Simultaneous Structure and Texture Image Inpainting

Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, *Member, IEEE*, and Stanley Osher

***Abstract*—An algorithm for the simultaneous filling-in of texture and structure in regions of missing image information is presented in this paper. The basic idea is to first decompose the image into the sum of two functions with different basic characteristics, and then reconstruct each one of these functions separately with structure and texture filling-in algorithms. The first function used in the decomposition is of bounded variation, representing the underlying image structure, while the second function captures the texture and possible noise. The region of missing information in the bounded variation image is reconstructed using image inpainting algorithms, while the same region in the texture image is filled-in with texture synthesis techniques. The original image is then reconstructed adding back these two sub-images. The novel contribution of this paper is then in the combination of these three previously developed components, image decomposition with inpainting and texture synthesis, which permits the simultaneous use of filling-in algorithms that are suited for different image characteristics. Examples on real images show the advantages of this proposed approach.**

***Index Terms*—Bounded variation, filling-in, image decomposition, inpainting, structure, texture, texture synthesis.**

I. Introduction

THE filling-in of missing information is a very important topic in image processing, with applications including image coding and wireless image transmission (e.g., recovering lost blocks), special effects (e.g., removal of objects), and image restoration (e.g., scratch removal). The basic idea behind the algorithms that have been proposed in the literature is to fill-in these regions with available information from their surroundings. This information can be automatically detected as in [5], [10], or hinted by the user as in more classical texture filling techniques [8], [14], [28].

Manuscript received August 8, 2002; revised April 15, 2003. This work was supported in part by the Office of Naval Research, the National Science Foundation, the National Institute of Health, the Office of Naval Research Young Investigator Award, the Presidential Early Career Awards for Scientists and Engineers (PECASE), a National Science Foundation CAREER Award, the National Science Foundation Learning and Intelligent Systems Program (LIS), and the Programa Ramon y Cajal (Ministerio de Ciencia y Tecnologia, Spain). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Ivan W. Selesnick.

M. Bertalmio is with the Department de Tecnologia, University of Pompeu-Fabra, 08003 Barcelona, Spain.

L. Vese and S. Osher are with the Department of Mathematics, University of California, Los Angeles, CA 90095 USA (e-mail: sjo@math.ucla.edu).

G. Sapiro is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455 USA (e-mail: guille@ece.umn.edu).

Digital Object Identifier 10.1109/TIP.2003.815261

The algorithms reported in the literature best perform for pure texture, [10], [14], [28], or pure structure, [2], [3], [5]. This means that for ordinary images such as the one in Fig. 1, different techniques work better for different parts. In [26], it was shown how to automatically switch between the pure texture and pure structure filling-in process. This is done by analyzing the area surrounding the region to be filled-in (inspired by [17]), and selecting either a texture synthesis or a structure inpainting technique. Since most image areas are not pure texture or pure structure, this approach provides just a first attempt in the direction of simultaneous texture and structure filling-in (attempt which was found sufficient for the particular application of transmission and coding presented in the paper). It is the goal of this paper to advance in this direction and propose a new technique that will perform both texture synthesis and structure inpainting in all regions to be filled-in.

The basic idea of our algorithm is presented in Fig. 2, which shows a real result from our approach. The original image (first row, left) is first decomposed into the sum of two images, one capturing the basic image structure and one capturing the texture (and random noise), second row. This follows the recent work by Vese and Osher reported in [30], [31]. The first image is inpainted following the work by Bertalmio-Sapiro-Caselles-Ballester described in [5], while the second one is filled-in with a texture synthesis algorithm following the work by Efros and Leung in [10], third row. The two reconstructed images are then added back together to obtain the reconstruction of the original data, first row, right. In other words, the general idea is to perform structure inpainting and texture synthesis not on the original image, but on a set of images with very different characteristics that are obtained from decomposing the given data. The decomposition is such that it produces images suited for these two reconstruction algorithms. We will show how this approach outperforms both image inpainting and texture synthesis when applied separately.

The proposed algorithm has then three main building blocks: Image decomposition, image (structure) inpainting, and texture synthesis. In the next three sections we briefly describe the particular techniques used for each one of them. As we show in the experimental section, these particular selections, which have been shown to produce state-of-the-art results in each one of their particular applications, outperform previously available techniques when combined as proposed in this paper. In the concluding remarks section we discuss the possible use of other approaches to address each one of these building blocks in order to further improve on the results here reported. In particular, the texture image can be further processed via segmentation (which can also be directly obtained from the

Fig. 1. Example of image with both texture and structure.

decomposition, see Section V and [31]), to further enhance the results of the texture synthesis algorithm.[1]

II. Image Decomposition

In this section, we review the image decomposition approach proposed in [30], [31], which is one of the three key ingredients of the simultaneous texture and structure image inpainting algorithm. As explained in the introduction, this decomposition produces images that are very well suited for the image inpainting and texture synthesis techniques described in the next sections. The description below is adapted from [31], where the technique was first introduced. The interested readers are referred to this work for more details, examples, and theoretical results.

The two main ingredients of the decomposition developed in [31] are the total variation minimization of [27] for image denoising and restoration, and the space of oscillating functions introduced in [23] to model texture or noise.

Let $I: I\!R^2 \to I\!R$ be a given observed image, $I \in L^2(I\!R^2)$. I could be just a noisy version of a true underlying image u, or could be a textured image, u then being a simple sketchy approximation or a cartoon image of I (with sharp edges). A simple relation between u and I can be expressed by a linear model, introducing another function v, such that $I(x,y) = u(x,y) + v(x,y)$. In [27], the problem of reconstructing u from I is posed as a minimization problem in the space of functions of bounded variation $\mathrm{BV}(I\!R^2)$, [12], allowing for edges

$$\inf_{u \in \mathrm{BV}} \left\{ F(u) = \int |\nabla u| + \lambda \|v\|_{L^2}^2, I = u + v \right\} \tag{1}$$

where $\lambda > 0$ is a tuning parameter. The second term in the energy is a fidelity term, while the first term is a regularizing term, to remove noise or small details, while keeping important features and sharp edges.

In [23], Meyer proved that for small λ the model will remove the texture. To extract both the $u \in \mathrm{BV}$ (a piecewise constant or cartoon representation of the image), and the v component as an oscillating function (texture or noise) from I; see Fig. 3, Meyer proposed the use of a different space of functions, which is in some sense the dual of the BV space (and therefore, contains oscillations). The idea is that if (1) (or wavelet-type decompositions) is used, then v will not just contain oscillations, but also undesired brightness edges. Meyer introduced the following definition, and also proved a number of results showing the explicit relationship between the $\| \, \|_*$ norm below and the model in [27] (see [23], [31] for details).

Definition 1: Let G denote the Banach space consisting of all generalized functions $v(x,y)$ which can be written as

$$v(x,y) = \partial_x g_1(x,y) + \partial_y g_2(x,y), \quad g_1, g_2 \in L^\infty(I\!R^2) \tag{2}$$

induced by the norm $\|I\|_*$ defined as the lower bound of all L^∞ norms of the functions $|g|$ where $g = (g_1, g_2), |g(x,y)| = \sqrt{g_1(x,y)^2 + g_2(x,y)^2}$ and where the infimum is computed over all decompositions (2) of I.

Meyer showed that if the v component represents texture or noise, then $v \in G$, and proposed the following new image restoration model:

$$\inf_u \left\{ E(u) = \int |\nabla u| + \lambda \|v\|_*, I = u + v \right\}. \tag{3}$$

In [30] and [31], the authors devised and solved a variant of this model, making use only of simple partial differential equations. This new model leads us to the decomposition we need for simultaneous structure and texture filling-in.

The following minimization problem is the one proposed in [31], inspired by (3)

$$\inf_{u, g_1, g_2} \left\{ G_p(u, g_1, g_2) = \int |\nabla u| + \lambda \int |I - u - \partial_x g_1 - \partial_y g_2|^2 \, dx \, dy + \mu \left[\int \left(\sqrt{g_1^2 + g_2^2} \right)^p dx \, dy \right]^{\frac{1}{p}} \right\} \tag{4}$$

where $\lambda, \mu > 0$ are tuning parameters, and $p \to \infty$. The first term ensures that $u \in \mathrm{BV}(I\!R^2)$, the second term ensures that $I \approx u + \mathrm{div}(g_1, g_2)$, while the third term is a penalty on the norm in G of $v = \mathrm{div}(g_1, g_2)$.

[1]After this paper was accepted for publication, we learned that a number of new papers on inpainting, inspired by [5], will be published at SIGGRAPH 2003 and CVPR 2003.

Fig. 2. Basic algorithm proposed in this paper. The original image in the first row, left (a section of Fig. 1) is decomposed into a structure image and a texture image, [31], second row. Note how the image on the left mainly contains the underlying image structure while the image on the right mainly contains the texture. These two images are reconstructed via inpainting, [5], and texture synthesis, [10], respectively, third row. The image on the left managed to reconstruct the structure (see for example the chair vertical leg), while the image on the right managed to reconstruct the basic texture. The resulting two images are added to obtain the reconstructed result, first row right, where both structure and texture are recovered.

For $p = 1$, as used in this paper, the corresponding Euler-Lagrange equations are [31]

$$u = I - \partial_x g_1 - \partial_y g_2 + \frac{1}{2\lambda}\mathrm{div}\left(\frac{\nabla u}{|\nabla u|}\right) \quad (5)$$

$$\mu\frac{g_1}{\sqrt{g_1^2+g_2^2}} = 2\lambda\left[\frac{\partial}{\partial x}(u-I) + \partial^2_{xx}g_1 + \partial^2_{xy}g_2\right] \quad (6)$$

$$\mu\frac{g_2}{\sqrt{g_1^2+g_2^2}} = 2\lambda\left[\frac{\partial}{\partial y}(u-I) + \partial^2_{xy}g_1 + \partial^2_{yy}g_2\right]. \quad (7)$$

As can be seen from the examples in [31] and the images in this paper, the minimization model (4) allows to extract from a given real textured image I the components u and v, such that u is a sketchy (cartoon) approximation of I, and $v = \mathrm{div}(g_1, g_2)$ represents the texture or the noise (note that this is not just a low/high frequency decomposition). For some theoretical results and the detailed semi-implicit numerical implementation

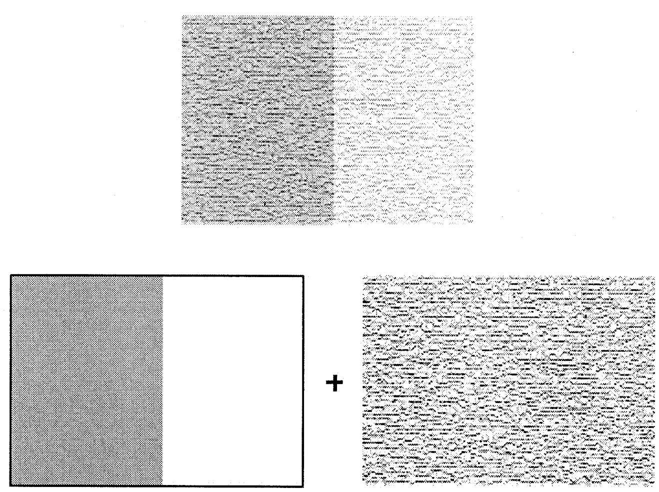


Fig. 3. Illustration of the desired image decomposition. The top image is decomposed in a cartoon type of image (left) plus an oscillations one (right, texture). Note that both images have high frequencies.

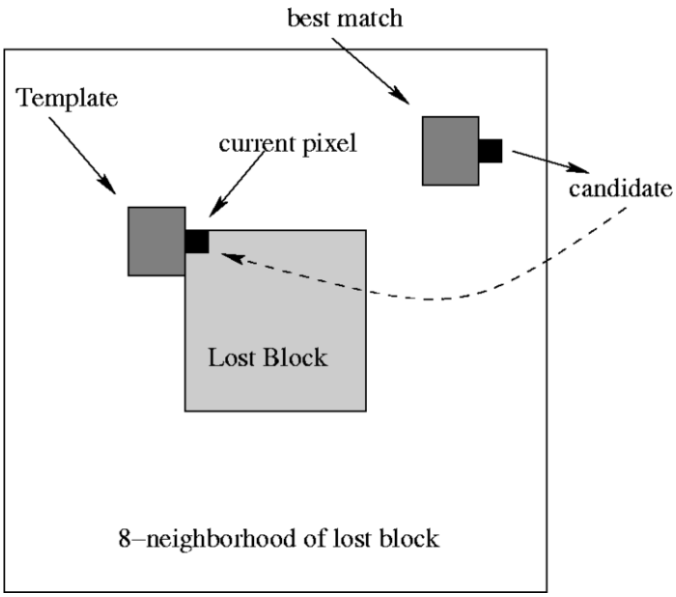


Fig. 4. Basic texture synthesis procedure.

of the above Euler-Lagrange equations, see [31]. We should note that v can be further segmented using information in (g_1, g_2) [31], or from u as in Section V, segmentation that can help the texture synthesis algorithm described below.

III. Texture Synthesis

We now describe the second key component of our scheme, the basic algorithm used to fill-in the region of missing information in v, the texture image. While for the examples in this paper, we use the algorithm developed in [10], this is not crucial and other texture synthesis techniques could be tested for this task. Note however that modulo the selection of a few parameters, this algorithm is fully automatic and produces very good texture synthesis results. Moreover, this algorithm is very well suited to natural images when the regions to be inpainted cover a large variety of textures. These are the basic reasons that lead us to the selection of this particular technique from the vast literature on texture synthesis.

Let the region to be filled be denoted by Ω. Ω will be filled, pixel by pixel, proceeding from the border $\partial\Omega$ inwards. Let I_t be a representative template, with known pixels, surrounding the pixel $p(i, j) \in \Omega$ to be filled-in next. We proceed to find a set of $\hat{I}_t$ from the available neighborhood, such that a given distance $d(I_t, \hat{I}_t)$ is below a pre-defined threshold. As per [10], d is the normalized sum of squared differences (SSD) metric. Once such a set of $\hat{I}_t$'s is found, we randomly chose one of the pixels whose location with respect to $\hat{I}_t$ corresponds to the same position of $p(i, j)$ with respect to I_t. We then fill $p(i, j) \in \Omega$ with the value of this pixel.

The template I_t can be a simple seed-block of 3×3 pixels as shown in Fig. 4. Then, of all 3×3 blocks with fully available data in the image, we look at those closer than a pre-defined threshold to I_t, and randomly pick one. We then replace the current pixel being filled-in in the lost block by the value of the corresponding pixel next to the selected block. This algorithm is considerably faster when using the improvements in [9], [13], [33]. Note also that a segmentation algorithm (to the texture image v) can be added to aid this texture synthesis algorithm.

IV. Image Inpainting

We now describe the third key component of our proposed scheme, the algorithm used to fill-in the region of missing information in the bounded variation image u. For the examples in this paper we use the technique developed in [5]. Other image inpainting algorithms such as [2], [3] could be tested for this application as well. The key idea behind these algorithms is to propagate the available image information into the region to be inpainted, information that comes from the hole's boundary and is propagated in the direction of minimal change (isophotes). We should also note that these works explicitly showed the need for high order partial differential equations for image inpainting (in order to smoothly propagate both gray values on gradient directions), thereby making simpler denoising algorithms such as anisotropic diffusion not appropriate.

Once again let Ω be the region to be filled in (inpainted) and $\delta\Omega$ be its boundary. The basic idea in inpainting is to smoothly propagate the information surrounding Ω in the direction of the isophotes entering $\partial\Omega$. Both gray values and isophote directions are propagated inside the region. Denoting by I the image, this propagation is achieved by numerically solving the partial differential equation (t is an artificial time marching parameter)

$$\frac{\partial I}{\partial t} = \nabla(\Delta I) \cdot \nabla^{\perp} I \tag{8}$$

where ∇, Δ, and $\nabla^{\perp}$ stand for the gradient, Laplacian, and orthogonal-gradient (isophote direction) respectively. This equation is solved only inside Ω, with proper boundary conditions in $\partial\Omega$ for the gray values and isophote directions [5].

Note that at steady state, $(\partial I)/(\partial t) = 0$, and $\nabla(\Delta I) \cdot \nabla^{\perp} I = 0$. This means that ΔI is constant in the direction $\nabla^{\perp} I$ of the isophotes (since $\nabla(\Delta I) \cdot \nabla^{\perp} I$ is just the derivative of ΔI in the direction $\nabla^{\perp} I$), thereby achieving a smooth continuation of the Laplacian inside the region to be inpainted. We have then obtained a smooth propagation of available image information (ΔI) surrounding the hole Ω, propagation done in the direction of minimal change (the isophotes, $\nabla^{\perp} I$).

For details on the numerical implementation of this inpainting technique, which follows the techniques introduced in [21], [27], as well as numerous examples and applications, see [5]. Note in particular that at every numerical step of (8), a step of anisotropic diffusion, [1], [25], is applied [5]. Multiresolution can also be applied to speed-up the convergence [5].

For image inpainting alternatives to this approach, see [2], [3]. In particular, [3] shows the relationship of the above equation with classical fluid dynamics, and presents a different flow

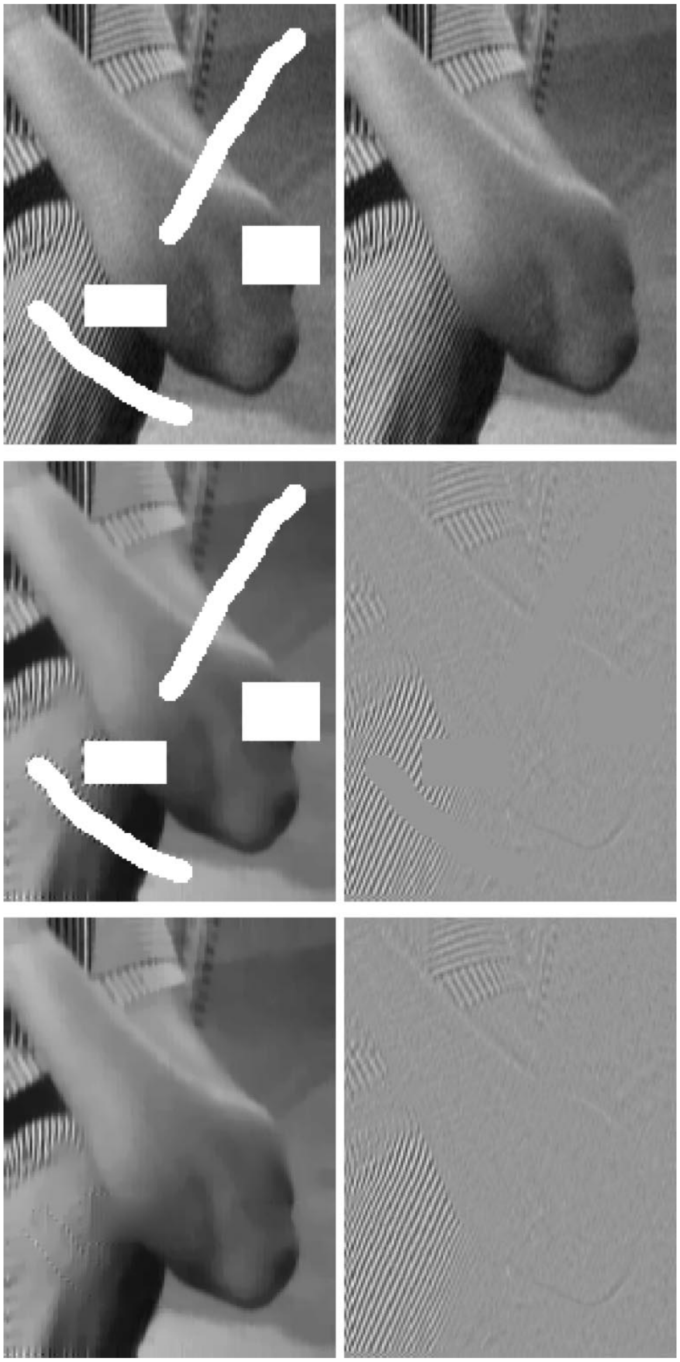


Fig. 5. Additional example, as in Fig. 2.

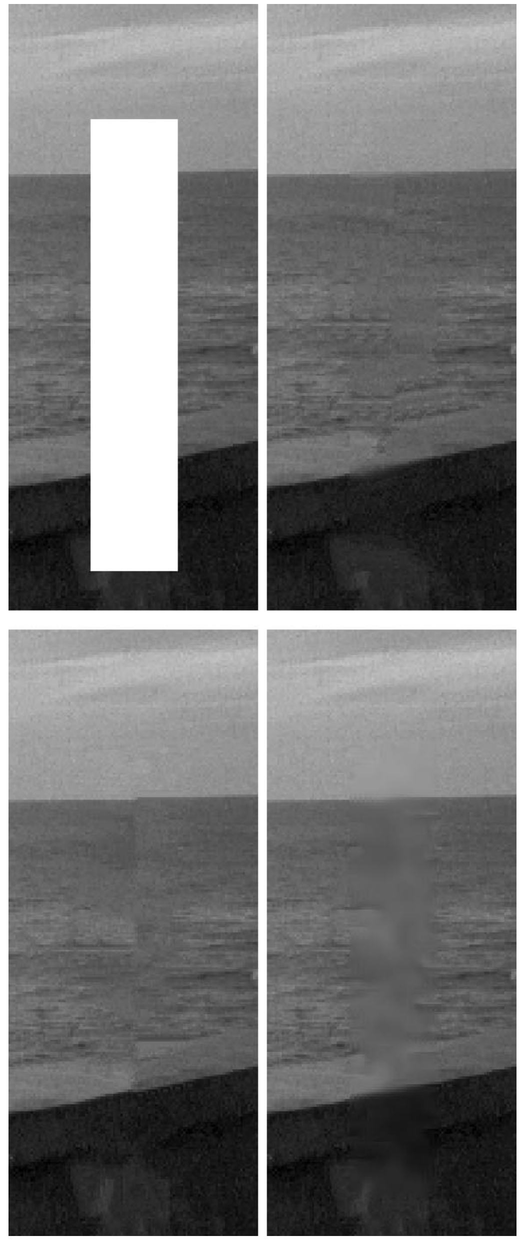


Fig. 6. Comparison of our proposed algorithm with pure texture synthesis and pure image inpainting. Note how our proposed technique manages to reconstruct both texture and structure (second image on first row), while pure texture synthesis fails to reconstruct the structure of the wall and produces artifacts in the water (first image on second row), while pure image inpainting reconstructs the wall but fails with the water (second image on second row).

to achieve the steady state $\nabla(\Delta I) \cdot \nabla^{\perp} I = 0$. The work in [2] presents a formal variational approach that leads to a system of coupled second order differential equations. All these works were in part inspired by [22], [24]. Full details can also be found at http://mountains.ece.umn.edu/~guille/inpainting.htm. Additional related work is described in [7], [15], [16], [19], and [20], while [6], [11], [18], and [32] provide literature on inpainting as done by professional restorators. Comments on these contributions and comparisons with the work just described are provided in [5].

V. Experimental Results

We now present additional experimental results and compare with the case when the image is not decomposed prior to filling-in, and just one algorithm, either image inpainting or texture synthesis, is applied. The initial condition for inpainting is given by running the texture synthesis algorithm. Color is in general treated as in [5] and [10]. While each of the three components of the algorithm here proposed has a number of parameters, all but two of them were left unchanged for all the

Fig. 7. Object removal. The original image is shown on top-left, followed by the result of our algorithm (top-right) and the results with pure texture synthesis (bottom-left), failing to reconstruct the shoulder and introducing artifacts, and pure inpainting (bottom-right), giving a smoother reconstruction.

examples in this paper. The only parameters that vary are λ and the number of steps in inpainting, although the results were found to be very stable to these parameters as well.[2] Texture synthesis can be performed reasonably fast with the extensions in [13], [33], while image inpainting also takes just a few seconds. The overall algorithm takes about 2–3 minutes in a Pentium III, 800 MHz, without any optimization. Most of the computing time is consumed by the texture synthesis algorithm since we are not using any of the speed improvements.

First, in Fig. 5 we repeat the steps as in Fig. 2 for a different portion of the image. Then, in Fig. 6 we compare the results of our algorithm with pure texture synthesis and pure image inpainting. Fig. 7 shows an example of object removal. The last example is presented in Fig. 8, where two different textures are simultaneously reconstructed. The inpainted cartoon image u is used to guide the texture synthesis algorithm. When reconstructing the texture of a given pixel (i, j) (via a straightforward vectorial extension to [10]), only pixels with cartoon value equal to $u(i, j)$ (the value of the cartoon u after inpainting has been performed) are searched. In other words, the inpainted cartoon image is used to provide a rough segmentation. Figs. 6–8 are all in color and can be seen at http://mountains.ece.umn.edu/~guille/inpainting.htm.

[2]For all the images we have used $\mu = 0.1$, the number of numerical steps of the decomposition is equal to 100, and the texture synthesis algorithm uses a 7×7 square template. Regarding the varying parameters, $\lambda = 0.1$ for Figs. 2 and 6 and $\lambda = 0.5$ for the others, while the number of inpainting steps (with a discrete time step of 0.1) are 200 for Figs. 2 and 6 and 2000 for the others (almost identical images were obtained when 2000 steps were used for Fig. 2).

VI. Conclusions and Future Directions

In this paper, we have shown the combination of image decomposition with image inpainting and texture synthesis. The basic idea is to first decompose the image into the sum of two functions, one that can be efficiently reconstructed via inpainting and one that can be efficiently reconstructed via texture synthesis. This permits the simultaneous use of these reconstruction techniques in the image domain they were designed for. In contrast with previous approaches, both image inpainting and texture synthesis are applied to the region of missing information, only that they are applied not to the original image representation but to the images obtained from the decomposition. The obtained results outperform those obtained when only one of the reconstruction algorithms is applied to each image region.

Further experiments are to be carried out to obtain the best combination of image decomposition, image inpainting, and texture synthesis. Since a number of algorithms exist for each one of these three key components, the combination that provides the best visual results is an interesting experimental and theoretical research topic. As mentioned before, an intermediate segmentation step of the texture image will further improve the results (see Fig. 8). Without it, images with large variability in texture types might not be correctly handled by the texture synthesis step. Different parameters selections at the image decomposition stage might also be needed for images containing textures at many different scales. This opens the door to investigate inpainting and texture synthesis combined with an image decomposition step that splits the data into more than two images (e.g., u and a series of v images at different scales).

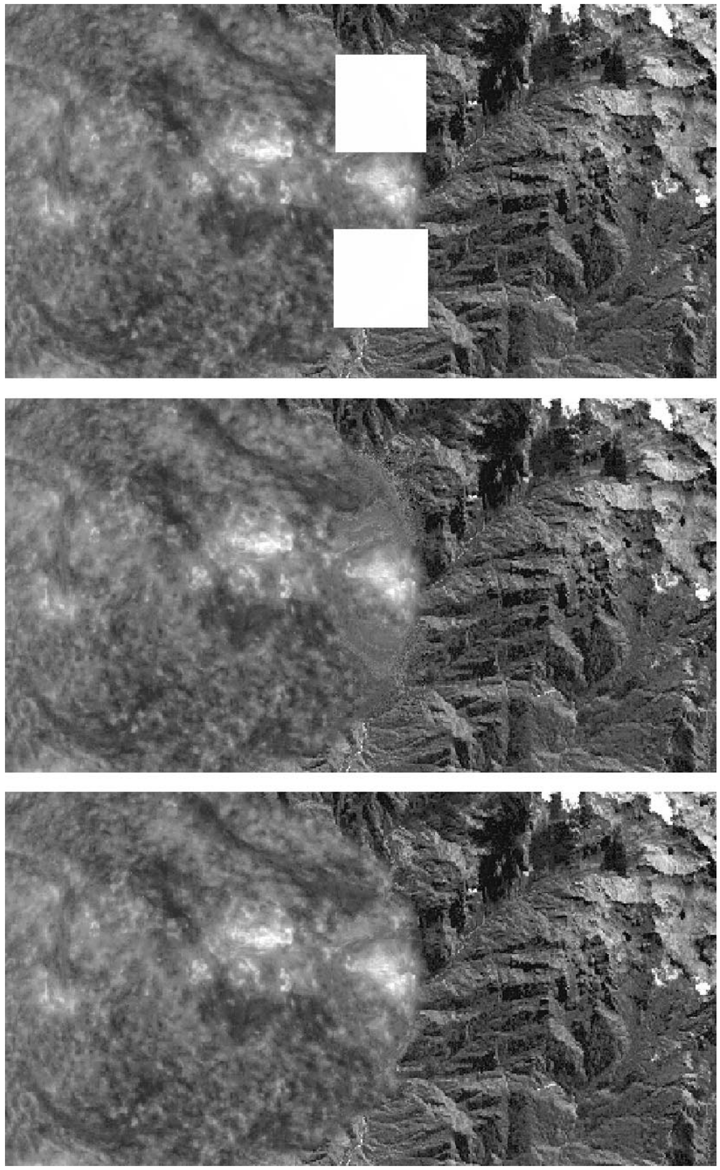


Fig. 8. Inpainting multiple textures. Original image, result of our algorithm, and the result of pure texture synthesis (note the drifting), respectively. Pure inpainting is not designed for this type of data.

We are also currently working on the extension of this work to video and 3-D data, based on the 3-D inpainting techniques developed in [4] and [29].

ACKNOWLEDGMENT

M. Bertalmio and G. Sapiro thank Prof. V. Caselles, with whom this work on image inpainting started. He is a constant source of intellectual and personal inspiration.

REFERENCES

[1] L. Alvarez, P. L. Lions, and J. M. Morel, "Image selective smoothing and edge detection by nonlinear diffusion," *SIAM J. Numer. Anal.*, vol. 29, pp. 845–866, 1992.

[2] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and grey levels," *IEEE Trans. Image Processing*, vol. 10, pp. 1200–1211, Aug. 2001.

[3] M. Bertalmio, A. L. Bertozzi, and G. Sapiro, "Navier-Stokes, fluid dynamics, and image and video inpainting," in *Proc. IEEE Computer Vision and Pattern Recognition (CVPR)*, Dec. 2001.

[4] M. Bertalmio and G. Sapiro, 3D inpainting, . in preparation.

[5] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, "Image inpainting," in *Comput. Graph. (SIGGRAPH 2000)*, July 2000, pp. 417–424.

[6] C. Braverman, *Photoshop Retouching Handbook*. New York: IDG, 1998.

[7] T. Chan and J. Shen, "Local inpainting models and TV inpainting," *SIAM J. Appl. Math.*, vol. 62, no. 3, pp. 1019–1043, 2001.

[8] J. S. De Bonet, "Multiresolution sampling procedure for analysis and synthesis of texture images," in *Proc. ACM SIGGRAPH*, July 1997.

[9] A. A. Efros and W. Freeman, "Image quilting for texture synthesis and transfer," presented at the *Proc. SIGGRAPH*, 2001.

[10] A. A. Efros and T. K. Leung, "Texture synthesis by nonparametric sampling," in *IEEE Int. Conf. Computer Vision*, Corfu, Greece, Sept. 1999, pp. 1033–1038.

[11] *The Restorer's Handbook of Easel Painting*, Van Nostrand Reinhold, New York, 1976.

[12] L. C. Evans and R. F. Gariepy, *Measure Theory and Fine Properties of Functions*. London, U.K.: CRC, 1992.

[13] G. Gorla, V. Interrante, and G. Sapiro, "Growing fitted textures," *IEEE Trans. Visual. Comput. Graphics*, to be published.

[14] D. Heeger and J. Bergen, "Pyramid based texture analysis/synthesis," in *Computer Graphics (SIGGRAPH 1995)*, July 1995, pp. 229–238.

[15] A. Hirani and T. Totsuka, "Combining frequency and spatial domain information for fast interactive image noise removal," in *Computer Graphics, SIGGRAPH 96*, 1996, pp. 269–276.

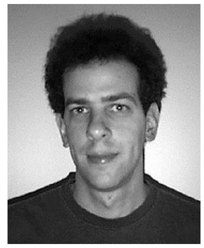
[16] L. Joyeux, O. Buisson, B. Besserer, and S. Boukir, "Detection and removal of line scratches in motion picture films," in *Proc. CVPR'99, IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Fort Collins, CO, June 1999.
[17] K. Karu, A. K. Jain, and R. M. Bolle, "Is there any texture in the image?," *Pattern Recognit.*, vol. 29, no. 9, pp. 1437–1446, 1996.
[18] D. King, *The Commissar Vanishes*. New York: Holt, 1997.
[19] A. C. Kokaram, R. D. Morris, W. J. Fitzgerald, and P. J. W. Rayner, "Detection of missing data in image sequences," *IEEE Trans. Image Processing*, vol. 11, no. 4, pp. 1496–1508, 1995.
[20] ——, "Interpolation of missing data in image sequences," *IEEE Trans. Image Processing*, vol. 11, no. 4, pp. 1509–1519, 1995.
[21] A. Marquina and S. Osher, "Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal," *SIAM J. Sci. Comput.*, vol. 22, pp. 387–405, 2000.
[22] S. Masnou and J. Morel, "Level-lines based disocclusion," presented at the *IEEE Int. Conf. Image Processing*, Oct. 1998.
[23] Y. Meyer, *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, ser. AMS Univ. Lecture Series, 2002, vol. 22.
[24] M. Nitzberg, D. Mumford, and T. Shiota, *Filtering, Segmentation, and Depth*. Berlin, Germany: Springer-Verlag, 1993.
[25] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 629–639, 1990.
[26] S. Rane, G. Sapiro, and M. Bertalmio, "Structure and texture filling-in of missing image blocks in wireless transmission and compression applications," *IEEE Trans. Image Processing*, to be published.
[27] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, no. 60, pp. 259–268, 1992.
[28] E. Simoncelli and J. Portilla, "Texture characterization via joint statistics of wavelet coefficient magnitudes," in *Proc. 5th IEEE Int. Conf. on Image Processing*, 1998.
[29] J. Verdera, V. Caselles, M. Bertalmio, and G. Sapiro. (2003, Jan.) Inpainting Surface Holes. [Online]. Available: http://www.ima.umn.edu/preprints/jan2003/jan2003.html
[30] L. Vese and S. Osher, "Modeling Textures with Total Variation Minimization and Oscillating Patterns in Image Processing,", vol. 02-19, UCLA CAM Rep., May 2002.
[31] ——, "Modeling textures with total variation minimization and oscillating patterns in image processing," *J. Sci. Comput.*, to be published.
[32] S. Walden, *The Ravished Image*. New York: St. Martin's Press, 1985.
[33] L. Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," presented at the *Computer Graphics (SIGGRAPH 2000)*, July 2000.

Marcelo Bertalmio was born in Montevideo, Uruguay, in 1972. He received the B.Sc and M.Sc degrees in electrical engineering from the Universidad de la República, Uruguay, in 1996 and 1998, respectively, and the Ph.D. degree from the University of Minnesota, Minneapolis, in 2001.

He is currently an Associate Professor at Universidad Pompeu Fabra, Spain.

Dr. Bertalmio received the Femlab Prize 2002 for research done as a student in scientific fields related to Prof. Stanley Osher's contributions, the Siemens "Excellent Student Paper Award" at the 2001 IEEE Workshop on variational and level set methods in computer vision, the 2001 Programa Ramón y Cajal by the Spanish Ministry of Science, and several fellowships and scholarships in the U.S. and Uruguay.

Luminita Vese, photograph and biography not available at time of publication.



Guillermo Sapiro (M'95) was born in Montevideo, Uruguay, on April 3, 1966. He received the B.Sc. (summa cum laude), M.Sc., and Ph.D. degrees from the Department of Electrical Engineering, The Technion—Israel Institute of Technology, Haifa, in 1989, 1991, and 1993 respectively.

After postdoctoral research at the Massachusetts Institute of Technology, Cambridge, he became Member of Technical Staff at the research facilities of HP Labs, Palo Alto, CA. He is currently with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis. He works on differential geometry and geometric partial differential equations, both in theory and applications in computer vision, computer graphics, medical imaging, and image analysis. He co-edited a special issue of the *Journal of Visual Communication and Image Representation*. He has authored and co-authored numerous papers and has written a book published by Cambridge University Press in January 2001.

Dr. Sapiro recently co-edited a special issue of the IEEE TRANSACTIONS ON IMAGE PROCESSING. He was awarded the Gutwirth Scholarship for Special Excellence in Graduate Studies in 1991, the Ollendorff Fellowship for Excellence in Vision and Image Understanding Work in 1992, the Rothschild Fellowship for Post-Doctoral Studies in 1993, the Office of Naval Research Young Investigator Award in 1998, the Presidential Early Career Awards for Scientist and Engineers (PECASE) in 1988, and the National Science Foundation Career Award in 1999. He is a member of SIAM.

Stanley Osher received the M.S. and Ph.D. degrees in 1996 from the Courant Institute, New York University.

After working at Brookhaven National Laboratory, the University of California at Berkeley, and the State University of New York (SUNY) at Stony Brook, he joined the University of California at Los Angeles in 1976. He is the coinventor of 1) level set methods for computing moving fronts (4200 references on Google.com), 2) ENO, WENO, and other numerical methods for computing solutions to hyperbolic conservation laws and Hamilton-Jacobi equations, and 3) total variation and other PDE-based image processing techniques. His work has been written up numerous times in the scientific and international media, e.g., *Science News* and *Die Zeit*. He is a highly cited researcher according to Web-of-Science—perhaps the most highly cited researcher in the field of scientific computing.

Dr. Osher has been a Fulbright and Alfred P. Sloan Fellow. He received the NASA Public Service Group Achievment Award, Japan Society of Mechanical Engineers Computational Mechanics Award, and was an invited speaker at the International Congress of Mathematicians.