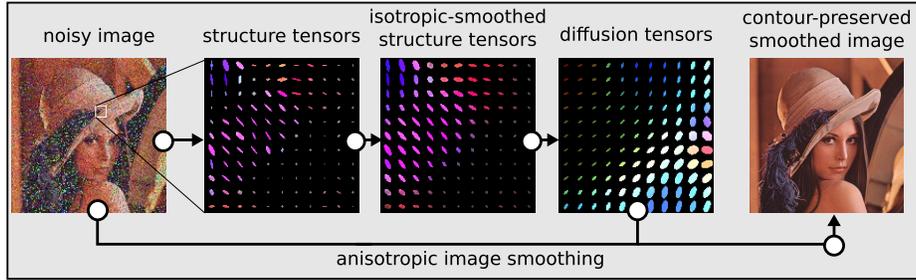
(see Fig. 5). As previously described, proposed geometric model describes the way to blend patches locally. Therefore, there is no need to compute patch blending at multiple scale since it is done per se with proposed tensor model. In this faster blending method, geometry of reconstructed patches are approximate using regularized blending tensor on their center. The computation time of this method is really smaller than those of elemental method. While in the elemental formulation there are as many weight functions as patches overlapping the position a specified pixel to compute, in the faster formulation only one weight function is computed per patch. Table 2 summarizes the computation times for some of the images of this paper. One can notice that compared to the elemental formulation, the faster one is more stable with respect to inpainting patch size. It is also fast with high resolution images.

Size	Missing	Patch size	Elemental	Fast
$500 \times 375$	7%	9	0.85	0.6
-	-	21	5.5	0.6
-	-	43	33.5	0.65
$1024 \times 768$	12%	9	5.2	2.6
-	-	21	29	2.2
-	-	43	33.5	2.74

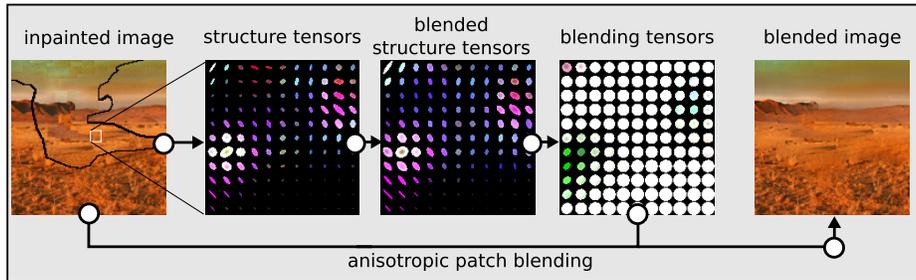
**Table 2.** Comparison of computation times (in sec) between elemental and fast anisotropic patch blending method.

## 2.4 High Level Process

The proposed process for tensor-directed patch blending is similar to anisotropic image smoothing, as described in Fig. 6. In the latter case, in image denoising for example, the structure tensor field is first extracted from the input image. This field is then smoothed using an isotropic Gaussian kernel. The effect is to regularize the tensor field and to get a more continuous version of it. Finally, the regularized structure tensor field is used to apply anisotropic smoothing on the input image.



(a) Anisotropic image smoothing process.



(b) Geometry-guided patch blending process.

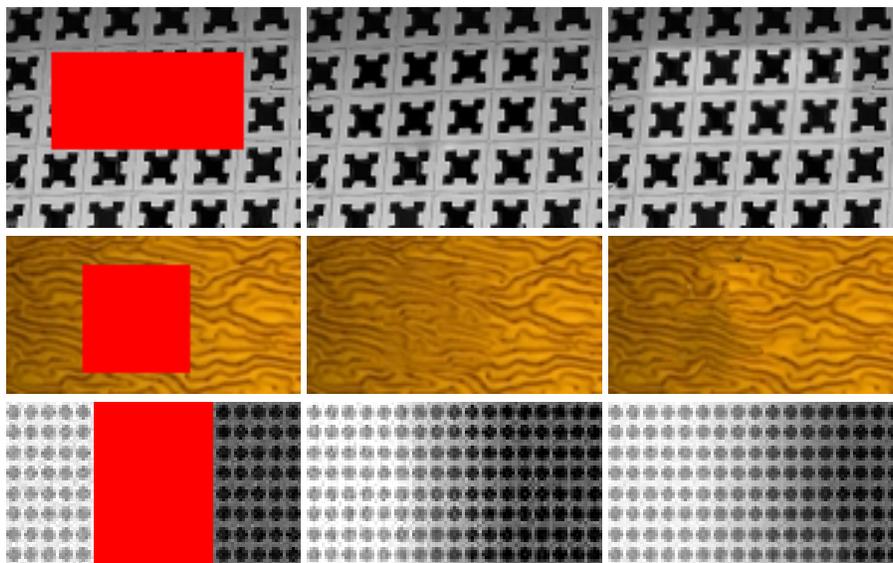
**Fig. 6.** Analogy between anisotropic image smoothing and proposed geometry-guided spatial patch blending.

In the case of the proposed tensor-directed spatial patch blending algorithm, the process is very similar with the isotropic one. This algorithm contains four main consecutive steps (see Fig. 6):

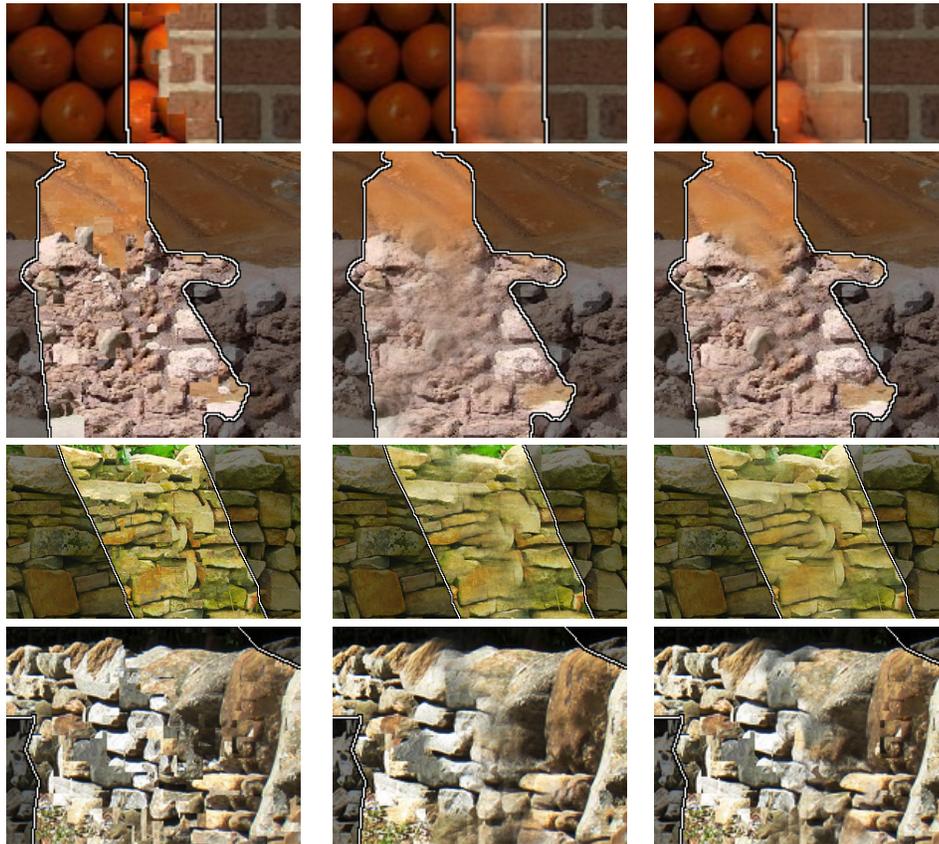
1. The blending tensor field is computed on the masked version of the image. This step is quite thorny since the image domain is not defined on  $\Omega$ . However, image gradient at the boundary of  $\Omega$  can be estimated using finite differences schemes depending on the known neighborhood of a pixel.
2. In order to obtain a complete tensor field, the structure tensor field is reconstructed simultaneously with the input image during inpainting. Also, it can be reconstructed from the warping map, after the inpainting.
3. Once the tensor field is reconstructed, it is blended in an isotropic fashion using the method of [12]. The effect is to smooth the tensors that are induced by the reconstruction seams between patches. In this way, they do not interfere in the anisotropic patch blending step.
4. Finally, patch blending is performed on the inpainted image with the previously described scheme (see. Eq. (3,4)).

### 3 Results, Comparisons and Discussions

Fig. 7 compares our approach with the one of [1]. On the tiles image (first row), the result of [1] is a little blurry while the edges with our approach are slightly more preserved. On the yellow texture image (second row), one can see that our method brings more geometric coherence. Finally on the dots image (third row), one can notice that the effect of our anisotropic blending is near of the effect of [1]. In Fig. 8 we show comparisons of our method results (right column) with those of [12] (middle column). In a common way, one can see that the results of [12] present some structure superposition effect due to the blending of structures and flat areas, like between the rocks (rows 2-4). Also, one can clearly notice that the structures of the rocks (third row) are more preserved with our method, than with the method of [12] that presents more evanescent structures. Through these results, one can see that the geometric model we propose for patch blending algorithm really reduces the smoothing effect present in [12], and provides similar or better results than state-of-the-art methods. In addition, proposed method frees users from choosing a scheme depending on the type of image. Concerning the computation time, proposed method is longer than isotropic one. This is because for isotropic method, Gaussian weights can be computed once at each scale, and reused for every patches. In the case of anisotropic blending, the Gaussian weight function depends on the local patch geometry. Hence, as many Gaussian weight function as number of patches have to be computed.



**Fig. 7.** Comparison between results of state-of-the-art method [1] (middle column) and proposed method (right column).



(a) Inpainting without blending. (b) Using blending method of [12]. (c) Proposed blending method.

**Fig. 8.** Comparison of results of the blending our method with the method of [12].

## 4 Conclusions and Future Work

This paper presents a structure and texture preservative method to reduce artifacts in image inpainting methods using patch-based synthesis. Using structure tensors, we compute “blending” tensors that aim at directing the patch blending along the structures, flattening the nearly placid area, and damage texture as less as possible. Experiments and comparisons were made on several challenging cases. They show that the proposed method really improves the quality of the spatial patch blending. In future works we plan to adapt this method to video inpainting by fully integrating the temporal dimension to our spatial patch blending geometric model.

## References

1. Arias, P., Facciolo, G., Caselles, V., Sapiro, G.: A variational framework for exemplar-based image inpainting. *International Journal of Computer Vision* 93(3), 319–347 (Jul 2011), <http://dx.doi.org/10.1007/s11263-010-0418-7> 2, 9
2. Ashikhmin, M.: Synthesizing natural textures. In: *Proceedings of the 2001 symposium on Interactive 3D graphics*. pp. 217–226. ACM (2001) 2
3. Ballester, C., Bertalmio, M., Caselles, V., Sapiro, G., Verdera, J.: Filling-in by joint interpolation of vector fields and gray levels. *IEEE Transactions on Image Processing* 10(8), 1200–1211 (2001) 1
4. Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. pp. 417–424 (2000) 1
5. Bornard, R., Lecan, E., Laborelli, L., Chenot, J.H.: Missing data correction in still images and image sequences. In: *Proceedings of the tenth ACM international conference on Multimedia*. pp. 355–361 (2002) 2
6. Bornemann, F., März, T.: Fast image inpainting based on coherence transport. *Journal of Mathematical Imaging and Vision* 28(3), 259–278 (2007) 1
7. Bugeau, A., Bertalmio, M., et al.: Combining texture synthesis and diffusion for image inpainting. *Combining Texture Synthesis and Diffusion for Image Inpainting*. pp. 26–33 (2009) 2
8. Cao, F., Gousseau, Y., Masnou, S., Pérez, P.: Geometrically guided exemplar-based inpainting. *SIAM Journal on Imaging Sciences* 4(4), 1143–1179 (2011) 2
9. Criminisi, A., Pérez, P., Toyama, K.: Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing* 13(9), 1200–1212 (Sep 2004) 2
10. Criminisi, A., Pérez, P., Toyama, K.: Object removal by exemplar-based inpainting. In: *Computer Vision and Pattern Recognition*. vol. 2, pp. II–721. IEEE (2003) 2
11. Daisy, M., Buysens, P., Tschumperlé, D., Lézoray, O.: A smarter exemplar-based inpainting algorithm using local and global heuristics for more geometry coherence. In: *International Conference on Image Processing*. Paris, France (2014) 2
12. Daisy, M., Tschumperlé, D., Lézoray, O.: A fast spatial patch blending algorithm for artefact reduction in pattern-based image inpainting. In: *SIGGRAPH Asia 2013 Technical Briefs* (2013) 3, 4, 6, 8, 9, 10
13. Daisy, M., Tschumperlé, D., Lézoray, O.: Spatial patch blending for artefact reduction in pattern-based inpainting methods. In: *Computer Analysis of Images and Patterns*. vol. LNCS 8048, pp. 523–530. York, UK (2013) 3
14. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: *International Conference on Computer Vision*. vol. 2, pp. 1033–1038. IEEE (1999) 2
15. Guillemot, C., Le Meur, O.: Image inpainting: Overview and recent advances. *Signal Processing Magazine, IEEE* 31(1), 127–144 (2014) 1
16. Jia, J., Tang, C.K.: Inference of segmented color and texture description by tensor voting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 26(6), 771–786 (2004) 2
17. Le Meur, O., Ebdelli, M., Guillemot, C.: Hierarchical super-resolution-based inpainting. *IEEE Transactions on Image Processing* 22(10), 3779–3790 (2013) 2
18. Le Meur, O., Gautier, J., Guillemot, C.: Exemplar-based inpainting based on local geometry. In: *International Conference on Image Processing*. pp. 3401–3404. Brussel, Belgium (2011), <http://hal.inria.fr/inria-00628074> 2
19. Masnou, S., Morel, J.M.: Level lines based disocclusion. In: *International Conference on Image Processing* (3). pp. 259–263 (1998) 1

20. Pérez, P., Gangnet, M., Blake, A.: Poisson image editing. *ACM Transactions on Graphics* 22(3), 313–318 (Jul 2003) [2](#)
21. Tschumperlé, D., Deriche, R.: Vector-valued image regularization with pdes: A common framework for different applications. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27(4), 506–517 (2005) [1](#), [2](#), [5](#)
22. Wexler, Y., Shechtman, E., Irani, M.: Space-time completion of video. *IEEE Transaction on Pattern Analysis and Machine Intelligence*. 29(3), 463–476 (Mar 2007) [2](#)
23. Zenzo, S.D.: A note on the gradient of a multi-image. *Computer Vision, Graphics, and Image Processing* 33(1), 116 – 125 (1986), <http://www.sciencedirect.com/science/article/pii/0734189X86902239> [2](#)