Simultaneous Structure and Texture Image Inpainting

Marcelo Bertalmio,[*]
Luminita Vese,[†]
Guillermo Sapiro,[‡]
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.

Keywords: Inpainting, filling-in, structure, texture, texture synthesis, bounded variation, image decomposition.

[*] Dept. de Tecnologia, University of Pompeu-Fabra, Pg. de Circumvallacio, 8 (Edifici Francia), 08003 Barcelona, Spain.

[†] Department of Mathematics, University of California, Los Angeles, CA 90095.

[‡] Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55455, USA, guille@ece.umn.edu (corresponding author).

[§] Department of Mathematics, University of California, Los Angeles, CA 90095.

1 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, 9], or hinted by the user as in more classical texture filling techniques [8, 13, 27].

The algorithms reported in the literature best perform for pure texture, [9, 13, 27], or pure structure, [2, 3, 5]. This means that for ordinary images such as the one in Figure 1, different techniques work better for different parts. In [25], 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 [16]), 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.

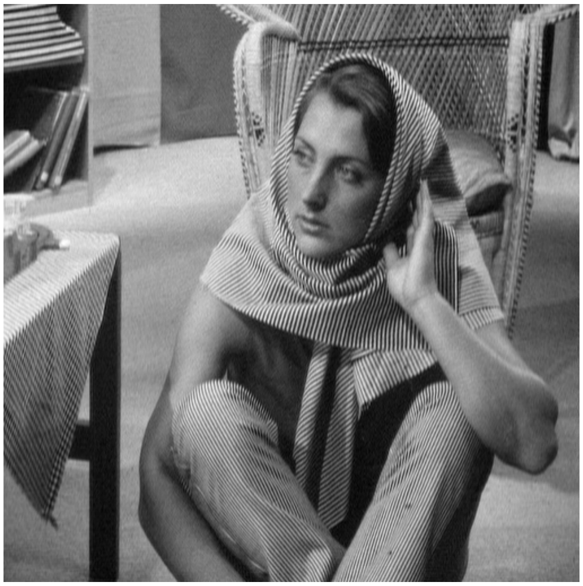

Figure 1: Example of image with both texture and structure.

The basic idea of our algorithm is presented in Figure 2, which shows a real result from our approach. The original image (first row) 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 [28, 29]. 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 [9], third row. The two reconstructed images are then added back together to obtain the reconstruction of the original data, fourth row. 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.

2 Image decomposition

In this section we review the image decomposition approach proposed in [28, 29], 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 [29], 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 [29] are the total variation minimization of [26] for image denoising and restoration, and the space of oscillating functions introduced in [22] 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 [26], the problem of reconstructing u from I is posed as a minimization problem in the space of functions of bounded variation $BV(I\!R^2)$, [11], allowing for edges:

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

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 [22], Meyer proved that for small λ the model will remove the texture. To extract both the $u \in BV$ and the v component as an oscillating function (texture or noise) from I, Meyer proposed the use of a different space of functions, which is in some sense the dual of the BV space. He introduced the following definition, and also proved a number of results showing the explicit relationship between the $\|\|_*$ norm below and the model in [26] (see [22, 29] for details):

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

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

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:

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

In [28, 29], 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 [29], inspired by (3):

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

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

For $p = 1$, as used in this paper, the corresponding Euler-Lagrange equations are [29]

$$u = I - \partial_x g_1 - \partial_y g_2 + \frac{1}{2\lambda} \text{div} \Big(\frac{\nabla u}{|\nabla u|} \Big), \tag{5}$$

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

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

As can be seen from the examples in [29] 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 $v = div(g_1, g_2)$ 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 of the above Euler-Lagrange equations, see [29].

3 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 [9], 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, touching 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 [9], 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 3. 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 [12, 31].

4 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 as [2, 3] could be tested for this application as well.

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, $\frac{\partial I}{\partial t} = 0$, and $\nabla(\Delta I) \cdot \nabla^{\perp} I = 0$. This means that ΔI is constant in the direction $\nabla^{\perp} I$ of the isophotes, thereby achieving a smooth continuation of the Laplacian inside the region to be inpainted.

For details on the numerical implementation of this inpainting technique, which follows the techniques introduced in [20, 26], 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, 24], 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 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 [21, 23]. Full details can also be found at mountains.ece.umn.edu/ guille/inpainting.htm. Additional related work is described in [7, 14, 15, 18, 19], while [6, 10, 17, 30] provides literature on inpainting as done by professional restorators. Comments on these contributions and comparisons with the work just described are provided in [5].

5 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. Color is treated as in [5, 9]. 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 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.[1]

First, in Figure 4 we repeat the steps as in Figure 2 for a different portion of the image. Then, in image Figure 5 we compare the results of our algorithm with pure texture synthesis and pure image inpainting. Figure 6 shows an example of object removal.

6 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.

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

Acknowledgments

This work was partially supported 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, by the National Science Foundation Learning and Intelligent Systems Program (LIS), and by the Programa Ramon y Cajal (Ministerio de Ciencia y Tecnologia, Spain). MB and GS thank Prof. Vicent Caselles, with whom all this work on image inpainting started. He is a constant source of intellectual and personal inspiration.

[1] 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 figures 2 and 5 and $\lambda = 0.5$ for the others, while the number of inpainting steps (with a discrete time step of 0.1) are 200 for figures 2 and 5 and 2000 for the others (almost identical images were obtained when 2000 steps were used for Figure 2).

References

[1] L. Alvarez, P.L. Lions, and J.M. Morel, “Image selective smoothing and edge detection by nonlinear diffusion,” *SIAM J. Numer. Anal.* **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* **10**, pp. 1200-1211, August 2001.

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

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

[5] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester, “Image inpainting,” *Computer Graphics (SIGGRAPH 2000)*, pp. 417–424, July 2000.

[6] C. Braverman, *Photoshop retouching handbook*, IDG Books Worlwide, 1998.

[7] T. Chan and J. Shen, “Local inpainting models and TV inpainting,” *SIAM J. Appl. Math.* **62:3**, pp. 1019-1043, 2001.

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

[9] A. A. Efros and T. K. Leung, “Texture synthesis by non-parametric sampling,” *IEEE International Conference on Computer Vision, Corfu, Greece*, pp. 1033–1038, Sept. 1999.

[10] G. Emile-Male, *The Restorer's Handbook of Easel Painting*, Van Nostrand Reinhold, New York, 1976.

[11] L.C. Evans and R.F. Gariepy, *Measure Theory and Fine Properties of Functions*, CRC Press, London, 1992.

[12] G. Gorla, V. Interrante, and G. Sapiro, “Growing fitted textures,” *IEEE Trans. Visualization and Computer Graphics*, to appear.

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

[14] A. Hirani and T. Totsuka, “Combining frequency and spatial domain information for fast interactive image noise removal,” *Computer Graphics*, pp. 269-276, SIGGRAPH 96, 1996.

[15] L. Joyeux, O. Buisson, B. Besserer, and S. Boukir, “Detection and removal of line scratches in motion picture films,” *Proceedings of CVPR'99, IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Fort Collins, Colorado, USA, June 1999.

[16] K. Karu, A. K. Jain, and R. M. Bolle, “Is there any texture in the image?,” *Pattern Recognition* **29:9**, pp. 1437–1446, 1996.

[17] D. King, *The Commissar Vanishes*, Henry Holt and Company, 1997.

[18] A.C. Kokaram, R.D. Morris, W.J. Fitzgerald, and P.J.W. Rayner, "Detection of missing data in image sequences," *IEEE Transactions on Image Processing* **11:4**, pp. 1496-1508, 1995.

[19] A.C. Kokaram, R.D. Morris, W.J. Fitzgerald, and P.J.W. Rayner, "Interpolation of missing data in image sequences," *IEEE Transactions on Image Processing* **11:4**, pp. 1509-1519, 1995.

[20] 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.*, **22**, pp. 387-405, 2000.

[21] S. Masnou and J. Morel, "Level-lines based disocclusion," *IEEE Int. Conf. Image Processing*, October 1998.

[22] Y. Meyer, *Oscillating Patterns in Image Processing and Nonlinear Evolution Equations*, AMS University Lecture Series **22**, 2002.

[23] M. Nitzberg, D. Mumford, and T. Shiota, *Filtering, Segmentation, and Depth*, Springer-Verlag, Berlin, 1993.

[24] P. Perona and J. Malik "Scale-space and edge detection using anisotropic diffusion," *IEEE-PAMI* **12**, pp. 629-639, 1990.

[25] 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 appear.

[26] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, no. 60, pp. 259–268, 1992.

[27] E. Simoncelli and J. Portilla, "Texture characterization via joint statistics of wavelet coefficient magnitudes," *Proc. 5th IEEE Int'l Conf. on Image Processing*, 1998.

[28] L. Vese and S. Osher, "Modeling textures with total variation minimization and oscillating patterns in image processing," *UCLA CAM Report* **02-19**, May 2002.

[29] L. Vese and S. Osher, "Modeling textures with total variation minimization and oscillating patterns in image processing," *Journal of Scientific Computing*, to appear.

[30] S. Walden, *The Ravished Image*, St. Martin's Press, New York, 1985.

[31] L. Y. Wei and M. Levoy, "Fast texture synthesis using tree-structured vector quantization," *Computer Graphics (SIGGRAPH 2000)*, July 2000.

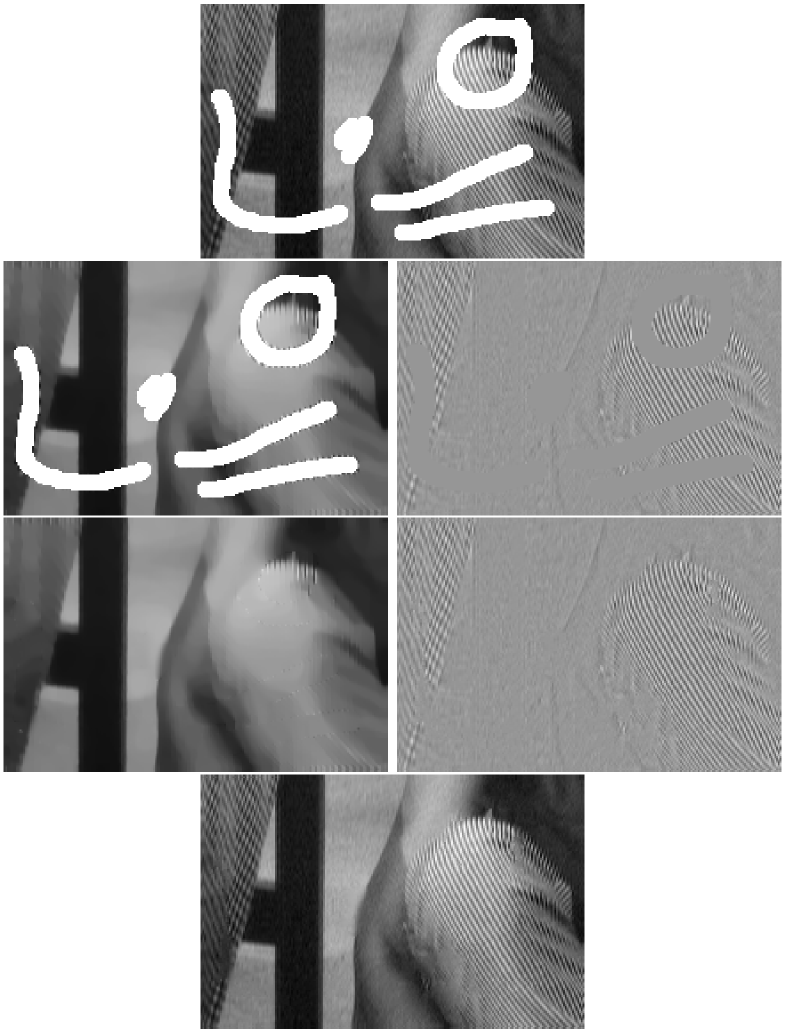

Figure 2: Basic algorithm proposed in this paper. The original image in the first row (a section of Figure 1) is decomposed into a structure image and a texture image, [29], second row. Note how the image on the left mainly contains the underlying image structure while the image on the right mainly contains the texture. These two images are reconstructed via inpainting, [5], and texture synthesis, [9], 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, last row, where both structure and texture are recovered.

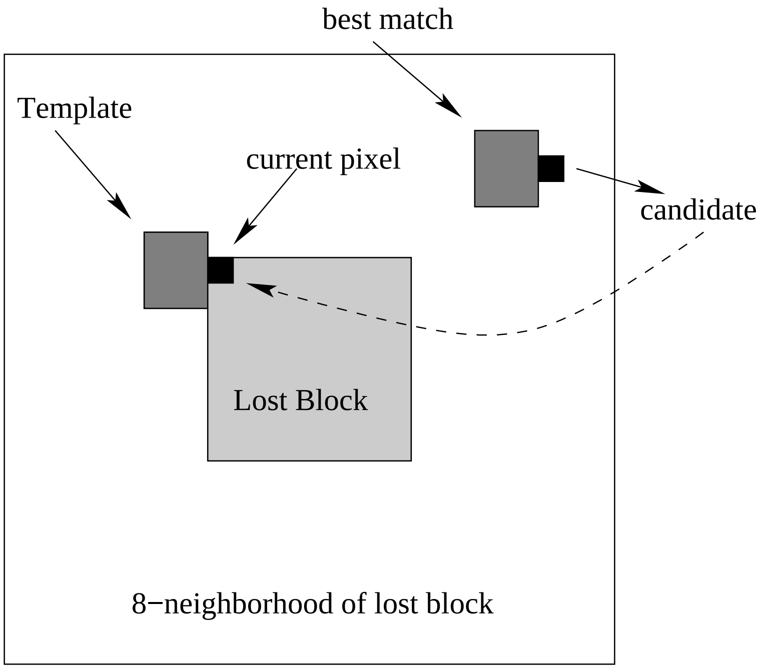

Figure 3: Basic texture synthesis procedure

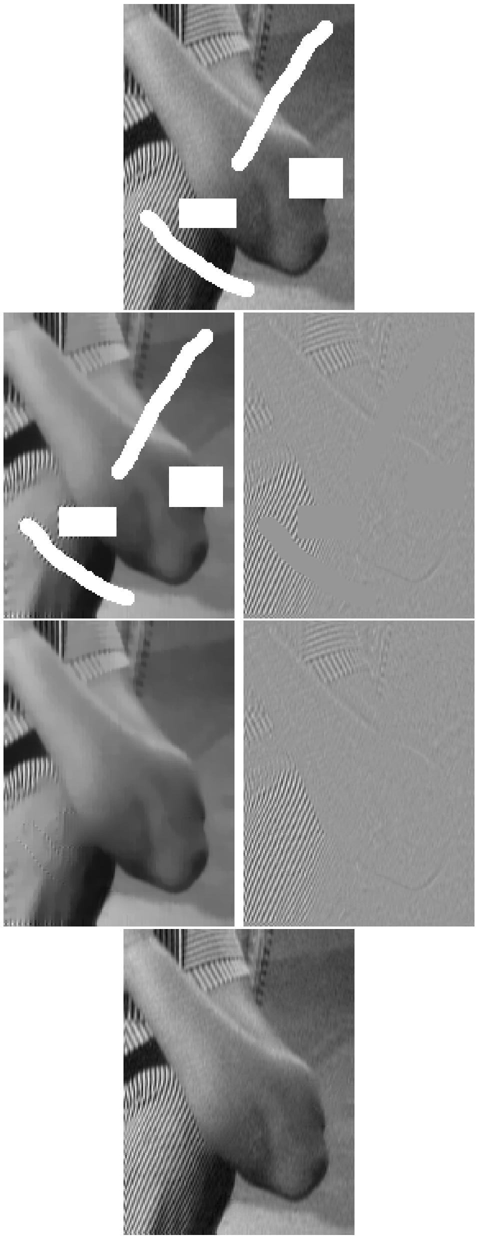

Figure 4: The original image is shown in the first row is decomposed into a structure image and a texture image, second row. These two images are reconstructed via inpainting and texture synthesis, respectively, third row. The resulting two images are added to obtain the reconstructed result, last row, where both structure and texture are recovered.

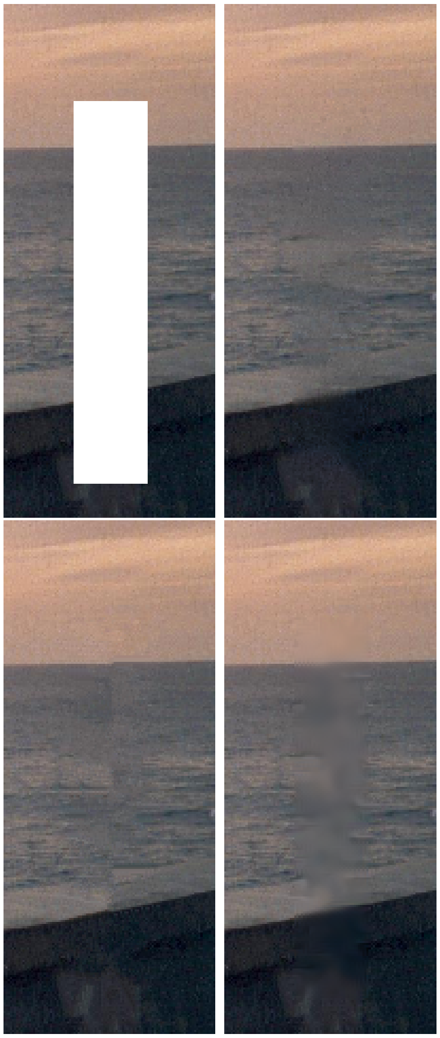

Figure 5: Comparison of our proposed algorithm with pure texture synthesis and pure image inpainting. Note how our proposed technique manages to reconstruct both texture and structure (second image on first row), while pure texture synthesis fails to reconstruct the structure of the wall and produces 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).

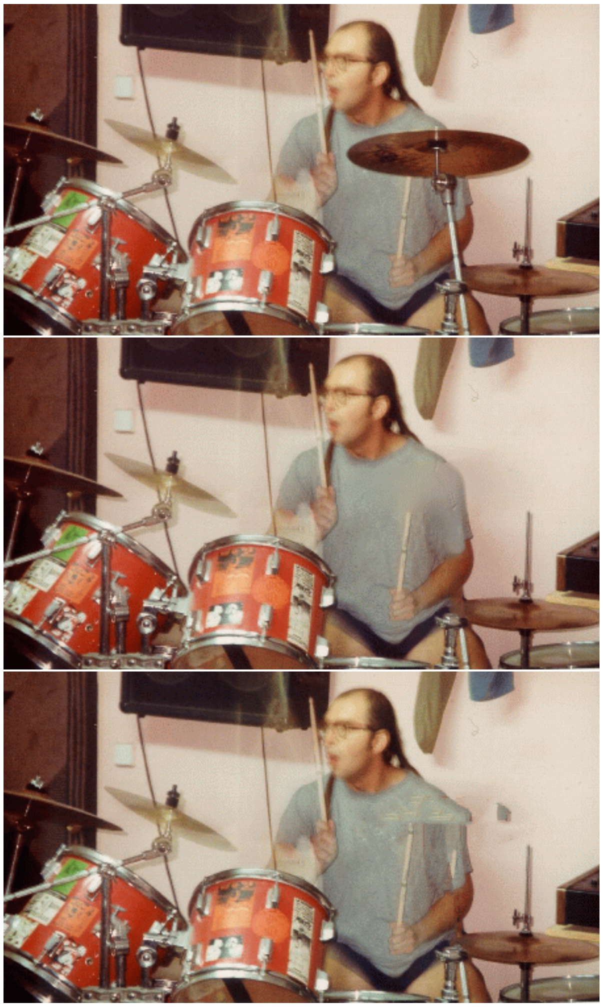

Figure 6: Object removal. The original image is shown on top, followed by the result of our algorithm (middle) and the result with pure texture synthesis (bottom).