Using the Complex Ginzburg–Landau Equation for Digital Inpainting in 2D and 3D

Harald Grossauer and Otmar Scherzer

Department of Computer Science
University of Innsbruck
Technikerstr. 25
A–6020 Innsbruck, Austria
{harald.grossauer,otmar.scherzer}@uibk.ac.at
http://informatik.uibk.ac.at/infmath/

Abstract. Recently, several different approaches for digital inpainting have been proposed in the literature. We give a review and introduce a novel approach based on the complex Ginzburg–Landau equation. The use of this equation is motivated by some of its remarkable analytical properties. While common inpainting technology is especially designed for restorations of two dimensional image data, the Ginzburg–Landau equation can straight forwardly be applied to restore higher dimensional data, which has applications in frame interpolation, improving sparsely sampled volumetric data and to fill in fragmentary surfaces. The latter application is of importance in architectural heritage preservation. We discuss a stable and efficient scheme for the numerical solution of the Ginzburg–Landau equation and present some numerical experiments. We compare the performance of our algorithm with other well established methods for inpainting.

Keywords: Ginzburg–Landau equation, inpainting, diffusion filtering, non–linear partial differential equations, variational problems

1 Introduction

Inpainting is the process of restoration of missing image data; it is typically done by artists. *Digital Inpainting* is performed by computers requiring the user only to mark areas to be inpainted in a digitized image. Digital inpainting has several applications in photography [1], such as scratch removal or retouching. Combining inpainting algorithms with scratch detection algorithms (see e.g. [2] and the references therein) allows to almost automatically restore large sets of degraded images or even complete movies. A difficulty associated with digital inpainting is to set up a measure of visual sensitivity towards defects which can be used in computer code. An attempt in this direction is the *perceptually based physical error metric* introduced by Ramasubramanian, Pattanaik and Greenberg [3]. Today, the common opinion is that the human perceptual system is more sensitive to edges than to texture and most sensitive to junctions; see Caselles, Coll and Morel [4] as a paradigm of this statement in the computer science and

L.D. Griffin and M. Lillholm (Eds.): Scale-Space 2003, LNCS 2695, pp. 225–236, 2003.

mathematical literature. As a consequence a good inpainting algorithm should connect corresponding edges and extrapolate textures smoothly.

In the following we survey some recently proposed inpainting methods based on *level line strategies*, *partial differential equations (PDEs)* and *variational methods*. These methods are most relevant for a comparison with our work. Other topics related to image inpainting such as *texture synthesis* with statistical methods (see e.g. [5,6] and references therein) and image interpolation with *sampling methods* (see e.g. [7,8]) are not considered in this paper.

- An inpainting algorithm based on level lines has been proposed by Masnou and Morel [9,10]. It consists of several stages:
  1. All T–junctions — that are points where level lines hit the boundary of the inpainting domain — are tabulated.
  2. A table of pairs of compatible T–junctions is generated. Two T–junctions are *compatible*, if their associated level lines belong to the same grey level intensity and have the same orientation.
  3. From the set of candidates of level lines connecting compatible T–junctions, the one having the lowest *total generalized elastica energy*

$$\sum \int_{L_{i,j}} (\alpha + \beta |\kappa|^p) ds$$

  is selected. Here $L_{i,j}$ denotes the level line connecting the T–junctions with the indices i and j, and the sum is with respect to all level lines. For convex inpainting domains these are just straight lines.

  The algorithm is computationally expensive: a triangulation of the inpainting domain has to be calculated and an optimal set of level lines out of all possible connections has to be found. Combination with a dynamic programming approach and sorting out inadmissible connections at an early stage keeps runtime complexity relatively low. The implementation presented in [9] does not allow inpainting domains with holes (doughnut shaped inpainting domains).
- Ballester et.al. [11] have proposed a variational method for inpainting. They derive a system of coupled PDEs to extrapolate grey level values and the gradient direction vector field smoothly into the inpainting domain. The system of PDEs is solved using level sets of the image intensity function. This makes the numerical results depend on implementational details, in particular the order in which the level sets are processed (cf. figure 6).
- Bertalmio et.al. [12] introduced an algorithm which imitates the work of manual inpainting of artists. The process is heuristically designed such that an "image smoothness measure" (in their case the image Laplacian) is constantly propagated along the level line direction into the inpainting domain. Numerically, their algorithm corresponds to an explicit finite difference scheme for the partial differential equation

$$\frac{\partial u}{\partial t} = |\nabla u| \left(\nabla \Delta u \cdot \frac{\nabla^\perp u}{|\nabla^\perp u|} \right) .$$

Here $\nabla u = \left(\frac{\partial u}{\partial x_1}, \frac{\partial u}{\partial x_2}\right)$ denotes the gradient, $\nabla^{\perp} u = \left(-\frac{\partial u}{\partial x_2}, \frac{\partial u}{\partial x_1}\right)$ denotes the vector orthogonal to the gradient, and $|\cdot|$ denotes the Euclidean distance. To stabilize the explicit scheme it is combined with a nonlinear diffusion filtering technique for u in the inpainting domain. Bertalmio, Bertozzi and Sapiro [13] have extended this work and embedded the algorithm into a framework of the Navier–Stokes equations.

– Chan and Shen [14,15] use TV–inpainting and solve the differential equation

$$\frac{\partial u}{\partial t} = \nabla \cdot \left(\frac{\nabla u}{|\nabla u|}\right) + \lambda_e \left(u^0 - u\right) \tag{1}$$

up to a stationary point. Here λ_e is an a–priori defined positive function which is zero on the domain to be inpainted. Outside the inpainting domain this equation denoises the image and thus makes the algorithm robust to noise. The inpainted image is composed from the stationary function in the inpainting domain and by the initial image u^0 outside.
Since this TV–inpainting fails to connect long thin structures Chan and Shen [15] propose to replace the diffusion term in (1) by $\nabla \cdot \left(\frac{g(|\kappa|)}{|\nabla u|}\nabla u\right)$, where κ denotes the *curvature* of u. This results in an inhomogeneous third order partial differential equation with forcing term $\lambda_e \left(u^0 - u\right)$.
In [16] Chan and Shen develop an inpainting algorithm based on connecting appropriate level lines by Euler elastica curves. Formulated as a fourth order PDE in terms of the image function u, this algorithm turns out to be a generalization of TV–inpainting and Bertalmio's algorithm, containing both of them as special cases if the involved parameters are set appropriately.
– Esedoglu and Shen [17] combine the Euler elastica functional and the Mumford–Shah functional to derive a fourth order PDE which concurrently does inpainting, denoising and segmentation of the image.
– Oliveira et.al. [18] propose an inpainting algorithm optimized for speed. It consists in repeatedly convolving an arbitrary continuation of u^0 in the inpainting domain with a filter mask. The inpainted image is composed from the convolved image in the inpainting domain and by u^0 outside. This solution behaves very similar to the solution of the linear diffusion equation on the inpainting domain with Dirichlet boundary conditions u^0. As it is well known, linear diffusion tends to blur edges, unless the user manually supplies additional information, e.g. an a–priori segmentation of the inpainting domain.

2 The Ginzburg–Landau Equation

2.1 Motivation

The Ginzburg–Landau equation was originally developed by Ginzburg and Landau [19] to phenomenologically describe phase transitions in superconductors near their critical temperature. The equation has proven to be useful in several distinct areas besides superconduction. It is used to model some types of

chemical reactions like the famous Belousov–Zhabotinsky reaction, to describe boundary layers in multi–phase systems, and to describe the development of patterns and shocks in non–equilibrium systems (see [20,21,22] and references therein).

Solutions of the real valued Ginzburg–Landau equation develop homogeneous areas, which are separated by *phase transition regions*, that are interfaces of minimal area, see the comments following equation (6). In image processing homogeneous areas correspond to domains of constant grey value intensities, and phase transitions to edges. Thus the quoted properties make the real valued Ginzburg–Landau equation a reasonable method for high quality inpainting of binary images.

2.2 Physical Foundations of the Ginzburg–Landau Equation

Investigating the thermodynamic potential of superconductors Ginzburg and Landau derived the following approximation for the corresponding energy functional, depending on the *order function* $u : \Omega \to \mathbb{C}$:

$$F(u, \nabla u) := \frac{1}{2} \int_\Omega \underbrace{|-i\nabla u|^2}_{\text{kinetic term}} + \underbrace{\alpha|u|^2 + \frac{\beta}{2}|u|^4}_{\text{potential term}} \tag{2}$$

where α and β are physical constants. For a nontrivial minimizer to exist $\alpha < 0$ and $\beta > 0$ is necessary. The factor '$-i$' in the kinetic term is a holdover from quantum mechanics and is not essential. The state of minimal energy satisfies the Euler equation of $F(u, \nabla u)$:

$$\frac{\delta F}{\delta u} := \Delta u + \frac{1}{\varepsilon^2}\left(1 - |u|^2\right) u = 0 \,. \tag{3}$$

The equation has been rescaled such that the minima of the potential term function are attained at the sphere $|u| = 1$, which corresponds to the choice $\alpha = -\frac{1}{\varepsilon^2}$ and $\beta = -\alpha$.

In physics ε is called the *coherence length* and correlates to the width of the transition region, that is the width of the transient separating different phases. This can be highlighted in the case when the order function u is one–dimensional and real valued, $u : \mathbb{R} \to \mathbb{R}$, satisfying the boundary condition $\lim_{x \to \pm\infty} u(x) = \pm 1$. In this case an analytical formula for the solution of (3) can be given:

$$u(x) = \frac{e^{\frac{\sqrt{2}}{\varepsilon}x} - 1}{e^{\frac{\sqrt{2}}{\varepsilon}x} + 1} \,. \tag{4}$$

In figure 1 we have plotted $u(x)$ corresponding to different values of ε. It can be realized that the width between the phases ± 1 is approximately 4ε. For $\varepsilon \searrow 0$, $u(x)$ approaches the Heaviside function.

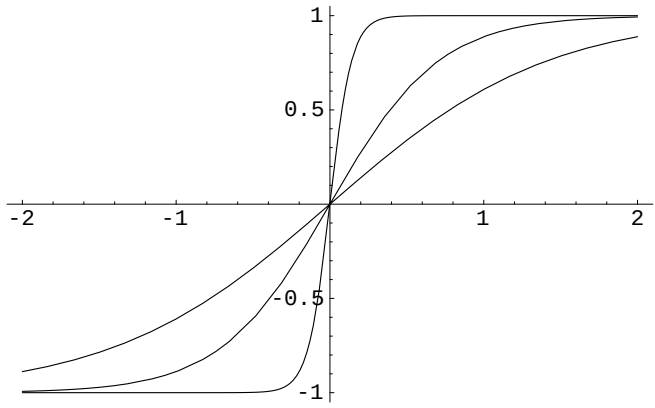

Fig. 1. Solutions of the one–dimensional real valued Ginzburg–Landau equation for $\varepsilon = 1, 0.5, 0.1$

2.3 Algorithm

In the following we describe the use of the Ginzburg–Landau equation for image inpainting of *grey* valued images. To this end we solve the *complex* Ginzburg–Landau equation on the inpainting domain with appropriate boundary data related to the input data.

Preparations. Let D be the domain of the image, usually a rectangular subset of $\mathbb{R}^2$. The *inpainting domain* is denoted by Ω; we assume that it is an open subset of D.

Let $\overline{u}^0 : D \to [-1, 1]$ be the grey–value intensity of an image scaled to the interval $[-1, 1]$; the values -1 and 1 correspond to pure white and black, respectively. The function $\overline{u}^0$ is identified with the real part of a complex valued function $u^0 : D \to \mathbb{C}$ by selecting the imaginary part as

$$\Im(u^0) = \sqrt{1 - \left(\Re(\overline{u}^0)\right)^2} \, .$$

such that $|u^0(x)| = 1$ for all $x \in D$.

A complex valued solution u of (3) will still have an absolute value of 1 almost everywhere but our inpainting (the real part of the solution) may contain any value from the interval $[-1, 1]$. Note that the complex Ginzburg–Landau equation can be considered a system of partial differential equations. Substituting back the solution for the imaginary component into the equation for the real component, we obtain a fourth order equation for the real part.

A common approach to inpaint color images, respectively vector valued images, is to inpaint the (color) components separately. In real world images the color components are typically not independent and a separation approach may lead to artifacts, like spurious colors or rainbow effects. The structure of equation (3) can formally be generalized to vector valued functions $u : D \to \mathbb{C}^n$ in

a straight forward way by replacing the Euclidean distance in (3) by an appropriate norm $\|\cdot\|$ on $\mathbb{C}^n$. For RGB color images we found the maximum norm of the RGB–components to be most appropriate:

$$\|u(x)\| := \max\{|u^1(x)|, |u^2(x)|, |u^3(x)|\} .$$

Note that with this setting the differential equation (3), where $|\cdot|$ is replaced by $\|\cdot\|$, cannot be derived from a variational principle.

Now the problem of inpainting consists in finding $u : \Omega \to \mathbb{C}$ (resp. $\mathbb{C}^3$ for color images) which satisfies equation (3) and the Dirichlet boundary condition $u|_{\partial\Omega} = u^0|_{\partial\Omega}$.

Solving the Equation. To find a solution of equation (3) with Dirichlet boundary condition numerically we use a relaxation method (i.e. a steepest descent method) and solve the differential equation

$$\frac{\partial u}{\partial t} = \Delta u + \frac{1}{\varepsilon^2}\left(1 - |u|^2\right) u \tag{5}$$

up to a stationary point in time.

The reaction–diffusion type equation (5) with real valued u is a variant of the Allen–Cahn equation

$$u_t = \Delta u + \psi'(u). \tag{6}$$

According to [23] equation (6) is called Ginzburg–Landau equation if u is vector valued or complex and the potential $\psi(u)$ has a stable minimum for $|u| = 1$, which is true in our case. In [23] equation (5) with real valued scalar u is used to approximate mean curvature motion. For every ε there exists a unique bounded solution, which — in the limit $\varepsilon \searrow 0$ — consists of sets where either $u = +1$ or $u = -1$ and the interface moves according to mean curvature motion. A lot of mathematical theory about this matter is available, see [23] and references therein. Much less is known though when u is complex or vector valued. A comprehensive study for $u : \mathbb{R}^2 \to \mathbb{C}$ is given in [24].

To numerically integrate equation (5) we use an explicit, forward in time, finite difference scheme (see e.g. [25]). While irregular inpainting domains (which appear in text and scratch removal applications, cf. figures 2 and 3) can easily be handled with explicit schemes, the matrix equations to be solved with *implicit* schemes do not reveal regular structures, are difficult to set up numerically, and cannot be solved efficiently.

In the case of rectangular inpainting domains we have been able to compare the explicit finite difference method with an *implicit nonlinearity lagging* scheme [26], as well as semi–implicit techniques, such as Peaceman–Rachford [26], or a semi–implicit Fourier–spectral method [27], and found that for $\varepsilon \ll 1$ these methods did not give any obvious advantage concerning stability and computation time. This observation is in accordance with [28] where it is argued (though not rigorously proven) that even for implicit schemes a timestep restriction $\delta t < \mathcal{O}(\varepsilon^2)$ is necessary.

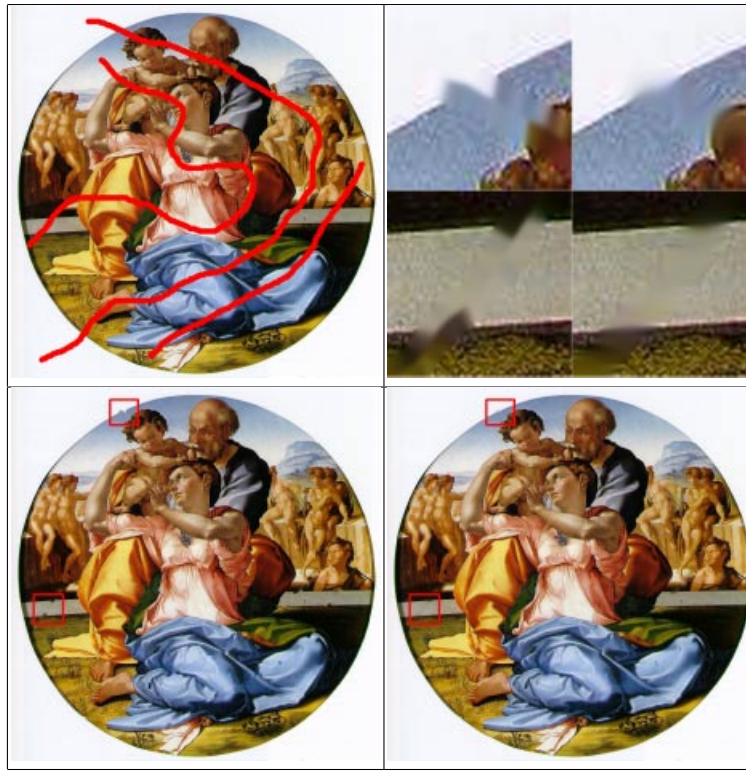

Fig. 2. The painting "Holy Family" from Michelangelo with scratches (*top left*). The scratches have been inpainted with the plain Ginzburg–Landau algorithm (*bottom left*). The picture shows the result of the same algorithm interleaved with some steps of coherence enhancing diffusion (*bottom right*). Detailed views of the red framed parts are compared (*top right*)

Fig. 3. Lenna inpainting

Discretizing (5) in space and time, the explicit finite difference method has the following form:

$$u_{i,j}^{t+1} = u_{i,j}^t + \delta t \cdot \left(\Delta u_{i,j}^t + \frac{1}{\varepsilon^2} \left(1 - \|u_{i,j}^t\|^2\right) u_{i,j}^t \right) \tag{7}$$

where

$$\Delta u_{i,j} = u_{i-1,j} + u_{i+1,j} + u_{i,j-1} + u_{i,j+1} - 4u_{i,j} \ .$$

Here $u_{i,j}$ denotes the color vector intensity at the pixel point (i, j). As initial value we set $u^0|_\Omega = 0$. Equation (7) has been rescaled to get rid of the influence of space discretization. According to what was said before we have to choose a timestep $\delta t < \varepsilon^2$, which in particular shows that the time steps have to be extremely small for $\varepsilon \ll 1$, which is required for high contrast inpainting purposes. The iteration process (7) is stopped at time $\bar{t}$ if

$$\max_{i,j} \{ |u_{i,j}^{\bar{t}} - u_{i,j}^{\bar{t}-1}| \}$$

drops below a certain threshold.

Postprocessing. The solution of the Ginzburg–Landau equation reveals high contrast in the inpainting domain, which makes it particularly suited for inpainting purposes. However, the level lines of the solution of the Ginzburg–Landau equation at the boundary of the inpainting domain might look kinky. In general this cannot be considered a bad habit as figure 4 shows, but in certain pictures (for instance in artistic drawings) this may look disturbing. For such applications we suggest to apply a couple of coherence enhancing diffusion steps (see e.g. [29]) to steer the direction of inpainting. In figure 2 we have inpainted a scratched digitized painting by Michelangelo. The inpainting with the Ginzburg–Landau equation is of high contrast, but the level lines look kinky as the enlargements of details in figure 2 show. The kinks can be smoothed via coherence enhancing diffusion.

2.4 Three Dimensional Inpainting

The Ginzburg–Landau equation can be generalized to any number of space dimension. Thus in particular it can be applied to inpaint three dimensional grey valued image intensity functions $u : \mathbb{R}^3 \to \mathbb{R}$; for example allowing optical improvement of sparsely sampled data or frame interpolation (treating a movie as a stack of images). The generalization for inpainting of three dimensional vector valued data (e.g. color images) is straight forward.

Inspired by the work of Davis et. al. [30] we also applied our algorithm for completion, respectively continuation, of 2D–surfaces, which are represented as zero level sets of auxiliary functions $u : \mathbb{R}^3 \to \mathbb{R}$. A typical settings for the auxiliary function is the signed distance function to the surface or by setting $u = \pm 1$ on opposite sides of the surface. Then the inpainting algorithm is applied to a volume containing the missing parts of the surface. The missing surface

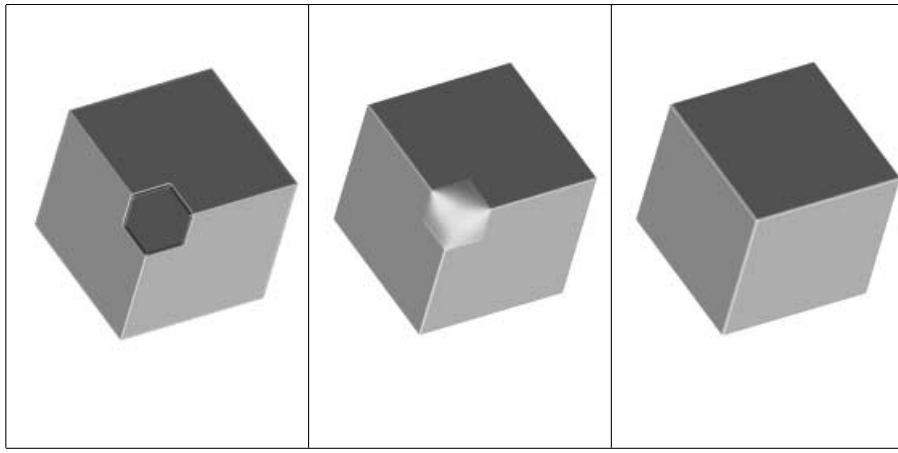

Fig. 4. A corner of the cube was manually cut out (*left*). The completion attained from the linear diffusion equation resembles a membrane stretched across the edges (*middle*). With the Ginzburg–Landau equation a perfect corner is achieved (*right*)

is completed by the zero level set of the solution. We view an example of our algorithm applied to a synthetic three dimensional object in figure 4. The missing corner of the cube was manually cut out. The middle picture shows the result of surface completion using the linear diffusion equation for inpainting. As expected the resulting surface looks like a membrane stretched over the existing edges. In contrast the Ginzburg–Landau inpainting algorithm develops a hard corner. It depends on the individual application whether one prefers smooth and soft surfaces or hard edges and corners. By tuning the value of ε Ginzburg–Landau inpainting allows the user to choose any degree of "smoothness".

A more realistic application is shown in figure 5 where a hole in the left cheekbone has been filled. Using this kind of processing for medical data is quite dangerous but could be useful for refinement of data obtained in heritage recording projects [31].

3 Results and Conclusion

Results of our inpainting algorithm for text removal are presented in figure 3. In the Lenna picture 28779 pixels (which is about 11% of the image) have been overdrawn with white color. The inpainting was finished within a few seconds on a 1.5 GHz Pentium 4 PC running unoptimized C++ code under Linux. On a first look no visual deficiencies can be seen in the inpainted image. Closer examination reveals some fringes and kinks, most notable at the bottom edge of the hat. Combination with a couple of coherence enhancing diffusion steps (cf. subsection "Postprocessing" in section 2.3) can reduce these disturbing effects.

Further numerical experiments have shown that best results are obtained for *locally small* inpainting areas which makes our approach well suited for removing

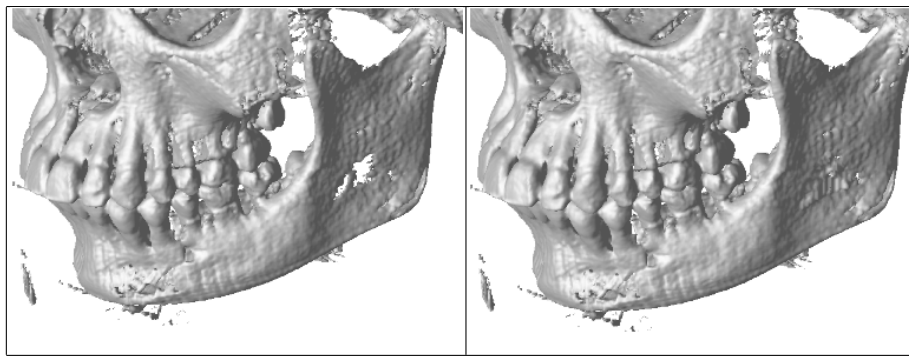

Fig. 5. The hole in the cheekbone has been filled

cracks or superimposed texts. By locally small we mean that the Hausdorff distance between Ω and $\partial\Omega$

$$d(\Omega) = \sup_{x\in\Omega} \inf_{y\in\partial\Omega} (|x-y|)$$

is smaller than the typical size of the image structures in the surrounding of Ω. This is intuitive since the intensity information can only be extended "reliably" from a given pixel into a small neighborhood. The inpaintings produced with the algorithms outlined in this paper differ by the contrast. Some reveal blurry inpainting away from the boundary like the Gaussian heat flow. The approach of Masnou & Morel produces high contrast inpaintings with straight or polygonal level lines, thus revealing artifical (desired or not) kinks. Our approach produces high contrast inpaintings as well. Using the parameter $\varepsilon > 0$ we are able to compromise between blurry and high contrast models, which can be used to weaken the visibility of unwanted kinks.

Our approach, as well as the inpainting algorithms outlined in section 1, is not applicable to inpaint textured regions. In such cases texture synthesis or combined texture–synthesis/inpainting algorithms are more appropriate, see [32] and references therein. See also [33] for a short comparison between local inpainting and texture synthesis approaches.

To juxtapose the results of the Ginzburg–Landau equation we discuss a frequently cited example in the area of inpainting: how should the noisy area in the first image of figure 6 be filled? There are many reasonable inpaintings since it is a synthetic image allowing no intuitive interpretation. Level line based algorithms are usually designed to establish short and smooth connections. Depending on the concrete numerical implementation the level set method automatically selects either the third or the fourth picture — the level set formulation itself does not have the information which inpainting to choose. From a mathematical point of view the Ginzburg–Landau inpainting (second picture) is most appropriate to retain the symmetry of the initial image.

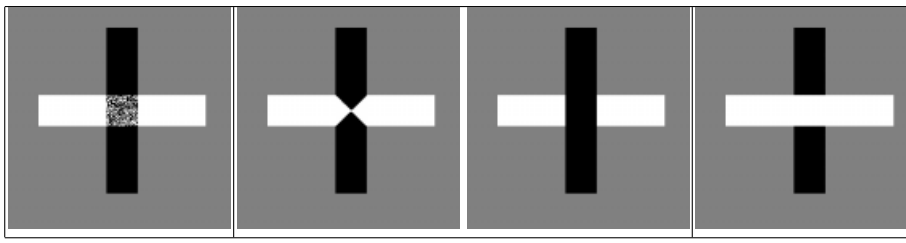

Fig. 6. Ambiguous example: (*first picture*) the noisy area should be inpainted, (*second picture*) inpainting via the Ginzburg–Landau algorithm, (*third and fourth picture*) inpainting via level set algorithm

Acknowledgement

This work is supported by the Austrian Science Fund (FWF), grant Y–123 INF.

References

1. *Lückenfüller und Farbmischer, Bildkorrekturverfahren: beim Menschen gelernt*, c't–Magazin für Computer Technik, 24/2002, Heise Zeitschriften Verlag, Hannover, p.190
2. L. Joyeux, O. Buisson, B. Besserer, 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
3. M. Ramasubramanian, S. Pattanaik, D. Greenberg, *A Perceptually Based Physical Error Metric for Realistic Image Synthesis*, Proceedings of SIGGRAPH 99. In Computer Graphics Proceedings, Annual Conference Series, 1999, ACM SIGGRAPH, p.73–82
4. V. Caselles, B. Coll, J. Morel, *A Kanizsa programme*, Progress in Nonlinear Differential Equations and their Applications, Vol. 25, p.35–55, 1996
5. A.A. Efros and T.K. Leung, *Texture synthesis by non–parametric sampling*, Proceedings of the Seventh International Conference on Computer Vision, Corfu, Greece, 1999
6. H. Igehy, L. Pereira, *Image Replacement through Texture Synthesis*, Proceedings of the 1997 IEEE International Conference on Image Processing
7. M. Unser, *Sampling–50 years after Shannon*, Proceedings of the IEEE, vol. 88, no. 4, p.569–587, April 2000
8. P. Thenaz, T. Blu, M. Unser, *Handbook of Medical Imaging, Processing and Analysis*, I.N. Bankman, Ed., Academic Press, San Diego CA, USA, p.393–420, 2000
9. S. Masnou, *Disocclusion: A Variational Approach Using Level Lines*, IEEE Transactions on Signal Processing, 11(2), February 2002, p.68–76
10. S. Masnou, J.-M. Morel, *Level Lines based Disocclusion*, Proceedings of the 1998 IEEE International Conference on Image Processing, p.259–263
11. C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, J. Verdera, *Filling–In by Joint Interpolation of Vector Fields and Gray Levels*, IEEE Transactions on Signal Processing, 10(8), August 2001, p.1200–1211

12. M. Bertalmio, G. Sapiro, C. Ballester, V. Caselles, *Image inpainting*, Computer Graphics, SIGGRAPH 2000, July 2000
13. M. Bertalmio, A. Bertozzi, G. Sapiro, *Navier–Stokes, Fluid Dynamics, and Image and Video Inpainting*, IEEE CVPR 2001, Hawaii, USA, December 2001
14. T. Chan, J. Shen, *Mathematical Models for Local Nontexture Inpaintings*, SIAM Journal of Applied Mathematics, 62(3), 2002, p.1019–1043
15. T. Chan, J. Shen, *Non–Texture Inpainting by Curvature–Driven Diffusions (CDD)*, Journal of Visual Communication and Image Representation , 12(4), 2001, p.436–449
16. T. Chan, S. Kang, J. Shen, *Euler's Elastica and Curvature Based Inpainting*, SIAM Journal of Applied Mathematics, 63(2), pp. 564–592, 2002
17. S. Esedoglu, J. Shen, *Digital Inpainting Based on the Mumford–Shah–Euler Image Model*, European Journal of Applied Mathematics, 13, pp. 353-370, 2002
18. M. Oliveira, B. Bowen, R. McKenna, Y. Chang, *Fast Digital Inpainting*, Proceedings of the International Conference on Visualization, Imaging and Image Processing (VIIP 2001), Marbella, Spain, pp. 261–266
19. L. Landau, V. Ginzburg, *On the Theory of Superconductivity*, Journal of Experimental and Theoretical Physics (USSR), 20 (1950), p.1064
20. M. Ipsen, P. Sorensen, *Finite Wavelength Instabilities in a Slow Mode Coupled Complex Ginzburg–Landau Equation*, Physical Review Letters, Vol. 84/11, p.2389, 2000
21. M. van Hecke, E. de Wit, W. van Saarloos, *Coherent and Incoherent Drifting Pulse Dynamics in a Complex Ginzburg–Landau Equation*, Physical Review Letters, Vol. 75/21, p.3830, 1995
22. T. Bohr, G. Huber, E. Ott, *The structure of spiral–domain patterns and shocks in the 2D complex Ginzburg–Landau equation*, Physica D, Vol. 106, p.95–112, 1997
23. L. Ambrosio, N. Dancer, *Calculus of Variations and Partial Differential Equations*, Springer Verlag
24. F. Bethuel, H. Brezis, F. Hélein, *Ginzburg–Landau Vortices*. In "Progress in Nonlinear Differential Equations and Their Applications", 13 (1994), Birkhäuser
25. W. Press, B. Flannery, S. Teukolsky, W. Vetterling, *Numerical Recipes*, Cambridge University Press
26. J. Thomas, *Numerical Partial Differential Equations: Finite Difference Methods*, Texts in Applied Mathematics, Vol. 22
27. L. Chen, J. Shen, *Applications of semi–implicit Fourier–spectral methods to phase field equations*, Computer Physics Communications, Vol. 108, p.147–158, 1998
28. R. H. Nochetto, M. Paolini, C. Verdi, *A Dynamic Mesh Algorithm for Curvature Dependent Evolving Interfaces*, Journal of Computational Physics, 123, 1996, p.296–310
29. J. Weickert, *Anisotropic Diffusion in Image Processing*, B.G.Teubner, Stuttgart, 1998
30. J. Davis, S. Marschner, M. Garr, M. Levoy, *Filling holes in complex surfaces using volumetric diffusion*, to appear in First International Symposium on 3D Data Processing, Visualization, and Transmission Padua, Italy, June 19–21, 2002
31. J. Taylor, *Demonstration of Canadian 3D Technology for Heritage Recording in China*, Proceedings of the Italy–Canada 2001 Workshop on 3D Digital Imaging and Modeling Applications, Padova, Italy, April 3–4, 2001
32. M. Bertalmio, L. Vese, G. Sapiro, S. Osher, *Simultaneous Structure and Texture Image Inpainting*, submitted
33. http://www.people.fas.harvard.edu/˜hchong/Spring2002/cs276r/