Combining Texture Synthesis and Diffusion for Image Inpainting.

Aurélie Bugeau, Marcelo Bertalmio

COMBINING TEXTURE SYNTHESIS AND DIFFUSION FOR IMAGE INPAINTING

Aurélie Bugeau
Image processing group, Barcelona Media, Barcelona, Spain
aurelie.bugeau@barcelonamedia.org

Marcelo Bertalmío
Dept. de Tecnologies de la Informació i les Comunicacions, Universitat Pompeu Fabra, Barcelona, Spain
marcelo.bertalmio@upf.edu

Keywords: Image inpainting, texture synthesis, structure, diffusion, image decomposition.

Abstract: Image inpainting or image completion consists in filling in the missing data of an image in a visually plausible way. Many works on this subject have been proposed these recent years. They can mainly be decomposed into two groups: geometric methods and texture synthesis methods. Texture synthesis methods work best with images containing only textures while geometric approaches are limited to smooth images containing strong edges. In this paper, we first present an extended state of the art. Then a new algorithm dedicated to both types of images is introduced. The basic idea is to decompose the original image into a structure and a texture image. Each of them is then filled in with some extensions of one of the best methods from the literature. A comparison with some existing methods on different natural images shows the strength of the proposed approach.

1 INTRODUCTION

The problem of image completion or *image inpainting* can be defined as follows: given an image containing missing regions (corresponding for example to some objects one wants to remove), the completion consists in filling in these missing parts in such a way that the reconstructed image looks natural (*i.e.* visually plausible). Basically, given a corrupted image $I : \Phi \rightarrow \mathbb{R}^3$ (or eventually $I : \Phi \rightarrow \mathbb{R}$ in case of grayscale images) and a mask $M : \Phi \rightarrow \{0,1\}$, the inpainting algorithm must fill in the region Ω where $M(p) = 0$ (here p is a pixel belonging to Φ).

This problem is for the moment far to be solved. Indeed, natural images are complex and may contain different textures, strong structures (edges), etc. Existing methods can be divided into two main groups: ”geometric” methods which allow to get regularized contours but do not give some nice reconstruction of textured areas and texture synthesis methods which basically do the opposite. Before going any further, we now briefly review these two types of approaches.

1.1 “Geometric” methods

This first category of image completion methods try to fill-in the missing regions of an image through a diffusion process, by propagating the information known on the boundary towards the interior of the holes. Generally these methods consist in finding the global minimum of an energy function, by converging to the steady state of the corresponding Partial Differential Equation (PDE) (section 1.1.1). As, most of the time, a lot of iterations are needed before the convergence can be reached, these methods are computationally expensive. Therefore some algorithms propagating the information from the boundary inwards without finding the global minimum of an energy function have also been proposed (see section 1.1.2).

1.1.1 PDE’s based image inpainting

The term of ”digital inpainting” was first introduced in (Bertalmio et al., 2000). In this paper, a third order PDE, solved only inside Ω with proper boundary conditions in $\delta\Omega$, was proposed. Its purpose is to propagate image Laplacians in the isophote (line of constant intensity) directions. The edges recovered with this approach are smooth and continuous at the boundary of the hole. However, it can not deal with texture and, whenever the area to be inpainted is too big, the result exhibits a lot of blur and is not contrast invariant. This work was partly inspired by the Image Disocclusion algorithm of Masnou and Morel (Masnou and Morel, 1998) (see also (Masnou, 2002), in which (an approximation of) the Elastica functional was minimized inside the image gap. In (Ballester et al., 2001) the Elastica functional is relaxed and then minimized with a system of two coupled PDE's.

Many other PDEs dedicated to inpainting have been proposed in the literature. Inspired by (Bertalmio et al., 2000), the authors of (Chan et al., 2001) proposed to minimize the total variation energy which corresponds to a second-order PDE. However, this PDE does not maintain the curvature. With this second-order PDE the contours are joined with straight isophotes, not necessarily smoothly continued from the mask boundary. Hence, it is only intended for small gaps. In (Chan and Shen, 2001), the previous method has been extended by adding a term diffusing the curvature in the direction normal to the isophote. It then becomes a third-order PDE but still creates visible corners due to its straight line connections. This is because the principle of good continuation (the human visual system completes missing curves with curves that are as short and as smooth as possible) is not respected. As a consequence, a fourth-order PDE was proposed in (Chan et al., 2002). It is based on the Euler's elastica (Mumford, 1994). Unfortunately, in pratice, this method is extremely computationally consuming since convergence requires a high number of iterations.

All previous PDEs where based on curvature diffusion. Other PDEs, also relying on the isophotes as was (Bertalmio et al., 2000), exist. For example, in (Bertalmío, 2006), a third-order PDE complying with the principle of good continuation was proposed. While this PDE is the best third order PDE for image inpainting, it still does not well permit to handle images containing textures and its high order makes it computationally expensive.

Another PDE method dedicated to anisotropic smoothing was developed in (Tschumperlé, 2006). This method is based on the observation that all previous PDEs locally smooth (or diffuse) the image along one or several directions that are different at each image point, the principal smoothing direction being parallel to the isophotes. As shown in this paper, another approach is to retrieve the geometry of the main structures inside the gap and to apply anisotropic diffusion following this geometry. The geometry can be retrieved using a structure tensor field and the direction of the smoothing comes from a second field of diffusion tensors.

1.1.2 Inpainting based on coherence transport

The global resolution of all the PDEs previously presented is made iteratively, which makes these methods very time consuming. A non iterative method, based on coherence transport, was proposed in (Bornemann and März, 2007). This algorithm relies on (Telea, 2004), in which Telea introduced a fast algorithm for image inpainting that calculates the image values inside the masks by traversing these pixels in just a single pass using the fast marching technique. The mask is filled in a fixed order, from its boundary inwards, using weighted means of already calculated image values.

The main principle of the algorithm from (Bornemann and März, 2007) is to define the weights such that the image information is transported in the direction of the isophotes. The directions of the edges are extracted using the eigenvalues of structure tensors.

1.2 Texture synthesis methods

Given a texture sample, the texture synthesis problem consists in synthesizing other samples from the texture. The usual assumption is that the sample is large enough to capture the stationarity of the texture. There have been many works extending texture synthesis to inpainting. We here only present two of the major works in this domain.

In the seminal paper (Efros and Leung, 1999), the authors have presented a simple yet effective non-parametric texture synthesis method based on local image patches. The texture is modelled as a MRF by assuming that the probability distribution of brightness values for one pixel given the brightness values of its spatial neighbourhood is independent from the rest of the image. The neighbourhood is a square window around the pixel and its size is fixed by hand. The input of the algorithm is a set of model image patches and the task is to select an appropriate patch to predict the value of an unknown pixel. This is done by computing a distance measure between the known neighbourhood of an unknown pixel and each of the input patches. The distance is a sum of squared differences (SSD) metric.

The authors of (Criminisi et al., 2004) proposed an extension of Efros and Leung's method, with two improvements. The first one concerns the order in which the pixels are synthesized. Indeed, a system for assigning priorities is used (assigning priorities to the pixels was also proposed in (Harrison, 2001)). The priority order at a pixel is the product of a confidence term, which measures the amount of reliable information surrounding the pixel, and a data term, that encourages linear structures to be synthesized first. The second improvement is a speed-up process. Contrary to the original method in (Efros and Leung, 1999), when synthesizing a pixel, not only the value of this pixel is inpainted in the output image (using the patch that gives the smallest distance metric), but all the pixels in its neighbourhood that have to be inpainted are filled in.

1.3 Overview of the paper

As already mentioned, none of the previously described methods is well adapted for all type of images. In this paper we will therefore combine diffusion and texture synthesis for image inpainting. Such a combination has already been proposed in (Bertalmio et al., 2003). In the current work, we propose an extension to this paper mainly using some of the methods mentioned in this introduction. After reviewing the method of (Bertalmio et al., 2003) in section 2, a new algorithm will be proposed in section 3. We end the paper with some experimental results and comparisons with existing approaches.

2 COMBINING TEXTURE SYNTHESIS WITH DIFFUSION

Both texture synthesis and diffusion have their own advantages and drawbacks for image inpainting. While diffusion gives blurred results it allows a continuity of the contours. Texture synthesis permits to conserve the textures but usually fails at preserving the edges and big structures. Therefore combining these two main types of approaches seems judicious. As proposed in (Sun et al., 2005), a user can interact to first specify important missing structures, where a specific structure propagation technique is applied. A texture synthesis approach is then used to fill in the rest of the image. Another algorithm compining both types of approaches without requiring any user intervention is (Bertalmio et al., 2003). It is based on the decomposition of the original image into a “structure” and a “texture” image.

2.1 Structure/Texture decomposition

A solution to keep the advantages of diffusion and texture synthesis is to separate the image into a texture image and a structure image. The problem is then to find a structure image u containing only the big structures of the original image I and a texture image v such that:

$$I = u + v. \tag{1}$$

A method directly dedicated to finding these two images has been proposed in (Vese and Osher, 2003). The problem consists in finding the structure image that minimizes the energy:

$$F(u) = \int |\nabla u| + \lambda \int \|v\|_{L^2}^2. \tag{2}$$

The first term of eq.(2) is a smoothness term whose role is to remove the noise, and the second term is a data fidelity term. This equation tends to remove the noise while preserving the important structures. Unfortunately, in (Meyer, 2001), Meyer showed that when using eq.(2), the texture image does not only contain oscillations (corresponding to noise) but also the brightness' edges. In fact, the L^2 space is not appropriate for modelling oscillatory patterns. He therefore suggested to use the Banach space. The authors of (Vese and Osher, 2003) proposed an algorithm to decompose the original image into a texture and a structure image in that case. An example of such a decomposition is shown in figure 1. Note how the structure image (second column) looks like a cartoon image that only contains important structures while the texture images (third columns) contains mainly small edges and noise.

2.2 An existing combination algorithm

An inpainting algorithm based on a structure/texture image decomposition has been presented in (Bertalmio et al., 2003). The idea is to first decompose the original image into a structure image and a texture image using the method of (Vese and Osher, 2003) previously presented. The structure image is reconstructed by the inpainting method of (Bertalmio et al., 2000) and the second one by the texture synthesis of (Efros and Leung, 1999). The resulting two images are finally added to obtain to final reconstructed image. This method gives less artefacts on the contours of the reconstructed objects than simple texture synthesis but it blurs the filling region. Furthermore it stays limited to small image gaps.

Figure 1: Result of the structure/texture decomposition algorithm from (Vese and Osher, 2003). The first column shows the original image, the second column the structure image, and finally the last one the texture image.

3 COMBINING TEXTURE SYNTHESIS WITH DIFFUSION: A NEW ALGORITHM

In this section we present the new algorithm we propose for image inpainting. It is an extension of the algorithm from (Bertalmio et al., 2003) presented in the previous section. It is based on some improvements to most of the steps involved. Therefore before enumerating the whole algorithm, the presentation of these improvements will be given.

3.1 Inpainting the structure image

In (Bertalmio et al., 2003) the so called "image inpainting algorithm" from (Bertalmio et al., 2000) was used to inpaint the structure image. This method was briefly presented in the introduction. It relies on the resolution of a third order PDE that may cause noticeable blur in the result image. Furthermore, this third order PDE needs many iterations to converge and is therefore computationally expensive. We presented, in the same section, another PDE based method (Tschumperlé, 2006) that relies on a second order PDE while rendering less blur. This method gives very satisfactory results even after a small number of iterations. Therefore, the inpainting of the structure image will here be done based on this algorithm which we now describe in further details.

The main idea of geometric approaches is to diffuse the colors along the isophotes which are considered to be the strong edges. The method from (Tschumperlé, 2006) follows the same scheme: the direction of the main structures inside the gap is first computed and an anisotropic diffusion following this geometry is then applied. A particularity here is that the geometry is retrieved using structure tensor fields (Weickert, 1998) and the geometry of the smoothing comes from a second field of diffusion tensors. For a multi-valued image, the structure tensor is defined, for each pixel p, as:

$$J(p) = \sum_{d=1}^{n} \nabla I_d(p) \nabla I_d^T(p), \tag{3}$$

where ∇I_d denotes the gradient image for the d^{th} (color) channel. This tensor is a 2x2 symmetric and semi-positive matrix. Its eigenvalues are positive and can be ordered as: $0 \leq \lambda^-(p) \leq \lambda^+(p)$. The corresponding eigenvectors are denoted as $\theta^-(p)$ and $\theta^+(p)$. These eigenvectors represent two orthogonal directions directed along the local maximum (generally normal) and minimum (generally tangent) variations of image intensities at pixel p. The eigenvalues then measure the effective variations (strength of an edge) of the image intensities along these vectors. The local smoothing geometry that should derive the diffusion process is given by the field of diffusion tensors (Weickert, 1998; Tschumperle and Deriche, 2005). This field, denoted as T, depends on the local geometry and is defined, for each pixel p, as:

$$T(p) = f^-(p)\theta^-(p)\theta^-(p)^T + f^+(p)\theta^+(p)\theta^+(p)^T. \tag{4}$$

The functions f^- and f^+ set the strengths of the desired smoothing along the respective directions $\theta^-(p)$ and $\theta^+(p)$. In (Tschumperlé, 2006), they are defined as:

$$f^-(p) = \frac{1}{(1+\lambda^-(p)+\lambda^+(p))^{p_1}}, \tag{5}$$

and

$$f^+(p) = \frac{1}{(1+\lambda^-(p)+\lambda^+(p))^{p_2}}, \tag{6}$$

where p_1 and p_2 are some input parameters such that $p1 < p2$.

The PDE proposed is derived from an analysis of two regularization PDEs. The first one is the divergence based PDE (Weickert, 1998) and the second one, the trace based PDE (Tschumperle and Deriche, 2005). Therefore, the curvature-preserving smoothing PDE of (Tschumperlé, 2006) that smoothed I along a field of vectors $\mathbf{w}$ is:

$$\frac{\partial I}{\partial t} = \text{trace}\left(\mathbf{w}\mathbf{w}^T \mathbf{H_I}\right) + \nabla I \mathbf{J_w} \mathbf{w} \tag{7}$$

where $\mathbf{J_w}$ is the Jacobian of $\mathbf{w}$. Its goal is to smooth I along $\mathbf{w}$ with locally oriented Gaussian kernels. The algorithm to find the steady state of this equation is clearly enounced in (Tschumperlé, 2006).

3.2 Inpainting the image before decomposing

If the parameters chosen for the decomposition algorithm from (Vese and Osher, 2003) are not very well adapted to the data, only the very strong structures may be kept in the structure image. Then when applying the algorithm from (Tschumperlé, 2006), the tensor of structure may not contain enough information and the inpainting result could be blurred. We have observed this phenomenon on several experiments. Therefore, instead of inpainting only the structure image with a diffusion method, we propose to do it on the original image. The decomposition of the image into a structure and a texture image is only performed afterwards. The obtained structure image is then already inpainted and only a texture synthesis inpainting on the texture image still needs to be performed.

3.3 Inpainting the texture image

In (Bertalmio et al., 2003), the texture image is inpainted with the algorihm from (Efros and Leung, 1999). As we saw in section 1.2, this algorithm can be improved using the method from (Criminisi et al., 2004). Indeed, setting an inpainting order (depending on the edges and the confidence) to the pixels cleary leads to better results. Furthermore, copying an entire patch instead of only one pixel is faster. We here propose some improvements to this algorithm. Some of these improvements require the structure image and therefore texture synthesis will only be applied after all the other processes.

3.3.1 Improving the texture synthesis algorithm

Priorities data term The first improvement concerns the data term in the priorities of (Criminisi et al., 2004). The data term is defined as

$$D(p) = \frac{|\nabla I^{\perp}(p)|}{\alpha} \tag{8}$$

where $\nabla^{\perp}$ is the orthogonal gradient and α is a normalization factor (equal to 255 for grayscale images). It encourages the linear structures to be synthesized first and depends on the isophotes (contours) that eventually pass by p. If we compute this term on the texture image, we will only take into account the small contours and the noise contained in the texture, but not the important edges. It is then better to compute it only on the structure image.

As the structure image is already inpainted at that stage, we can also use a better representation of the contours, *i.e.* of the geometry of the image. Indeed, we can now use the tensors of texture $J(p)$ (equation (3)) to compute the data term. The eigenvalues at pixel p are $\lambda^-(p)$ and $\lambda^+(p)$ and corresponding eigenvectors are $\theta^-(p)$ and $\theta^+(p)$. Using these notations, the new data term is given by:

$$D(p) = \frac{\lambda^+(p) - \lambda^-(p)}{\alpha}. \tag{9}$$

Direction of the search Another change concerns the directions in which the candidate patches are searched for. It can be applied directly to the algorithm from (Criminisi et al., 2004) and does not necessarily require the structure image. Nevertheless, as we have it at this stage, we will use it. The idea is to remark that the best patch $\Psi_{\widehat{p}}$ for the source patch Ψ_p is probably in the direction of the isophotes (the isophote direction is given by the eigenvector $\theta^-(p)$). We then propose to only look for the candidate patches Ψ_q that verify the following test:

$$\theta^-(p) \cdot \frac{p-q}{\|p-q\|} > 0.9. \tag{10}$$

In the original texture synthesis method, finding the candidate patch $\Psi_{\widehat{p}}$ (centered at pixel $\widehat{p}$) corresponds to solving

$$\widehat{p} = \operatorname*{argmin}_{q \in \bar{\Omega}} \sum_{r \in \Psi_p} M(r)(I(r) - I(q+r-p))^2, \tag{11}$$

where d is the sum of square differences (SSD) function. The best patches correspond to the patches that have the smallest associated distance measures. Restricting the direction of the search, eq.(11) becomes:

$$\widehat{p} = \operatorname*{argmin}_{q \in \bar{\Omega} | \theta^-(p) \cdot \frac{p-q}{\|p-q\|} > 0.90} \sum_{r \in \Psi_p} M(r)(I(r) - I(q+r-p))^2. \tag{12}$$

Note that we compute the distance both on the texture and the structure images because the texture image can sometimes only contain non informative noise. We then finally get the following equation:

$$\widehat{p} = \operatorname*{argmin}_{q} \sum_{r \in \Psi_p} M(r) \Big(\big(u(r) - u(r')\big)^2 + \big(v(r) - v(r')\big)^2 \Big), \tag{13}$$

where $r' = r - p + q$.

3.3.2 Texture or structure?

The problem of texture synthesis is that it usually fails at reconstructing regularized structures. In particular some shocks can be visible on connected edges. On the other hand with diffusion methods this problem is almost always solved. Furthermore we currently have at our disposal a texture image inpainted with anisotropic diffusion. Indeed, the two first steps of the complete algorithm are: inpaint the image with (Tschumperlé, 2006) and decompose it. Therefore what we should do is to keep the current intensity values on the important structures and only apply texture synthesis on the other pixels. We then need to decide whether or not a pixel belongs to a strong structure.

The notion of strong structure depends on the force of the gradient which can be characterized by the tensor of structure eigenvalues. We then propose the following test:

$$\lambda^+(p) - \lambda^-(p) < \beta, \tag{14}$$

where β is a threshold equal to the mean:

$$\beta = \frac{\sum_{q\in\bar{\Omega}} \lambda^+(q) - \lambda^-(q)}{|\Omega|}. \tag{15}$$

Of course this threshold is basic and probably not always the best choice and some further work should concentrate on a better automatic estimation.

3.4 Algorithm

We can now sum up the whole algorithm:

1. Inpaint the image with the anisotropic smoothing method from (Tschumperlé, 2006).
2. Decompose the image into a structure and a texture image using the texture-structure image decomposition from (Vese and Osher, 2003)
3. Compute the tensors, the eigenvalues and the eigenvectors on the structure image resulting from step 2.
4. For all the pixels p of the mask, compute the priorities $P(p) = C(p) * D(p)$ (for the others, $p \in \bar{\Omega}$, $P(p) = 0$). The pixels with higher values of P will be inpainted first.
5. Inpaint the texture image:
   (a) Find the pixel $p \in \Omega$ having the highest priority value and that has not been inpainted yet.
   (b) Texture or structure?
      - if $\lambda^+(p) - \lambda^-(p) < \beta$, then apply texture synthesis to the pixel p using equation (13) (in practice one patch is arbitrarily chosen between the best ones as in (Efros and Leung, 1999)).
        Copy image data from $\Psi_{\hat{p}}$ to Ψ_p for all the pixels of $\Psi_p \cap \Omega$.
      - else do not change the pixel value.
   (c) Set $\Omega = \Omega \backslash p$.
   (d) Return to (a).
6. Combine (sum) the inpainted texture image with the structure image resulting from step 2.

3.5 Results

In this section, some results of this new algorithm are presented. They are compared with three of the methods already mentioned in the introduction: the diffusion methods of (Tschumperlé, 2006) and (Bornemann and März, 2007), and the texture synthesis algorithm from (Criminisi et al., 2004). All of these methods require the tuning of some parameters. This is also true for our algorithm since its relies on two of these existing approaches. As suggested in the original papers, we set the contour preservation parameter p_1 equal 0.001, the structure anisotropy $p_2 = 100$, the time step $dt = 150$ and the number of iterations nb equal to 100 for the method from (Tschumperlé, 2006) and for our algorithm. For the one from (Bornemann and März, 2007), we set the averaging radius $\varepsilon = 5$, the sharpness parameter $\kappa = 25$, the scale parameter for pre-smoothing $\sigma = 1.4$ and for post-smoothing $\rho = 4$. The last parameter that has to be set is the patch size required both by our method and by the one from (Criminisi et al., 2004). For all the results presented hereafter, we used patches of size 9x9. We could probably have obtained better results on some of the images by tuning the parameters for each experiments. Nevertheless, we have preferred to show the performance of the method regardless of all these parameters values.

We now present some results on four different images (figure 2). For each figure, the first row represents the original corrupted image, the second one shows in black the mask to be inpainted. Then we are presenting respectively the results from (Tschumperlé, 2006), the ones from (Bornemann and März, 2007) and (Criminisi et al., 2004) and finally the results of our algorithm.

By looking[1] at the results of the diffusion methods (row (c) and (d)), we observe the properties that were already mentioned: these methods permit to obtain regularized contours but are not able to reconstruct the textures. Therefore the resulted images are blurred. Remark that the method from (Tschumperlé, 2006) gives better results than the one from (Bornemann and März, 2007) as it takes into account the global geometry of the image and not only the most important isophotes. When looking at the fifth row, we observe the opposite, textures are pretty well reconstructed but the edges are not continuous. This is particularly visible on the second of the fourth images, for which the gap is bigger.

For all the images, the results obtained with our algorithm are very satisfactory. Indeed, our method combines the advantages of the two other types of approaches. Contrary to all the other techniques, there is not any discontinuity on the boundary of the mask. Furthermore, the textures are well propagated. Nevertheless, all these results are not perfect: the last image still contains a bit of blur. Another drawback can be highlighted from this result. Indeed, one can remark that the result of the first step of the algorithm, that is

[1] The properties of the results can be more easily noticed by zooming on the images.

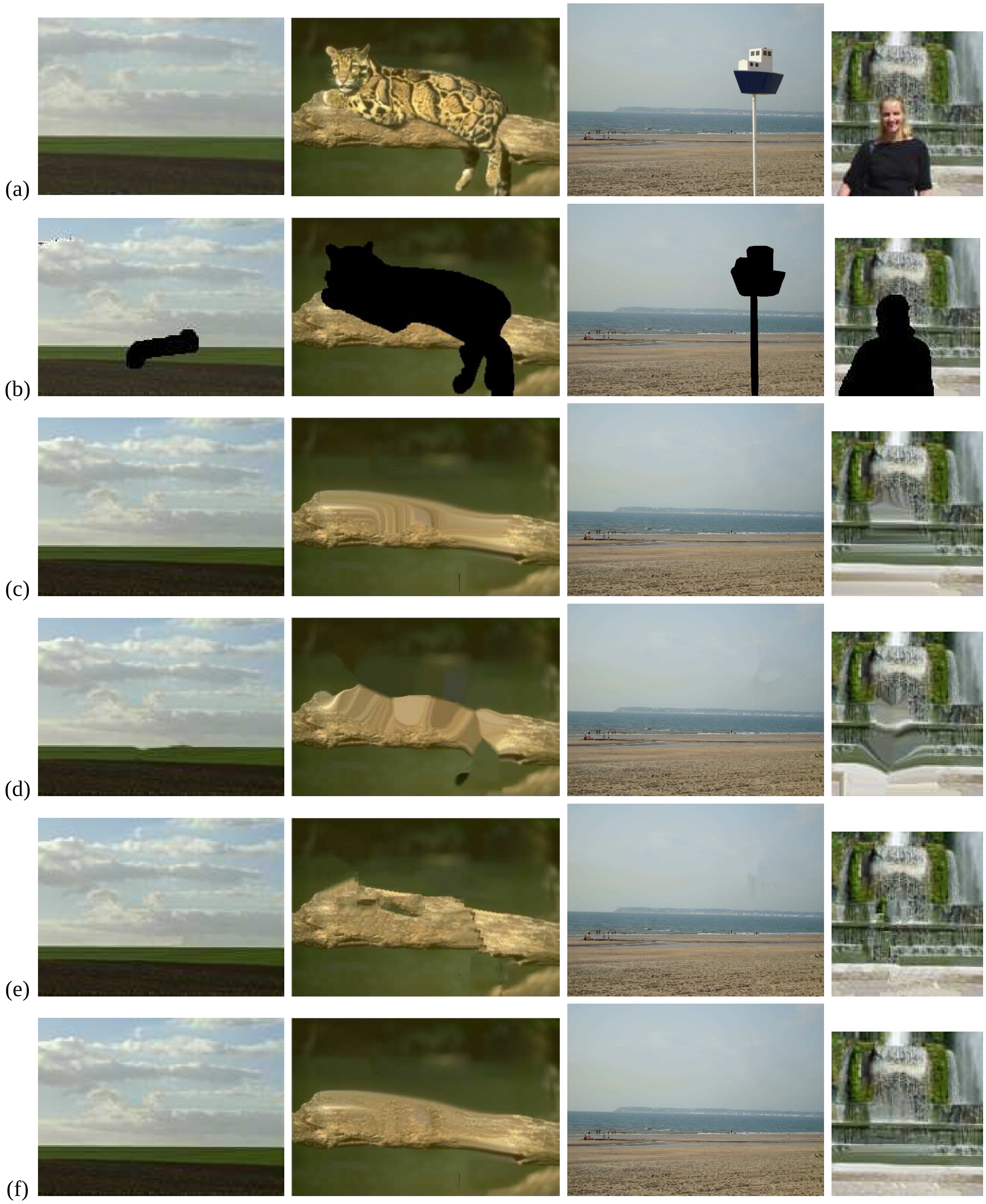

Figure 2: Results of the proposed algorithm on four images. (a) Original image. (b) In black the mask to be inpainted. (c) Results from (Tschumperlé, 2006). (d) Results from (Bornemann and März, 2007). (e) Results from (Criminisi et al., 2004). (f) Results from our algorithm.

the result of the diffusion method from (Tschumperlé, 2006) influences the final result. Therefore, if the result of this method is not satisfactory, the reconstruction given by our algorithm may contain more visual artefacts.

4 CONCLUSION

In this document, a state of the art of diffusion and texture synthesis methods has first been presented. Some chosen algorithms have been described in more detail and their results analysed. From this analysis, it appeared judicious to combine a texture synthesis method with a diffusion algorithm. Therefore, we have proposed an algorithm that combines these two types of approaches. It decomposes the original image into the sum of a texture and a structure image and inpaints each image independently. The inpainted of the structure image is directly obtained with the algorithm from (Tschumperlé, 2006) while for the completion of the texture image we have proposed some extensions of the algorithm from (Criminisi et al., 2004).

Some promising results have been shown. However, the quality of the results may still be improved, as they depend on the diffusion method. Another drawback is the influence of the parameters. The results presented were all obtained with the same parameters. Nevertheless, tuning them automatically taking into account the type of data, would probably improve the quality of our method. This will be the topic of our future research.

ACKNOWLEDGEMENTS

This work was supported by the Torres Quevedo Program of the Ministerio de Educación y Ciencia of Spain and partially founded by Mediapro through the Spanish project CENIT-2007-1012 i3media.

REFERENCES

Ballester, C., Bertalmío, M., Caselles, V., Sapiro, G., and Verdera, J. (2001). Filling-in by joint interpolation of vector fields and gray levels. *IEEE Trans. on Im. Processing*, 10(8):1200–1211.

Bertalmío, M. (2006). Strong-continuation, contrast-invariant inpainting with a third-order optimal pde. *IEEE Trans. on Im. Processing*, 15(7):1934–1938.

Bertalmio, M., Sapiro, G., Caselles, V., and Ballester, C. (2000). Image inpainting. In *SIGGRAPH: ACM Special Interest Group on Comp. Graphics and Interactive Techniques*.

Bertalmio, M., Vese, L., Sapiro, G., and Osher, S. (2003). Simultaneous structure and texture image inpainting. *IEEE Trans. on Im. Processing*, 12(8):882–889.

Bornemann, F. and März, T. (2007). Fast image inpainting based on coherence transport. *Journal of Mathematical Imaging and Vis.*, 28(3):259–278.

Chan, T., Kang, S., and Shen, J. (2002). Euler's elastica and curvature based inpaintings. *Journal of Appl. Math.*, 63:564–592.

Chan, T., Osher, S., and Shen, J. (2001). The digital tv filter and nonlinear denoising. *IEEE Trans. on Im. Processing*, 10(2):231–241.

Chan, T. and Shen, J. (2001). Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Im. Representation*, 12(4):436–449.

Criminisi, A., Pérez, P., and Toyama, K. (2004). Region filling and object removal by exemplar-based inpainting. *IEEE Trans. on Im. Processing*, 13(9):1200–1212.

Efros, A. and Leung, T. (1999). Texture synthesis by non-parametric sampling. In *In Proc. of the Int. Conf. on Comp. Vis.*

Harrison, P. (2001). A non-hierarchical procedure for re-synthesis of complex textures. In *WSCG Conf. Proc.*

Masnou, S. (2002). Disocclusion: a variational approach using level lines. *IEEE Trans. on Im. Processing*, 11(2):68–76.

Masnou, S. and Morel, J. (1998). Level-lines based disocclusion. In *Proceedings of 5th IEEE Intl Conf. on Image Process., Chicago*, volume 3, pages 259–263.

Meyer, Y. (2001). *Oscillating patterns in image processing and nonlinear evolution equations*. American Mathematical Society, Providence, RI.

Mumford, D. (1994). *Algebraic Geometry and its Applications*, chapter Elastica and Computer Vision, pages 491–506. Chandrajit Bajaj, New York, Springer-Verlag.

Sun, J., Yuan, L., Jia, J., and Shum, H. (2005). Image completion with structure propagation. In *SIGGRAPH: ACM Special Interest Group on Comp. Graphics and Interactive Techniques*.

Telea, A. (2004). An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1):23–34.

Tschumperlé, D. (2006). Fast anisotropic smoothing of multi-valued images using curvature-preserving pde's. *Int. Journal of Comp. Vis.*, 68(1):65–82.

Tschumperle, D. and Deriche, R. (2005). Vector-valued image regularization with pdes: a common framework for different applications. *IEEE Trans. on Pat. Analysis and Machine Intelligence*, 27(4):506–517.

Vese, L. and Osher, S. (2003). Modeling textures with total variation minimization and oscillating patterns in image processing. *Journal scientific Computing*, 19(1-3):553–572.

Weickert, J. (1998). *Anisotropic diffusion in image processing*. Teubner Verlag, Stuttgart.