

Fast Image Inpainting Based on Coherence Transport

Folkmar Bornemann · Tom März

Published online: 31 August 2007
© Springer Science+Business Media, LLC 2007

Abstract High-quality image inpainting methods based on nonlinear higher-order partial differential equations have been developed in the last few years. These methods are iterative by nature, with a time variable serving as iteration parameter. For reasons of stability a large number of iterations can be needed which results in a computational complexity that is often too large for interactive image manipulation.

Based on a detailed analysis of stationary first order transport equations the current paper develops a fast *non-iterative* method for image inpainting. It traverses the inpainting domain by the fast marching method just once while transporting, along the way, image values in a coherence direction robustly estimated by means of the structure tensor. Depending on a measure of coherence strength the method switches continuously between diffusion and directional transport. It satisfies a comparison principle. Experiments with the inpainting of gray tone and color images show that the novel algorithm meets the high level of quality of the methods of Bertalmio et al. (SIG-GRAPH '00: Proc. 27th Conf. on Computer Graphics and Interactive Techniques, New Orleans, ACM Press/Addison-Wesley, New York, pp. 417–424, 2000), Masnou (IEEE Trans. Image Process. 11(2):68–76, 2002), and Tschumperlé (Int. J. Comput. Vis. 68(1):65–82, 2006), while being faster by at least an order of magnitude.

Keywords Image inpainting · Disocclusion · Hyperbolic equation · Eikonal equation · Skeleton · Coherence direction · Structure tensor · Fast marching

1 Introduction

Nontexture *image inpainting*, also termed image interpolation, is the task of restoring the values of a digital image for a destroyed, or consciously masked, subregion of the image domain. It thus belongs to the area of digital image processing. On the other hand, if we consider a region that partially occludes some objects of the image the related computational task of making these objects fully visible is termed *disocclusion* in the area of image analysis.

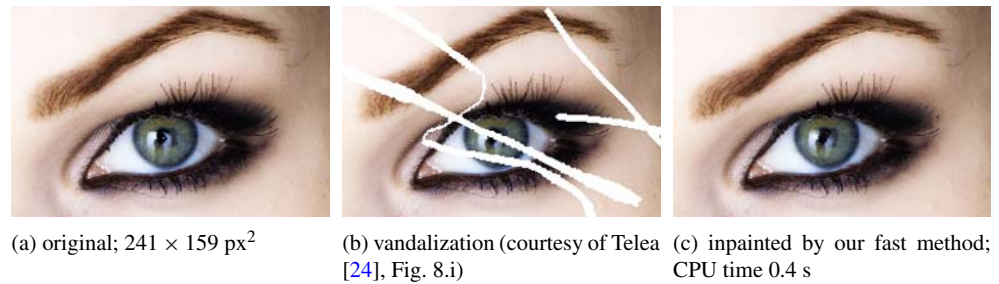
The qualifying term *nontexture* refers to the case that *local* features and *short range* correlations of the image are sufficient for a result of high perceptive quality. There is no detection of more *global* structures like symmetry or *long range* spatial correlations perceived as patterns and textures. If the task at hand required such global information, one would have to rely on the methodology of pattern recognition and texture synthesis. It is also not possible to restore semantic content of an image, such as given by the physiology of the shown object or the physics of the illumination.

Nevertheless, impressive results are obtainable by nontexture image inpainting as illustrated, e.g., in Fig. 1. The inpainting result (using the method of this paper) that is shown in Fig. 1(c) is of such a high quality that a nonexpert observer might consider it as the original image. Note that even the flow-like (local) texture of the eyebrow has been restored in a plausible way. However, a look at the original shown in Fig. 1(a) reveals that some information such as

F. Bornemann (✉) · T. März
Zentrum Mathematik, Technische Universität München,
Boltzmannstr. 3, 85747 Garching, Germany
e-mail: bornemann@ma.tum.de

T. März
e-mail: maerzt@ma.tum.de

Fig. 1 A digital image (a), its vandalization (b), and its reconstruction by inpainting¹ (c)



reflections on the pupil have not been restored, making the inpainted result appear less vivid.

The State of Affairs Considerable interest, and excitement, has been generated for the inpainting problem in the applied mathematics community by the celebrated papers of Masnou and Morel [17] and Bertalmio et al. [3]. These papers have shown that sophisticated mathematical tools such as variational principles and partial differential equations can be fruitful here. Many interesting applications of the inpainting methodology have been documented since then, such as the restoration of old photographs, removal of superimposed text or selected objects, digital zoom-in, and edge decoding. For a survey of the mathematics involved, of many methods and applications, see [8], Chap. 6.

At a first sight, it might appear surprising that tools from continuous mathematics bear power on solving problems for discrete objects such as digital images. However, the abstraction of a continuous, analog image that is only approximated by the digital one allows for a genuine handling of the multiscale nature of images (see [14], Fig. 20.1). The high-level of abstraction that is given by the tools of continuous mathematics considerably contributes to the understanding of the problems of image processing and of the discrete algorithms used. But it also contributes to the construction of new discrete algorithms. This approach to digital image processing is elaborated, e.g., in the books of Aubert and Kornprobst [1], Chan and Shen [8], Guichard and Morel [14], Kimmel [15], Sapiro [19], and Weickert [27].

We call the abstraction of the underlying continuous image the *high-resolution limit* u of the discrete image u_h , with h being a measure of the fineness of the resolution.² Note, that the technological progress can in fact be described as realizing $h \rightarrow 0$: professional photographers already use digital cameras that make images of a size of 4992×3328 px², giving a resolution of 832 dpi for a 6'' \times 4'' photo print. Such fine resolutions are barely met in day-to-day simulations with numerical partial differential equations. Certainly,

¹Using the default parameters (23).

²Here, to be specific, we take h being related to the resolution by $h^{-1} = \sqrt{\# \text{ pixels}}$.

some engineers would consider such grids as “numerically converged”.

Basically, there are two major mechanisms to get image information into the inpainting domain: diffusion and transport (see [6]). For the inpainting problem, nonlinear partial differential equations combining these mechanisms have been obtained phenomenologically (e.g., [3] or [25]), axiomatically (e.g., [4]), or from a variational principle (e.g., [5, 7, 17], or [11]). The resulting numerical algorithms are *iterative* by nature and the iteration is often realized by the introduction of a time variable that is discretized by an explicit time stepping scheme. Thus, a CFL stability condition applies, restricting the time step to $\tau \propto h^\nu$, where the exponent ν is, e.g., $\nu = 1$ for advection type equations, but can be as large as $\nu = 4$ for fourth order parabolic equations. Reaching the stationary state of the inpainted result requires a number of iterations (time steps) as large as

$$\# \text{ iterations} \propto (\# \text{ image pixels})^{\nu/2}.$$

For instance, the transport equation underlying the method of Bertalmio [3] has $\nu = 1$. For the inpainting problems of Figs. 12 and 13 we thus predict a number of iterations of order 10^3 . In fact, Bertalmio et al. ([3], p. 420) report to have used 3000 time steps to inpaint these problems. On the other hand, the numerical scheme given by Chan et al. ([5], §6.1) for their *Elastica* variational inpainting method has $\nu = 4$ (see [12], §6.3). Thus, for the reproduction of the example found in [5], Fig. 6.4, which is basically a 140×32 px² detail of the inpainting problem in Fig. 12, we predict a necessary number of iterations of order 10^7 . In fact, [12], Fig. 6.9, reports to have used 12 000 000 time steps to reproduce the published result. Consequently, in their recent survey on variational image inpainting Chan and Shen ([9], p. 614) list as one major open problem the “fast and efficient digital realization” of these methods.

A notable exception from the CFL stability restriction is the recent method of Tschumperlé [25], which basically calls for the solution of heat equations along flow lines of the coherence vector field. He obtains an unconditionally stable scheme by line convolution with Gaussians.

It is not surprising that no high-quality inpainting method has become established so far in commercial, interactive im-

age manipulation software, such as Photoshop. The plug-in of Tschumperlé’s [25] method to the GNU software *The GIMP* is a pioneering step in this direction.

The Contribution of our Paper Our aim in this paper is to develop an inpainting algorithm that matches the high level of quality of the methods by Bertalmio et al. [3] and Tschumperlé [25] while being considerably faster.

Our novel algorithm is based on two observations. The first observation is about the celebrated method of Bertalmio et al. [3] which is, neglecting some anisotropic diffusion steps that are interleaved for stabilization, a time stepping method for the transport equation

$$u_t = \nabla^\perp u \cdot \nabla L, \quad L = \Delta u.$$

The phenomenological rationale ([3], §3.2) usually given for this equation goes as follows (see also [19], §8.2.2, or [8], §6.11.2): a measure of smoothness L of the image u is transported along the field of isophotes induced by the vector field $\nabla^\perp u$. However, since u_t but not L_t appears on the left hand side, it is *not* a transport equation for L . Instead, by writing the equation in the equivalent form

$$u_t = -\nabla^\perp \Delta u \cdot \nabla u$$

we see that it really is a transport equation for the image value u being convected by the field $\nabla^\perp \Delta u$ (see also [2], p. 357). This field is in the direction of the level lines of Δu which are related to Marr–Hildreth edges (see Sect. 4). Thus, the formal stationary state

$$\vec{c} \cdot \nabla u = 0, \quad \vec{c} = \nabla^\perp \Delta u,$$

transports the image values along continuations of edges from the boundary of the inpainting domain into its interior. Of course, this transport along characteristics will cross somewhere. However, if we are lucky, crossing characteristics might carry similar information. (Like two teams digging a tunnel from both ends are meant to meet somewhere.) Thus, well-posedness of such hyperbolic boundary problems is a central issue in our paper (see Theorem 3).

The second observation occurred while we analyzed the discrete, linear, and fast noniterative inpainting method of Telea [24]. This method traverses the pixels of the inpainting domain from the boundary to its interior by following the fast marching method for the approximation of the distance map of the boundary. Along the way local weighted averages of already known values propagate the image values into the inpainting domain. This fast method behaves strongly diffusive (see Fig. 3) and creates peculiar transport patterns (see Fig. 4). We have developed a deeper mathematical understanding of these effects by studying what we

call the high-resolution, vanishing viscosity limit of the algorithm (see Theorem 1). In fact, it turns out that the algorithm approximates the stationary transport equation

$$\vec{n} \cdot \nabla u = 0,$$

where \vec{n} denotes the field of normals to the level lines of the distance map, that is, the field of steepest ascent from the boundary. This transport direction has already been identified by Bertalmio et al. ([3], p. 419) as being an unsuccessful choice for the propagation of image information.

Our two observations are tied together by the question whether it is possible to modify the weight function of Telea’s algorithm (keeping its fast performance) such as to obtain the formal stationary state of Bertalmio et al.’s method as limit equation (keeping its high level of quality). We will answer this question positively, up to a well understood exceptional case (that is, the last case in (14)), in Theorem 2.

Finally we increase the robustness of the method by replacing the edge-oriented transport direction of Bertalmio et al.’s [3] method by the coherence direction. This second moment information has successfully been used by Weickert [27, 29] for the enhancement of coherent information in anisotropic diffusion or shock-filtering. As we have already mentioned, the coherence flow also underlies the inpainting method of Tschumperlé [25].

Outline of our Paper In Sect. 2 we recall the details of Telea’s [24] fast single-pass algorithm. We carefully separate specific choices from the generic architecture of the algorithm and show that it satisfies strong stability properties. In Sect. 3 we establish the high-resolution, vanishing viscosity limit of the algorithm. The modification of the weight function for matching a given field of directions is subject of Sect. 4. Here, an exceptional case is related to the well-posedness of the limit equation. The choices of flow fields that are suitable for image inpainting are discussed in Sect. 5. We will show that the noniterative approach of this paper requires a subtle modification of the coherence flow to avoid undesirable boundary effects. Implementation details, such as an updating formula for the structure tensor and the treatment of color images, are given in Sect. 6. Finally, in Sect. 7 we show computational results for some synthetic and natural inpainting problems. In particular, a comparison with the methods of Bertalmio et al. [3], Masnou [16], and Tschumperlé [25] will show that we meet the high level of quality of these methods while being at least an order of magnitude faster. We also demonstrate the application of the inpainting methodology to the denoising of images affected by impulse noise with a noise frequency as large as 80%.

Notation Because amplitude quantization is not an issue in this paper, we generally assume that all gray tone images,

or channels of color images, take values in the real interval $[u_{\min}, u_{\max}]$. We will distinguish between *discrete* (digital) images defined on finite sets of pixels and *continuous* (analog) images defined on open subsets of \mathbb{R}^2 . The latter are thought of as the high-resolution limit of the former. This distinction will be indicated by using the index h for the discrete notions while omitting it for the corresponding continuous ones. Pixels will be identified with their midpoint.

- Ω_h is the *image domain*, i.e., the finite array of pixels for the final, restored image $u_h : \Omega_h \rightarrow [u_{\min}, u_{\max}]$.
- $D_h \subset \Omega_h$ is the *inpainting domain* for which the values of u_h have to be determined.
- $\Omega_h \setminus D_h$ is the *data domain*, for which the values of the image are given as $u_h^0 = u_h|_{\Omega_h \setminus D_h}$.
- $\partial D_h \subset D_h$ is the discrete boundary, that is, the set of inpainting pixels that have at least one neighbor (with respect to the chosen 4- or 8-connectivity, see ([23], §2.6.2)) in the data domain.

Continuous quantities are defined correspondingly. Finally, we define discrete and continuous ϵ -neighborhoods by

$$B_{\epsilon,h}(x) = \{y \in \Omega_h : |y - x| \leq \epsilon\},$$

$$B_\epsilon(x) = \{y \in \Omega : |y - x| \leq \epsilon\},$$

where $|\cdot|$ denotes the Euclidean norm.

2 Telea’s Single-Pass Algorithm

Telea [24] has presented a fast algorithm for image inpainting that calculates u_h by traversing the pixels of the inpainting domain in just a single pass. Basically, the simple idea is to fill the inpainting domain in a fixed order, from its boundary inwards, by using weighted means of already calculated image values. Here, the order depends on the geometry of the inpainting domain, but not on the data image u_h^0 to be inpainted.

To prepare our mathematical analysis and modification of Telea’s approach in later sections, we break down his algorithm into three aspects which are logically quite independent from each other. The first is the generic frame of the algorithm, the second and third are specific choices made by Telea.

The Generic Single-Pass Algorithm We number the pixel of the inpainting domain according to the chosen order, $D_h = \{x_1, \dots, x_N\}$, and call

$$B_{\epsilon,h}^<(x_k) = B_{\epsilon,h}(x_k) \setminus \{x_k, \dots, x_N\}, \quad k = 1, \dots, N,$$

the neighborhood of already inpainted pixels. Mathematically, the single-pass algorithm now reads generically as fol-

lows:

$$u_h|_{\Omega_h \setminus D_h} = u_h^0, \tag{1}$$

$$u_h(x_k) = \frac{\sum_{y \in B_{\epsilon,h}^<(x_k)} w_h(x_k, y) u_h(y)}{\sum_{y \in B_{\epsilon,h}^<(x_k)} w_h(x_k, y)}, \quad k = 1, \dots, N. \tag{2}$$

Here, $w_h(x, y) \geq 0$ are called the *weights* of the algorithm and we assume that

$$\sum_{y \in B_{\epsilon,h}^<(x)} w_h(x_k, y) > 0, \quad x \in D_h.$$

Obviously, because of the weighted means this algorithm is linear, for fixed w_h , in the data image u_h^0 and enjoys the following desirable stability properties:

- *Comparison Principle* If the data image satisfies $u_{\min} \leq u_h^0 \leq u_{\max}$ with two constants u_{\min} and u_{\max} , the inpainting result satisfies the same inequalities

$$u_{\min} \leq u_h \leq u_{\max}.$$

- *l_∞ -stability* Two different data images u_h^0 and \hat{u}_h^0 yield inpainting results u_h and \hat{u}_h satisfying

$$\|u_h - \hat{u}_h\|_\infty \leq \|u_h^0 - \hat{u}_h^0\|_\infty,$$

where the maximum defining the norm $\|\cdot\|_\infty$ is taken over Ω_h and D_h , respectively.

Note that the comparison principle still holds if w_h depends on u_h .

The Order Telea [24] realizes the intuitive idea of filling the inpainting domain progressively from its boundary inwards by ordering the pixels x of D_h according to their (approximate) Euclidean distance $T_h(x)$ to the boundary ∂D_h :

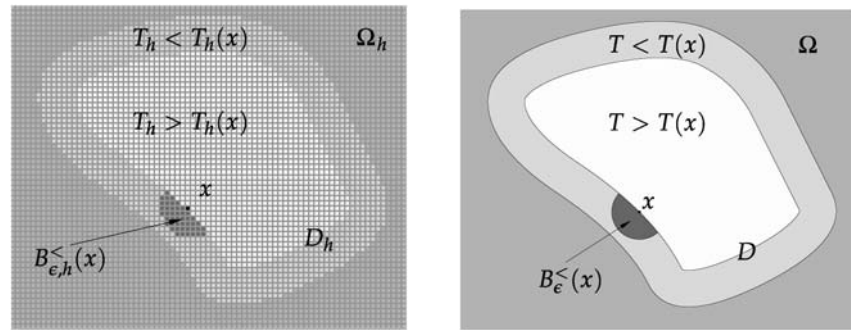
$$T_h(x_j) < T_h(x_k) \Rightarrow j < k. \tag{3}$$

The geometric idea of the generic single-pass algorithm with this distance ordering is illustrated in Fig. 2(a).

Using a priority queue data structure, such as a heap, the function T_h can be calculated on D_h in the order of increasing values using $O(N \log N)$ operations. This fundamental approach, which is similar to Dijkstra’s [10] algorithm for finding shortest paths in weighted graphs, was found for the problem at hand independently in different areas of applied mathematics:

- For the fast calculation of Euclidean geodesic distance maps in mathematical morphology [22], see also [23], §7.4.
- For the fast calculation of value functions of certain optimization problems satisfying Bellman’s principle [26]. The distance map is a special instance.

Fig. 2 The geometric idea of the generic single-pass inpainting algorithm with distance ordering



(a) geometric setting of the actual, discrete algorithm

(b) geometric setting of the high-resolution limit (6)

- For the fast calculation of level-set discretizations of monotonically advancing fronts [20], see also [21], §8 or [15], §7. The level sets of the Euclidean distance maps are instances of such fronts.

We will adopt the last point of view, called *fast marching method* in the literature. Here, T_h is obtained as an upwind discretization of the Euclidean distance map to the boundary of the continuous inpainting domain, $T(x) = \text{dist}(x, \partial D)$. This function is the unique viscosity solution of the Dirichlet problem for the eikonal equation,

$$|\nabla T| = 1 \quad \text{in } D, \quad T|_{\partial D} = 0. \tag{4}$$

The Weights Telea [24] suggests a variety of weights. However, they do not yield significantly different results in practice, and all give the same high-resolution limit that we will discuss in the next section. Therefore, we just take the simplest instance

$$w_h(x, y) = \frac{|\nabla_h T_h(x) \cdot (x - y)|}{|x - y|^2},$$

where ∇_h denotes a central difference approximation of the gradient. This weight approximates the continuous one³

$$w(x, y) = \frac{|\nabla T(x) \cdot (x - y)|}{|x - y|^2} = \frac{|\cos \angle(\vec{n}(x), x - y)|}{|x - y|}, \tag{5}$$

$$\vec{n}(x) = \nabla T(x),$$

where by (4) $\vec{n}(x) = \nabla T(x)$ is the unit vector (normal to the level lines) that points into the direction in which the inpainting progresses. The rationale behind this choice is twofold: an already inpainted pixel y in the vicinity of x contributes to the value of $u_h(x)$ with a larger weight if, by the denominator, it is closer to x and, by the numerator, it lies in the

³The (nonoriented) angle between two vectors \vec{a}_1 and \vec{a}_2 is defined as $0 \leq \angle(\vec{a}_1, \vec{a}_2) \leq \pi$ with

$$\cos \angle(\vec{a}_1, \vec{a}_2) = \frac{\vec{a}_1 \cdot \vec{a}_2}{|\vec{a}_1| \cdot |\vec{a}_2|}.$$



(a) $241 \times 159 \text{ px}^2$ (courtesy of Telea [24], Fig. 8.i) (b) Telea's method, $\epsilon = 5 \text{ px}$; CPU time 0.08 s

Fig. 3 A vandalized digital image (a) and its inpainting by Telea's method (b). (Compare Fig. 1)

direction of steepest descent from x , that is, comes from a shorter distance to the given data.

Examples We show the limitations of Telea's algorithm with two typical examples. Fig. 3(a) shows a vandalized digital image with an inpainting domain of approximately 10 pixels thickness, Fig. 3(b) the inpainted result⁴ using $\epsilon = 5 \text{ px}$. Considerable blur is produced, clearly visible when edges are entering the inpainting domain. It appears that color values are not only diffused but also transported orthogonal to the direction of the edges.

To enhance this effect, we apply Telea's algorithm to the problem of continuing a straight edge through a large inpainting domain, shown in Fig. 4. Besides some blur a rather peculiar transport pattern can be observed. This pattern will be explained by the theory that we will develop in the next section.

3 High-resolution Vanishing Viscosity Limit of Telea's Algorithm

The sequential generic single-pass inpainting algorithm (2) can be thought of as a forward substitution for the equivalent

⁴Telea's algorithm is applied independently to each color channel of the RGB color space, see p. 273.



(a) $300 \times 180 \text{ px}^2$; inpainting domain: $228 \times 108 \text{ px}^2$ (b) Telea's method, $\epsilon = 6 \text{ px}$; CPU time 0.05 s

Fig. 4 A synthetic digital image (a) and its inpainting by Telea's method (b)

system of linear equations

$$\sum_{y \in B_{\epsilon,h}^<(x)} (u_h(x) - u_h(y))w_h(x, y) = 0, \quad x \in D_h;$$

$$u_h|_{\Omega_h \setminus D_h} = u_h^0.$$

This is, because of the definition of the neighborhoods $B_{\epsilon,h}^<(x)$ the system is triangular with respect to the chosen order of pixels. Now, at least formally, we consider the high-resolution limit $h \rightarrow 0$ by looking at this system of equations as a discretization of a continuous integral equation. With a scale factor that will turn out to be convenient later on we obtain

$$\frac{1}{\pi \epsilon^2} \int_{B_\epsilon^<(x)} (u(x) - u(y))w(x, y) dy = 0, \quad x \in D; \tag{6}$$

$$u|_{\Omega \setminus D} = u^0.$$

Because of the order by distance (3) used to define $B_{\epsilon,h}^<(x)$, these sets are discretizations of the truncated balls (see Fig. 2(b) for an illustration)

$$B_\epsilon^<(x) = \{y \in B_\epsilon(x) : T(y) < T(x)\}.$$

Here, we have continued T as the signed distance map into the data domain by defining

$$T(x) = -\text{dist}(x, \partial D), \quad x \in \Omega \setminus D.$$

The integral equation combines directional effects due to both, the truncation of the balls and the anisotropic choice of the weights, with the diffusion caused by the linear averaging. The amount of viscosity is determined by the radius ϵ . To distill the directional effects we study the vanishing viscosity limit $\epsilon \rightarrow 0$.

A prominent role in the formulation of this limit will be played by the (Euclidean) skeleton $\Sigma \subset D$ of the inpainting domain. There are at least four different equivalent definitions of the skeleton, see ([15], §6.4). We choose to define Σ as the set of singularities (locations of the ridges) of the distance map $T(x) = \text{dist}(x, \partial D)$ to the boundary. That is, Σ is the smallest closed set such that $T \in C^1(D \setminus \Sigma)$. In fact, we will assume throughout the paper even more regularity of ∂D , namely $T \in C^2(D \setminus \Sigma)$.

Theorem 1 We consider a function $u \in C^1(D \setminus \Sigma)$.

- (a) For weights of the form $w(x, y) = |x - y|^{-1}k(x, \epsilon^{-1} \cdot (x - y))$ with k uniformly bounded and smooth we have, as $\epsilon \rightarrow 0$,

$$\begin{aligned} & \frac{1}{\pi \epsilon^2} \int_{B_\epsilon^<(x)} (u(x) - u(y))w(x, y) dy \\ &= \vec{c}_0(x) \cdot \nabla u(x) + O(\epsilon), \quad x \in D \setminus \Sigma. \end{aligned}$$

We express \vec{c}_0 using polar coordinates with respect to an orthonormal basis⁵ \vec{a} and \vec{a}^\perp of \mathbb{R}^2 . Let $\vec{e}(\phi) = \cos(\phi)\vec{a} + \sin(\phi)\vec{a}^\perp$ and define the polar angle θ of \vec{n} by

$$\vec{n}(x) = \nabla T(x) = \vec{e}(\theta(x)), \quad x \in D \setminus \Sigma.$$

Then

$$\vec{c}_0(x) = \frac{1}{\pi} \int_{\theta(x)-\pi/2}^{\theta(x)+\pi/2} k_*(x, \vec{e}(\phi))\vec{e}(\phi) d\phi, \tag{7}$$

$$k_*(x, \vec{\eta}) = \int_0^1 k(x, r\vec{\eta})r dr.$$

- (b) Telea's weight function (5), that is, $k(x, \vec{\eta}) = |\vec{n}(x) \cdot \vec{\eta}|/|\vec{\eta}|$, $\vec{n}(x) = \nabla T(x)$, yields

$$\vec{c}_0(x) = \vec{n}(x)/4.$$

Proof Let us consider a fixed $x \in D \setminus \Sigma$ and define the semi-disk

$$S_{\epsilon, \vec{\eta}}(x) = \{y \in B_\epsilon(x) : \vec{\eta} \cdot (x - y) \geq 0\}.$$

By construction the inner boundaries of $S_{\epsilon, \vec{n}(x)}(x)$ and $B_{\epsilon,h}^<(x)$ touch each other tangentially in x while $B_{\epsilon,h}^<(x)$ becomes $S_{\epsilon, \vec{n}(x)}(x)$ asymptotically as $\epsilon \rightarrow 0$. Therefore, the area of the symmetric difference of the two sets is of the order $O(\epsilon^3)$. Since u is assumed to be continuously differentiable in x we obtain that

$$\frac{u(x) - u(y)}{|x - y|} = \nabla u(x) \cdot \frac{x - y}{|x - y|} + O(\epsilon), \quad y \in B_\epsilon(x),$$

which implies, in particular, the boundedness of the expression. Using these two estimates we obtain

$$\begin{aligned} & \frac{1}{\pi \epsilon^2} \int_{B_\epsilon^<(x)} (u(x) - u(y))w(x, y) dy \\ &= \frac{1}{\pi \epsilon^2} \int_{B_\epsilon^<(x)} \frac{u(x) - u(y)}{|x - y|} k(x, \epsilon^{-1}(x - y)) dy \end{aligned}$$

⁵This basis is not necessarily assumed to be positively oriented. Thus, in this paper, \vec{a}^\perp denotes a fixed but arbitrary choice of the two vectors that are normal to $\vec{a} \neq 0$ with $|\vec{a}^\perp| = |\vec{a}|$.

$$\begin{aligned}
 &= \frac{1}{\pi \epsilon^2} \int_{S_{\epsilon, \vec{n}(x)}(x)} \frac{u(x) - u(y)}{|x - y|} k(x, \epsilon^{-1}(x - y)) dy \\
 &\quad + O(\epsilon) \\
 &= \frac{1}{\pi \epsilon^2} \int_{S_{\epsilon, \vec{n}(x)}(x)} k(x, \epsilon^{-1}(x - y)) \nabla u(x) \cdot \frac{x - y}{|x - y|} dy \\
 &\quad + O(\epsilon) \\
 &= \frac{1}{\pi \epsilon^2} \int_{S_{\epsilon, \vec{n}(x)}(0)} k(x, \epsilon^{-1}(-y)) \nabla u(x) \cdot \frac{-y}{|-y|} dy \\
 &\quad + O(\epsilon) \\
 &= \frac{1}{\pi} \int_{S_{1, \vec{n}(x)}(0)} k(x, -y) \nabla u(x) \cdot \frac{-y}{|-y|} dy + O(\epsilon).
 \end{aligned}$$

Introducing polar coordinates, we realize that the vectors $-y$, $y \in S_{1, \vec{n}(x)}(0)$, are characterized by a polar angle between $\theta(x) - \pi/2$ and $\theta(x) + \pi/2$ and a radius between 0 and 1. Hence we get

$$\begin{aligned}
 &\frac{1}{\pi \epsilon^2} \int_{B_\epsilon^c(x)} (u(x) - u(y)) w(x, y) dy \\
 &= \frac{1}{\pi} \int_{\theta(x) - \pi/2}^{\theta(x) + \pi/2} \int_0^1 k(x, r \vec{e}(\phi)) \nabla u(x) \cdot \vec{e}(\phi) r dr d\phi \\
 &\quad + O(\epsilon) \\
 &= \frac{1}{\pi} \int_{\theta(x) - \pi/2}^{\theta(x) + \pi/2} k_*(x, \vec{e}(\phi)) \vec{e}(\phi) \cdot \nabla u(x) d\phi + O(\epsilon) \\
 &= \vec{c}_0(x) \cdot \nabla u(x) + O(\epsilon),
 \end{aligned}$$

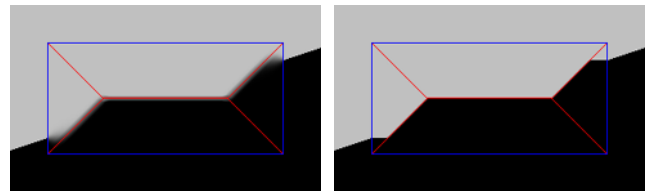
which proves assertion (a).

Specifying Telea’s weight function $k(x, \vec{\eta}) = |\vec{n}(x) \cdot \vec{\eta}|/|\vec{\eta}|$ gives

$$\begin{aligned}
 k_*(x, \vec{e}) &= \int_0^1 k(x, r \vec{e}) r dr = |\vec{n}(x) \cdot \vec{e}| \int_0^1 r dr \\
 &= \frac{1}{2} |\vec{n}(x) \cdot \vec{e}|, \quad |\vec{e}| = 1.
 \end{aligned}$$

Now, for $\theta(x) - \pi/2 \leq \phi \leq \theta(x) + \pi/2$ we have $\vec{n}(x) \cdot \vec{e}(\phi) = \vec{e}(\theta(x)) \cdot \vec{e}(\phi) \geq 0$ and therefore

$$\begin{aligned}
 \vec{c}_0(x) &= \frac{1}{\pi} \int_{\theta(x) - \pi/2}^{\theta(x) + \pi/2} k_*(x, \vec{e}(\phi)) \vec{e}(\phi) d\phi \\
 &= \frac{1}{2\pi} \int_{\theta(x) - \pi/2}^{\theta(x) + \pi/2} \vec{n}(x) \cdot \vec{e}(\phi) \vec{e}(\phi) d\phi \\
 &= \frac{1}{2} \left(\underbrace{\frac{1}{\pi} \int_{\theta(x) - \pi/2}^{\theta(x) + \pi/2} \vec{e}(\phi) \vec{e}(\phi)^T d\phi}_{=A} \right) \vec{n}(x) \\
 &= A \vec{n}(x)/2.
 \end{aligned}$$



(a) Telea’s method, $\epsilon = 6$ px (b) Telea’s method, $\epsilon = 1$ px

Fig. 5 (Color online) Explanation for the effect of Fig. 4. The skeleton Σ is shown in red, the boundary ∂D in blue

By a direct calculation involving trigonometric functions we obtain $A = I/2$. Alternatively, one can argue that by the symmetry $\vec{e}(\phi \pm \pi) = -\vec{e}(\phi)$

$$2A = \frac{1}{\pi} \int_{\theta - \pi}^{\theta + \pi} \vec{e}(\phi) \vec{e}(\phi)^T d\phi = \frac{1}{\pi} \int_{-\pi}^{\pi} \vec{e}(\phi) \vec{e}(\phi)^T d\phi.$$

This shows that A is invariant under orthogonal coordinate transformations and therefore $A = \lambda I$. Now

$$\begin{aligned}
 2\lambda = \text{tr}(A) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \text{tr}(\vec{e}(\phi) \vec{e}(\phi)^T) d\phi \\
 &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \vec{e}(\phi) \cdot \vec{e}(\phi) d\phi = 1.
 \end{aligned}$$

Summarizing, we get $\vec{c}_0(x) = \vec{n}(x)/4$, which finishes the proof of (b). \square

Theorem 1 shows that the vanishing viscosity limit of the equations (6), which are underlying Telea’s algorithm, is given by the following problem:

$$\vec{n}(x) \cdot \nabla u(x) = 0 \quad \text{on } D \setminus \Sigma, \quad u|_{\partial D} = u^0|_{\partial D}. \quad (8)$$

Even though it is a hyperbolic Dirichlet problem, this boundary value problem is well-posed (see Theorem 3) because we have excluded the skeleton Σ from the domain of validity of the equation. The reason is that the skeleton Σ is precisely the set where the characteristics of the vector field $\vec{n}(x) = \nabla T(x)$ intersect for the first time and shocks are formed, see ([15], §6.4). Thus, the limit equations give the following continuous inpainting algorithm (see also [3], Fig. 2):

The known image value $u^0(x)$ of a boundary point $x \in \partial D$ is transported along the straight line of the normal pointing inwards into the inpainting domain D until this normal meets the skeleton Σ , that is, until it intersects with a different normal transporting different image values. There is no transport of information across Σ .

This inpainting procedure is clearly visible in Fig. 5; the result for $\epsilon = 1$ px is exactly the same as an application of the limit equation (8) would yield. Thus, our analysis has led

to a simple explanation of the peculiar behavior of Telea’s algorithm shown in Fig. 4.

4 Towards the Realization of Arbitrary Transport Directions

The vanishing viscosity limit (8) clearly reveals two major shortcomings of Telea’s algorithm:

- Inpainting information is only transported in a direction given by the geometry of the inpainting domain D .
- There is no transport of inpainting information across the skeleton Σ of D .

Moreover, there is the problem of a considerable amount of blur for larger values of the inpainting radius ϵ . We aim to resolve these issues, at least for the most parts, by a suitable modification of Telea’s weight function.

The inpainting problem is basically about appropriately closing isophotes which, being level lines of u , are not intersecting. This property is shared by the trajectories (characteristics) of the flow of a unit vector field \vec{c} on D , which we therefore take as a model for the unknown isophotes. The actual construction of vector fields \vec{c} that are useful for the inpainting problem is the subject of the next section. Here, we assume that it is given. However, we note that isophotes do not have an orientation. Thus, only the slope but not the sign of the vector field \vec{c} is of importance. In analogy with (8) we therefore aim for a vanishing viscosity limit of the form

$$\pm \vec{c}(x) \cdot \nabla u(x) = 0 \quad \text{on } D \setminus \Sigma, \quad u|_{\partial D} = u^0|_{\partial D}. \tag{9}$$

The signs should be chosen, depending on x , in a way to make the hyperbolic Dirichlet problem well-posed; the formation of shocks has to be confined to the skeleton Σ .

Summarizing, we aim at choosing the weight w of the generic single-pass algorithm with distance ordering in such a way that the normalized limit vector field $\vec{c}_* = \vec{c}_0/|\vec{c}_0|$ of Theorem 1 matches the given $\pm \vec{c}$. In view of Theorem 1 case (b) it is tempting to replace Telea’s weight function (5) by the expression

$$w(x, y) = \frac{|\vec{c}(x) \cdot (x - y)|}{|x - y|^2}. \tag{10}$$

However, the resulting match is not good enough (see Fig. 6). The directional dependence of w has to be much more pronounced; the result for a carefully crafted exponential dependence on the direction is given in the next theorem.

Theorem 2 *Let $\vec{c}(x)$ be a normalized smooth vector field on $D \setminus \Sigma$. Consider the weight function*

$$w(x, y) = \sqrt{\frac{\pi}{2}} \frac{\mu}{|x - y|}$$

$$\cdot \exp\left(-\frac{\mu^2}{2\epsilon^2} |\vec{c}^\perp(x) \cdot (x - y)|^2\right). \tag{11}$$

and its induced vector field \vec{c}_0 as defined in Theorem 1. Then, \vec{c}_0 is nonvanishing and yields a normalized vector field $\vec{c}_* = \vec{c}_0/|\vec{c}_0|$ on $D \setminus \Sigma$ having the same smoothness as \vec{n} . There is a positive constant ρ_μ such that $\vec{n}(x) \cdot \vec{c}_*(x) \geq \rho_\mu > 0$, $x \in D \setminus \Sigma$. With

$$\theta(x) = \angle(\vec{n}(x), \vec{c}(x)) \in [0, \pi],$$

the deviation angle between \vec{c}_* and \vec{c} is of the form $\angle(\vec{c}_*(x), \vec{c}(x)) = \Delta(\theta(x))$. The continuous function $\Delta(\theta)$ satisfies the symmetry law

$$\Delta(\pi - \theta) = \pi - \Delta(\theta), \quad 0 \leq \theta \leq \pi, \tag{12}$$

and the asymptotic expansion (as $\mu \rightarrow \infty$)

$$\Delta(\theta) = \sqrt{\frac{2}{\pi}} \log\left(\tan\left(\frac{\theta}{2} + \frac{\pi}{4}\right)\right) \mu^{-1} + O(\mu^{-2}), \quad 0 \leq \theta < \frac{\pi}{2}. \tag{13}$$

Here, the estimate $O(\mu^{-2})$ is uniform for θ uniformly bounded away from $\pi/2$.

Proof We consider a fixed but arbitrary $x \in D \setminus \Sigma$; the dependence on it will be suppressed in the notation if convenient. We define $\vec{a} = \pm \vec{c}$ and a^\perp by choosing the sign and orientation such that $\vec{n} \cdot \vec{a} \geq 0$ and $\vec{n} \cdot \vec{a}^\perp \geq 0$. Then, using polar coordinates with respect to the orthonormal basis \vec{a}, \vec{a}^\perp we obtain (with the notation of Theorem 1), writing $\theta_* = \angle(\vec{n}, \vec{c})$ in this proof,

$$\vec{n} = \vec{e}(\theta), \quad 0 \leq \theta = \min(\theta_*, \pi - \theta_*) \leq \pi/2.$$

The case $\theta = \pi - \theta_*$ corresponds to the choice $\vec{a} = -\vec{c}$. The weight function (11) can be written in the form discussed in Theorem 1 with

$$k(x, \vec{\eta}) = \mu \sqrt{\frac{\pi}{2}} \exp\left(-\frac{\mu^2}{2} |\vec{c}^\perp \cdot \vec{\eta}|^2\right).$$

Using $|\vec{c}^\perp \cdot \vec{e}(\phi)|^2 = \sin^2 \phi$ we get by (7)

$$\vec{c}_0 = \frac{1}{\sqrt{2\pi}} \int_{\theta-\pi/2}^{\theta+\pi/2} \frac{1 - \exp(-\frac{\mu^2}{2} \sin^2 \phi)}{\mu \sin^2 \phi} \vec{e}(\phi) d\phi.$$

Writing $c^0 = \vec{c}_0 \cdot \vec{a}$ and $c^1 = \vec{c}_0 \cdot \vec{a}^\perp$ for the two components of \vec{c}_0 , we get

$$\begin{pmatrix} c^0 \\ c^1 \end{pmatrix} = \frac{1}{\sqrt{2\pi}} \int_{\theta-\pi/2}^{\theta+\pi/2} \frac{1 - \exp(-\frac{\mu^2}{2} \sin^2 \phi)}{\mu \sin^2 \phi} \begin{pmatrix} \cos \phi \\ \sin \phi \end{pmatrix} d\phi$$

Using the symmetries of the trigonometric functions at $\phi = 0$ and $\phi = \pi/2$ we shorten the intervals of integration to

$$c^0 = \sqrt{\frac{2}{\pi}} \int_0^{\pi/2-\theta} \frac{1 - \exp(-\frac{\mu^2}{2} \sin^2 \phi)}{\mu \sin^2 \phi} \cos \phi \, d\phi,$$

$$c^1 = \sqrt{\frac{2}{\pi}} \int_0^\theta \frac{1 - \exp(-\frac{\mu^2}{2} \cos^2 \phi)}{\mu \cos^2 \phi} \cos \phi \, d\phi.$$

In particular, these two integrals imply that $c^0 > 0$, $0 \leq \theta < \pi/2$, and $c^1 > 0$, $0 < \theta \leq \pi/2$, showing that \vec{c}_0 is nonvanishing for all θ . Hence, \vec{c}_* is well-defined. Since $\vec{n} \cdot \vec{e}(\phi) = \vec{e}(\theta) \cdot \vec{e}(\phi) > 0$, $\theta - \pi/2 < \phi < \theta + \pi/2$, we get

$$\vec{n} \cdot \vec{c}_0 = \frac{1}{\sqrt{2\pi}} \int_{\theta-\pi/2}^{\theta+\pi/2} \frac{1 - \exp(-\frac{\mu^2}{2} \sin^2 \phi)}{\mu \sin^2 \phi} \cdot \vec{e}(\theta) \cdot \vec{e}(\phi) \, d\phi > 0,$$

and therefore $\vec{n} \cdot \vec{c}_* > 0$ for all θ . Continuity yields the asserted existence of the constant $\rho_\mu > 0$.

The signs and forms of c^0 and c^1 imply further that $\vec{c}_* = \vec{e}(\Delta(\theta))$ with a continuous function $\Delta(\theta)$ of $\theta \in [0, \pi/2]$ that satisfies $\Delta(\pi/2) = \pi/2$. This yields the asserted expression $\angle(\vec{c}_*, \vec{c}) = \Delta(\theta_*)$ by continuously extending $\Delta(\theta)$ according to the symmetry law (12) if $\theta_* = \pi - \theta$ and thus $\vec{c} = -\vec{a}$.

Finally, we prove the asymptotic expansion (13). To this end we assume that $0 \leq \theta < \pi/2$. Substituting $\xi = \mu \sin \phi$ in the integral expression for c^0 gives

$$c^0 = \sqrt{\frac{2}{\pi}} \int_0^{\mu \cos \theta} \frac{1 - \exp(-\xi^2/2)}{\xi^2} \, d\xi$$

$$= \sqrt{\frac{2}{\pi}} \int_0^\infty \frac{1 - \exp(-\xi^2/2)}{\xi^2} \, d\xi$$

$$- \sqrt{\frac{2}{\pi}} \int_{\mu \cos \theta}^\infty \frac{1 - \exp(-\xi^2/2)}{\xi^2} \, d\xi$$

$$= 1 - \sqrt{\frac{2}{\pi}} \left(\int_{\mu \cos \theta}^\infty \frac{1}{\xi^2} \, d\xi \right)$$

$$\cdot \left(1 - \sigma_0 \exp\left(-\frac{\mu^2}{2} \cos^2 \theta\right) \right)$$

$$= 1 - \sqrt{\frac{2}{\pi}} \sec(\theta) \mu^{-1}$$

$$\cdot \left(1 - \sigma_0 \exp\left(-\frac{\mu^2}{2} \cos^2 \theta\right) \right),$$

with $0 < \sigma_0 < 1$ obtained by the mean value theorem. Likewise we get

$$c^1 = \sqrt{\frac{2}{\pi}} \int_0^\theta \frac{1 - \exp\left(-\frac{\mu^2}{2} \cos^2 \phi\right)}{\mu \cos \phi} \, d\phi$$

$$= \sqrt{\frac{2}{\pi}} \left(\int_0^\theta \sec \phi \, d\phi \right) \mu^{-1}$$

$$\cdot \left(1 - \sigma_1 \exp\left(-\frac{\mu^2}{2} \cos^2 \theta\right) \right)$$

$$= \sqrt{\frac{2}{\pi}} \log\left(\tan\left(\frac{\theta}{2} + \frac{\pi}{4}\right)\right) \mu^{-1}$$

$$\cdot \left(1 - \sigma_1 \exp\left(-\frac{\mu^2}{2} \cos^2 \theta\right) \right),$$

where $0 < \sigma_1 < 1$. Since the estimate

$$1 - \sigma_j \exp\left(-\frac{\mu^2}{2} \cos^2 \theta\right) = 1 + O(\mu^{-1}), \quad j = 0, 1,$$

is uniform for θ uniformly bounded away from $\pi/2$, we get

$$\Delta(\theta) = \arctan\left(\frac{c^1}{c^0}\right) = \frac{c^1}{c^0} \left(1 + O\left(\frac{c^1}{c^0}\right) \right)$$

$$= \sqrt{\frac{2}{\pi}} \log\left(\tan\left(\frac{\theta}{2} + \frac{\pi}{4}\right)\right) \mu^{-1} + O(\mu^{-2}),$$

that is, the asserted asymptotic expansion.⁶ □

Figure 6 shows the deviation angle $\Delta = \angle(\vec{c}_*, \vec{c})$ as a function of $\theta = \angle(\vec{n}, \vec{c})$ for Telea’s original weight function (5), the simple modification (10), and the exponentially confined one (11) of Theorem 2. There is a good match of \vec{c}_* with \vec{c} if $\Delta \approx 0$, and with $-\vec{c}$ if $\Delta \approx \pi$. Thus, we observe that the simple modification (10) does not even come close to this match, but is quantitatively just a minor modification of Telea’s weight function. In fact, it behaves quite similar to Telea’s algorithm in practice.

On the other hand, the exponentially confined weight (11) results in excellent matches for larger values of the parame-

⁶The reason of the nonuniformity of the expansions near $\theta = \pi/2$ is not unboundedness, but a change of the type of asymptotic expansion in μ . In fact, we can prove that (γ denoting Euler’s constant)

$$c^0|_{\theta=\pi/2} = 0,$$

$$c^1|_{\theta=\pi/2} = \sqrt{\frac{2}{\pi}} \log(\mu) \mu^{-1} + \frac{\gamma + \log 2}{\sqrt{2\pi}} \mu^{-1} + O(\mu^{-2}).$$

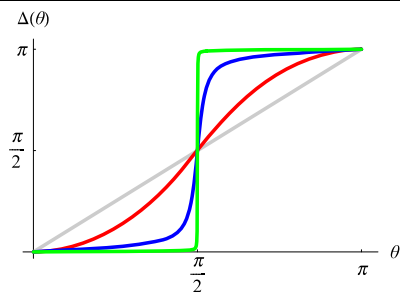


Fig. 6 (Color online) Deviation angle $\Delta(\theta)$ for different weights; gray: Telea’s weight (5); red: the modified weight (10); blue: the exponentially confined weight (11), $\mu = 10$; green: the exponentially modified weight (11), $\mu = 100$

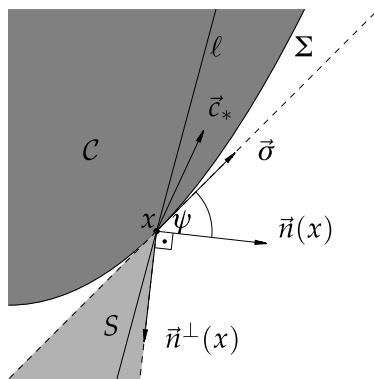


Fig. 7 Location of the straight line ℓ through $\vec{c}_*(x)$, $x \in \Sigma_{\text{trans}}$ (being an inflow point for the component C)

ter μ if θ is bounded away from $\pi/2$. In fact, the asymptotic expansion (13) proves that

$$\lim_{\mu \rightarrow \infty} \vec{c}_*(x) = \begin{cases} \vec{c}(x), & \vec{n}(x) \cdot \vec{c}(x) > 0, \\ -\vec{c}(x), & \vec{n}(x) \cdot \vec{c}(x) < 0, \\ \vec{n}(x), & \vec{n}(x) \perp \vec{c}(x). \end{cases} \quad (14)$$

The change of the sign is needed for the well-posedness which the limit equation inherits from the single-pass algorithm (see Theorem 3); it keeps the algorithm’s directional averaging to be inherently *upwinding*. The price to pay for it is, by continuity, the *unavoidable* sudden rotation of \vec{c}_* into the perpendicular direction \vec{n} if the flow generated by \vec{c} becomes close to tangential to the level lines of the distance map (that is, $\vec{c} \perp \vec{n}$). We therefore predict, but for such cases only, some visual effects reminiscent of Telea’s algorithm.

Moreover, all the weights shown in Fig. 6 satisfy $0 < \Delta(\theta) \leq \theta$ for $0 < \theta \leq \pi/2$. This can be proved by (13) for large values of μ ; for all other cases we rely on the shown numerical evidence. This property is needed in the following theorem that proves the well-posedness of the limit equation.

Theorem 3 Let $D \subset \mathbb{R}^2$ be a bounded domain whose boundary ∂D and skeleton Σ are made of finitely many C^1 -

arcs. We assume that for each of the finitely many components C of $D \setminus \Sigma$ the gradient \vec{n} of the distance map T can be continued to the boundary as $\vec{n} \in C^1(\bar{C}, \mathbb{R}^2)$. Let \vec{c} be a normalized C^1 -vector field on \bar{D} , and \vec{c}_* a normalized vector field on $D \setminus \Sigma$ with the same smoothness properties as \vec{n} . We assume that \vec{c}_* is uniformly pointing inwards, that is, there is a positive constant ρ such that

$$\vec{n}(x) \cdot \vec{c}_*(x) \geq \rho > 0, \quad x \in D \setminus \Sigma. \quad (15)$$

We assume further that the deviation angle between \vec{c} and \vec{c}_* is of the form $\angle(\vec{c}_*(x), \vec{c}(x)) = \Delta(\theta(x))$, $\theta(x) = \angle(\vec{n}(x), \vec{c}(x))$, with a continuous function $\Delta(\theta)$ satisfying (see Fig. 6)

$$\Delta(\pi - \theta) = \pi - \Delta(\theta) \quad \text{and} \quad 0 < \Delta(\theta) \leq \theta, \quad 0 < \theta \leq \pi/2. \quad (16)$$

Then, there is a geometrically defined subset $\Sigma_{\text{trans}} \subset \Sigma$ of transparent points of the skeleton (see Fig. 7) and a well-defined linear solution operator $C(\partial D) \rightarrow C((D \setminus \Sigma) \cup \partial D \cup \Sigma_{\text{trans}})$, $u^0 \mapsto u$, of the Dirichlet problem

$$\vec{c}_*(x) \cdot \nabla u(x) = 0 \quad \text{on } D \setminus \Sigma, \quad u|_{\partial D} = u^0|_{\partial D}, \quad (17)$$

which is l_∞ -stable and satisfies the comparison principle (see p. 262).

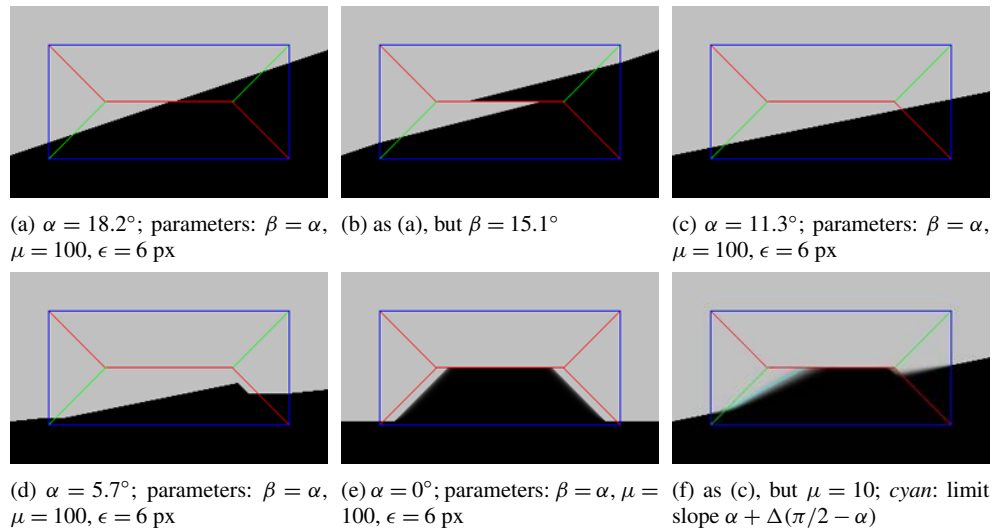
Proof The assumed continuity properties imply that the backward flow of \vec{c}_* connects the points of a component C of $D \setminus \Sigma$ either with the boundary ∂D or the skeleton Σ . We call the endpoints of the backward flow *inflow points* of C . On the other hand, condition (15) shows that the distance to ∂D is strictly increasing along the forward flow of \vec{c}_* . That is, all the forward characteristics must connect to the skeleton. We call the endpoints of the forward flow *outflow points* of C . Below we will show that a point x in the relative interior of an C^1 -arc of Σ , that is, a point belonging to the boundary of exactly two components of $D \setminus \Sigma$, cannot be an inflow point for both of the components. Thus, a backward characteristics ending in x in one of the components can be continued by a backward characteristics starting at x in the neighboring component. We call such a point of the skeleton *transparent* with respect to \vec{c}_* . By a compactness argument, and since by (15) the distance to ∂D is strictly decreasing along backward characteristics, after finitely many such continuations through transparent points we must finally reach the boundary ∂D .

This way we have connected each⁷ point $x \in D \setminus \Sigma$ by a unique path of backward characteristics with a boundary

⁷Well, not quite. There may be exceptional points x that are connected by a backwards characteristic to a triple point of Σ , that is, a point at which several C^1 -arcs of Σ join. However, these exceptional points are nowhere dense and u can be defined for them by a continuity argument.

Fig. 8 (Color online)

Inpainting a straight edge (see also Figs. 4 and 5). $\Sigma \setminus \Sigma_{\text{trans}}$ is shown in red, Σ_{trans} in green, ∂D in blue



point $x_0 \in \partial D$. We set $u(x) = u^0(x_0)$. Since the path is independent of u^0 this defines a linear operator $u^0 \mapsto u$ that is obviously l_∞ -stable and satisfies the comparison principle. Since by construction u is constant along characteristics it satisfies the equation $\vec{c}_* \cdot \nabla u = 0$ within each component. If we denote the set of transparent points by Σ_{trans} , it is easy to see that u can be continued as a continuous function to $\partial D \cup \Sigma_{\text{trans}}$. Thus, points of discontinuity (shocks) belong to the set $\Sigma \setminus \Sigma_{\text{trans}}$.

We finish by proving that a point x in the relative interior of an C^1 -arc of Σ that is an inflow point for one of its neighboring components \mathcal{C} must be transparent. In the following, we simplify the argument by assuming, without loss of generality, that $x = 0$ and suppress it in the notation if convenient.

We begin by studying the geometry met by the inflow vector \vec{c}_* (that is, the forward characteristic of \vec{c}_* enters \mathcal{C} at x). Let the unit vector $\vec{\sigma}$ be tangential to Σ at x , with the sign chosen so that $0 < \psi = \angle(\vec{n}, \vec{\sigma}) \leq \pi/2$ (see Fig. 7). We choose \vec{n}^\perp so that $\angle(\vec{n}^\perp, \vec{\sigma}) = \psi + \pi/2$. Now, since the inflow vector must satisfy $\vec{c}_* \neq \vec{n}$, conditions (15) and (16) imply that \vec{c}_* points to the interior of the acute sector between \vec{n} and the straight line ℓ spanned by \vec{c} . The opening angle of this sector must be larger than ψ since otherwise \vec{c}_* would not point to an inflow direction. Also, $\vec{n} \perp \ell$ is impossible since this would, by (16), imply $\vec{c}_* = \vec{n}$. We conclude that ℓ passes through the interior of the angular sector \mathcal{S} between the tangent and \vec{n}^\perp (see Fig. 7). The opening angle of \mathcal{S} is $\pi/2 - \psi < \pi/2$.

Now, suppose that x is an inflow point of the neighboring component \mathcal{C}' , too. Since the tangent of Σ at x is the bisectrix of the two different continuations \vec{n} and \vec{n}' ,⁸ the

⁸This follows from elementary geometry if the two points on ∂D that are closest to x (and therefore at equal distance from x by the defini-

geometric constraints discussed for ℓ with respect to \mathcal{C} apply with respect to \mathcal{C}' after being reflected at the tangent. That is, ℓ must pass through the interior of the angular sectors \mathcal{S} and its reflection \mathcal{S}' . This is impossible, because the opening angle of \mathcal{S} is smaller than $\pi/2$. \square

An Illustrative Example We illustrate the limit theory developed in this section by applying the generic single-pass algorithm with distance ordering, using the exponential confining weight function (11), to the synthetic inpainting problem of Fig. 4(a). Here, we generalize this inpainting problem by considering varying slopes $\alpha = \angle(\text{edge, horizontal})$ of the broken edge between the black and the gray region. The inpainting domain has on purpose been chosen rather large to make all effects of the algorithm clearly visible.

For the exponentially confining weight function we choose a constant vector field \vec{c} with a given slope $\beta = \angle(\vec{c}, \text{horizontal})$. The proof of Theorem 3 (see Fig. 7) shows that the parts of Σ which are shown in green in Fig. 8 are transparent for prescribed slopes $0 < \beta < \pi/4$.

As expected by the limit theory, Fig. 8(a) shows that for $\beta = \alpha = 18.2^\circ$ the broken edge of the original inpainting problem is perfectly closed ($\mu = 100, \epsilon = 6$ px). On the other hand, if we prescribe the smaller slope $\beta = 15.1^\circ$, the broken edge cannot be closed.⁹ Fig. 8(b) shows that, in accordance with the theory, a shock forms at the nontransparent (red) part of the skeleton.

As the angle α gets smaller, the vector field \vec{c} becomes nearly parallel to ∂D along the broken edge. The theory (see

tion of the skeleton) belong to parts of ∂D that are line segments. The general case follows from that by linearization.

⁹The closing of edges at the skeleton is comparable to the digging of a tunnel from two ends: if the measurements, the plan, and the performance were good the digging teams will meet somewhere in the middle (at the skeleton). If not, they will fail badly.

Fig. 6) tells us that \vec{c}_* rotates into the direction of \vec{n} in the corresponding component of $D \setminus \Sigma$. That is, we expect that the actual slope of the inpainted edge becomes increasingly larger than α , forming a shock at the nontransparent part of the skeleton. In fact, there is a transition from a perfect continuation ($\alpha = 11.3^\circ$) in Fig. 8(c), passing a clearly visible shock ($\alpha = 5.7^\circ$) in Fig. 8(d), to a result corresponding to Telea’s algorithm ($\alpha = 0^\circ$) in Fig. 8(e). We observe that the shocks are, in parts, not exactly located at the skeleton but have an offset of $\epsilon = 6$ px. Even though the inpainting radius $\epsilon = 6$ px is far from being small, there is next to no diffusion visible in Fig. 8(a–e). (Compare with the result of Telea’s weight in Fig. 4.) This is due to the sharp directional confinement caused by the large value $\mu = 100$.

Finally, in Fig 8(f) we study the influence of the parameter μ by decreasing its value from the perfect $\mu = 100$ of Fig 8(c) to $\mu = 10$. The limit theory predicts, in the critical component of $D \setminus \Sigma$, a pronounced rotation of the prescribed slope $\alpha = 11.3^\circ$ by the amount of

$$\angle(\vec{c}_*, \vec{c}) = \Delta\left(\frac{\pi}{2} - \alpha\right)\Big|_{\mu=10} = 16.8^\circ,$$

which is consistent with the actual behavior shown in Fig 8(f). Further, there is more visible diffusion here. However, different from the limit theory, in the realm of the discrete setting of the actual algorithm there is no further improvement of Fig. 8(d) if we increase the value of μ . The reason is that there are only finitely many different slopes $x - y, y \in B_{\epsilon,h}^<(x)$, that enter the exponent of the expression

$$w(x, y) = \sqrt{\frac{\pi}{2}} \frac{\mu}{|x - y|} \exp\left(-\frac{\mu^2}{2\epsilon^2} |\vec{c}^\perp(x) \cdot (x - y)|^2\right).$$

Moreover, generally none of them will be exactly parallel to the preassigned direction \vec{c} . This means that values of μ which are too large simply result, by underflow, in a weight w that is identical to a floating point zero.

5 Coherence Direction and Modified Structure Tensor

Generally, the vector field \vec{c} that enters our weight function (11) has to be extracted from the (already known) image values u . A reasonable choice of \vec{c} has to be aligned with the level lines (isophotes) of u or, at least, with those that carry pronounced information, e.g., the edges.

Edge Detecting Flow As we have already discussed in the introduction, the celebrated inpainting method of Bertalmio et al. [3] (see also [19], §8.2.2, or [8], §6.11.2) is based on a third order pde that can be written in the equivalent forms

$$u_t = \nabla^\perp u \cdot \nabla \Delta u = -\nabla^\perp \Delta u \cdot \nabla u.$$

The time evolution of this equation has to be stabilized in practice; the authors suggest a hybrid method by interleaving every 15 time steps of size $\tau = 0.1$ two time steps of an anisotropic diffusion equation; see [3], p. 420 or [19], p. 350. Leaving the anisotropy aside, this hybrid method is basically a splitting scheme for the integro-differential equation

$$u_t = -\nabla^\perp \Delta u_\sigma \cdot \nabla u, \quad u_\sigma = K_\sigma \star u,$$

$$K_\sigma(x) = \frac{1}{2\pi\sigma^2} e^{-\frac{|x|^2}{2\sigma^2}},$$

Here, $K_\sigma \star u$ denotes the spatial convolution with the Gaussian (heat kernel) K_σ of standard deviation σ . Formally, the stationary state of this equation is given by

$$\pm \nabla^\perp \Delta u_\sigma \cdot \nabla u = 0.$$

In view of the “ideal” limit transport equation (9) of the previous section, this suggests the choice

$$\vec{c} = \nabla^\perp \Delta u_\sigma. \tag{18}$$

Besides the formal relation to Bertalmio et al.’s [3] method, the following rationale can be given for this choice. The zero crossings of Δu_σ are the classical characterization of edges as given by Marr and Hildreth; see, e.g., [27], p. 5, or [8], §7.2.1.¹⁰ Now, if x is a point of a Marr–Hildreth edge, the vector $\vec{c}(x) = \nabla^\perp \Delta u_\sigma(x)$ defines the *tangent* direction of that edge. Thus, for x in between the edges, the still well-defined expression $\vec{c}(x)$ yields a reasonable interpolation. We therefore call the flow generated by (18) the *edge detection flow*.

We illustrate the behavior of this choice by applying it to a digitized fingerprint. Figure 9(e) shows a visible alignment of the edge detection flow with the ridges of the post-processed fingerprint in Fig. 9(d). However, for the original digitization shown in Fig. 9(a) the edge detection flow in Fig. 9(b) closely follows minor local features; it loses relation with the global coherent flow of information. Thus, the example reveals that the edge detection flow, like edge detection itself, has problems with its robustness. Since a better choice can be made we do not advocate its use in combination with the weight function (11) for image inpainting.

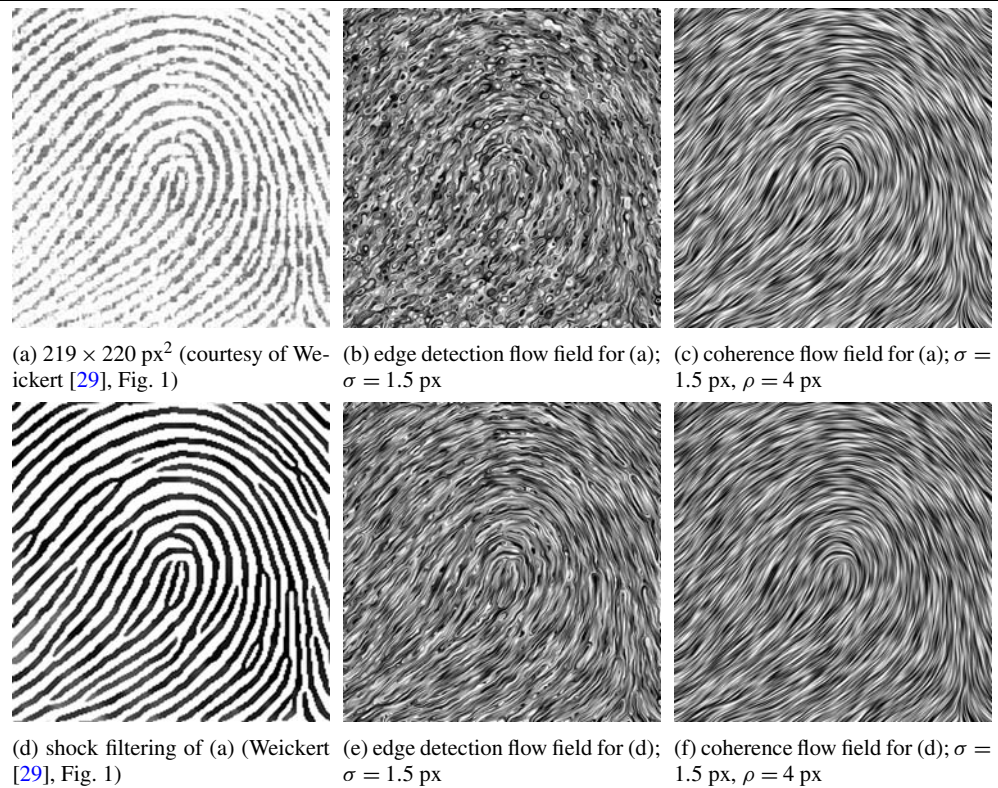
Structure Tensor and Coherence Flow We follow Weickert’s [27, 29] approach of using the structure tensor to the robust determination of coherence directions in images; see ([27], §5.2) for coherence-enhancing anisotropic diffusion

¹⁰The anisotropic diffusion actually used by Bertalmio et al. [3] relates presumably to the choice

$$\vec{c} = \nabla^\perp D^2 u_\sigma (\nabla u_\sigma / |\nabla u_\sigma|, \nabla u_\sigma / |\nabla u_\sigma|),$$

which can be motivated by the Canny–Haralick edge detector; see, e.g., [8], §7.2.1.

Fig. 9 Edge detection flow (b and e) vs. coherence flow (c and f) for a digitized fingerprint (a and d)



and [29] for coherence-enhancing shock-filtering.¹¹ In fact, Weickert’s approach will not only give a coherence direction but also a measure of coherence strength which allows us to adapt the parameter μ of the weight function (11) to local features of the image.

The structure tensor J_ρ of a gray tone image u is the symmetric positive semidefinite 2×2 -matrix defined by

$$J_\rho(\nabla u_\sigma) = K_\rho \star (\nabla u_\sigma \otimes \nabla u_\sigma), \quad u_\sigma = K_\sigma \star u.$$

Its eigenvalues are $0 \leq \lambda_1(x) \leq \lambda_2(x)$ with the corresponding normalized eigenvectors denoted as $\vec{w}_1(x)$ and $\vec{w}_2(x)$. We note that for the rank-one matrix $J_0(\nabla u_0) = \nabla u \otimes \nabla u$, $\nabla u \neq 0$, we have $\lambda_1 = 0$, $\lambda_2 = |\nabla u|^2$, $\vec{w}_1 = \nabla^\perp u / |\nabla u|$, $\vec{w}_2 = \nabla u / |\nabla u|$. Hence, the coherence direction

$$\vec{c} = \vec{w}_1, \quad w_1 \text{ normalized eigenvector}$$

$$\text{to the minimal eigenvalue of } J_\rho(\nabla u_\sigma), \quad (19)$$

is a smoothing of the isophote direction $\nabla^\perp u_\sigma$ on a second-moment scale ρ . The induced flow will be called coherence flow.

The robustness of the coherence flow, and its power to describe larger-scale information flow, is illustrated in

¹¹In the latter reference one can also find pointers to earlier, independent uses of the structure tensor in the literature. There, it is known by the names *second-moment matrix*, *scatter matrix*, *Förstner interest operator*.

Figs. 9(c) and (f). There is barely a difference visible if applied to the digitization (Fig. 9(a)) or to the image post-processed by coherence-enhancing shock-filtering (Fig. 9(d)). In fact, the post-processed image was calculated using the coherence flow of the digitized image (see [29], Fig. 1). We observe that small¹² scale features that are present in the edge-detecting flow, such as small closed loops of ridges, are neglected by the coherence flow.

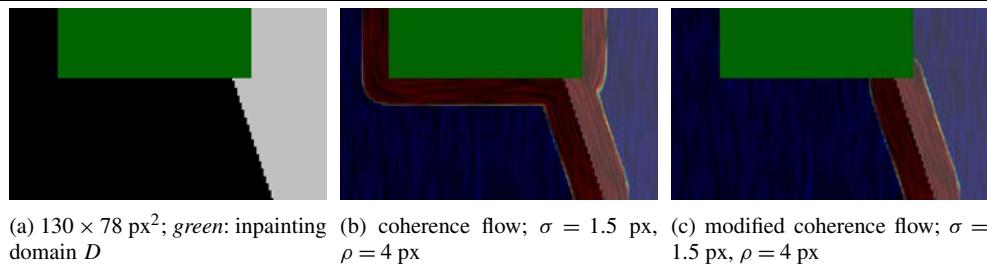
If $\lambda_1 \approx \lambda_2$ the determination of the two eigendirections of the structure tensor becomes ill conditioned. In fact, for $\lambda_1 = \lambda_2$ the structure tensor becomes a multiple of the identity matrix and any system of orthonormal vectors will serve as \vec{w}_1 and \vec{w}_2 . That is to say, w_1 does not contain any viable information then; there is just a weak coherence present. For this reason, we introduce the following measure of local coherence strength

$$\mu(x) = \begin{cases} 1 & \text{if } \lambda_1(x) = \lambda_2(x), \\ 1 + \kappa \exp\left(\frac{-\delta_{\text{quant}}^4}{(\lambda_2(x) - \lambda_1(x))^2}\right) & \text{otherwise.} \end{cases} \quad (20)$$

(A similar coherence measure was used by Weickert ([27], p. 128) for coherence-enhancing anisotropic diffusion.) To make the expression scale invariant, we take δ_{quant} as the resolution of the quantization, that is, the difference of two successive gray levels. Note that $1 \leq \mu(x) \leq \kappa + 1$. The theory of the previous section has shown that using large values

¹²That is, small with respect to the post-smoothing scale ρ .

Fig. 10 (Color online) Coherence vs. modified coherence flow for inpainting problems (strong coherence is red, weak is blue)



of μ in the weight function (11) allows to faithfully following the given vector field \vec{c} with just a small amount of diffusion, whereas small values ($\mu \approx 1$) yield a considerable amount of diffusion. Since the first behavior is desirable for strong coherence and the second for a weak one, we suggest taking the image-adapted parameter $\mu = \mu(x)$ in the weight function (11).

Boundary Effects and Modified Coherence Flow In the realm of the single-pass inpainting algorithm we must base the calculation (19) of the coherence vector field $\vec{c}(x)$ on the already inpainted image values $u|_{\Omega(x)}$, $\Omega(x) = \{y \in \Omega : T(y) < T(x)\}$. The problem here is that the smoothing by the Gaussians K_σ and K_ρ reaches into the yet-to-be-inpainted domain $D \setminus \Omega(x)$. A straightforward approach would just continue u by zero, that is, would consider the structure tensor

$$J_\rho(\nabla(\mathbb{1}_{\Omega(x)}u)_\sigma)(x),$$

where $\mathbb{1}_{\Omega(x)}(y) = [y \in \Omega(x)]$ denotes the indicator function of $\Omega(x)$. However, in general, this makes $\partial\Omega(x)$ to a spurious edge, aligning the coherence flow tangentially to it; an effect that is illustrated in Fig. 10(b) for the inpainting problem of Fig. 10(a). The coherence strength is color coded from blue for weak strengths to red for strong ones. Since we know from Sect. 4 that a tangential vector field \vec{c} is rotated into the normal direction $\vec{c}_* = \vec{n}$, we would basically end up with Telea’s algorithm once again.

Also symmetry boundary conditions, as typically used for the image boundary $\partial\Omega$ in digital image processing, would not solve the problem. They tend to align the coherence flow field with the normal to the boundary as is clearly visible at the bottom of Fig. 10(b). In fact, they are related to Neumann boundary conditions $\nabla u \cdot \vec{n} = 0$ which gives $\nabla^\perp u \parallel \vec{n}$. Thus, once more, we would obtain a vector field \vec{c} that yields $\vec{c}_* \approx \vec{n}$ and therefore to the basic behavior of Telea’s algorithm.

Experimentally (see Fig. 10(c)), we have observed that the problem can be solved by appropriately rescaling the smoothing of $\mathbb{1}_{\Omega(x)}u$ before taking gradients. That is, we use the following *modified structure tensor*, which is again

a symmetric positive semi-definite 2×2 -matrix:

$$\hat{J}_{\sigma,\rho}(x) = \frac{(K_\rho \star (\mathbb{1}_{\Omega(x)} \nabla v_\sigma \otimes \nabla v_\sigma))(x)}{(K_\rho \star \mathbb{1}_{\Omega(x)})(x)},$$

$$v_\sigma = \frac{K_\sigma \star (\mathbb{1}_{\Omega(x)}u)}{K_\sigma \star \mathbb{1}_{\Omega(x)}}. \tag{21}$$

Note that the auxiliary function v_σ is well-defined in a vicinity of x . We call

$$\vec{c} = \vec{w}_1, \quad w_1 \text{ normalized eigenvector}$$

to the *minimal* eigenvalue of $\hat{J}_{\sigma,\rho}$, (22)

the *modified coherence direction*, which induces the *modified coherence flow*. The eigenvalues $0 \leq \lambda_1 \leq \lambda_2$ of $\hat{J}_{\sigma,\rho}$ define by (20) a *modified coherence strength* $\mu(x)$.

A deeper mathematical understanding of the modified coherence flow—based on a detailed analysis of a model situation such as Fig. 10—will be the subject of future work.

Summary For the convenience of the reader we summarize the building blocks of our novel inpainting algorithm that we have developed so far:

- the generic single-pass algorithm (1) and (2),
- the distance ordering (3) based on using the fast marching method,
- the exponentially confining weight function (11) with
 - \vec{c} taken as the vector field of modified coherence directions (22),
 - μ taken as the modified coherence strength defined as in (20).

Because of the thus established intricate nonlinear dependence of the weight function on the inpainted result there are many interesting mathematical questions about the mathematics of the limit $h \rightarrow 0, \epsilon \rightarrow 0$, that we leave to future work. However, despite this nonlinear dependence, the algorithm still satisfies the comparison principle that we have generally shown to hold for the generic single-pass algorithm in Sect. 2.

6 Some Remarks on the Implementation

Calculation of the Modified Structure Tensor The most time-consuming part of our algorithm is the calculation of the modified structure tensor (21). It is therefore imperative to organize it by an update procedure, which we describe now for the auxiliary function v_σ . The update procedure for $\hat{J}_{\sigma,\rho}$ is similar.

The smoothing with the Gaussians K_σ is done by truncating it to a quadratic convolution mask of side length 4σ . The corresponding quadratic neighborhood of a point x in Ω_h will be denoted by $Q_\sigma(x)$. Now, let $x \in D_h$ be the actual point that has just been inpainted and $x' \in D_h$ the next one in the distance order established by the fast marching method. Let the functions \hat{v}_σ and χ_σ carry, up to a common scale factor, the actual values of $K_\sigma \star (\mathbb{1}_{\Omega(x)}u)$ resp. $K_\sigma \star \mathbb{1}_{\Omega(x)}$. These functions are updated by the formulas

$$\begin{aligned} \hat{v}_\sigma(y) &\leftarrow \hat{v}_\sigma(y) + K_\sigma(x - y)u(x), \\ \chi_\sigma(y) &\leftarrow \chi_\sigma(y) + K_\sigma(x - y), \quad y \in Q_\sigma(x), \end{aligned}$$

to carry the values of $K_\sigma \star (\mathbb{1}_{\Omega(x')}u)$ resp. $K_\sigma \star \mathbb{1}_{\Omega(x')}$. Then $v_\sigma = \hat{v}_\sigma/\chi_\sigma$ can be calculated to update the modified structure tensor $\hat{J}_{\sigma,\rho}$ in a likewise fashion to become relevant for x' .

Color Images The algorithm so far applies to gray tone images. A straightforward approach to color images is to consider them as a set of three images and apply the algorithm independently to each one. However, to avoid the appearance of spurious colors, one should use a color space which separates luminance as one component from two chrominance components (see, e.g., [3], p. 420).

For the algorithm at hand even better results can be obtained by using one and the same coherence direction \vec{c} for the inpainting of the three color components. Such a direction is derived from a *common* structure tensor which we define, following Weickert's ([28], p. 204) suggestion for coherence-enhancing diffusion, as a convex combination of the structure tensors of the component images. Specifically, for RGB images $u = (u_R, u_G, u_B)$ we form the convex combination by using the same coefficients that define the luminance image of the YIQ and YUV color spaces. (Historically, these coefficients stem from matching color with monochromatic TV, see [13], §6.2.1.). That is, with the luminance image given as

$$u_Y = 0.299u_R + 0.587u_G + 0.114u_B$$

we define the common modified structure tensor of the color image as

$$\hat{J}_{\sigma,\rho}|_u = 0.299\hat{J}_{\sigma,\rho}|_{u_R} + 0.587\hat{J}_{\sigma,\rho}|_{u_G} + 0.114\hat{J}_{\sigma,\rho}|_{u_B}.$$

Note that the amount of work spent for the eigenvalue problems is the same for color and gray tone images. Therefore, color images take just about twice the CPU time needed for inpainting the corresponding luminance image.

7 Experiments and Comparison with Other Methods

Here, we report on a few computational experiments that should help to compare our algorithm with some of the known methods in the literature. For this reason we have extracted noncompressed data images u^0 from the electronically published PDF or postscript versions of the papers that we compare with. The experiments were performed on a notebook with a 2 GHz Pentium M CPU and 1 GB of RAM. The code is written in C and has an interface to Matlab's image processing toolbox.¹³

Choice of the Parameters The execution of the algorithm depends on the choice of four parameters:

- ϵ , the averaging radius.
- κ , the sharpness parameter.
- σ and ρ , the scale parameters for pre- and post-smoothing.

Here, ϵ is the local scale of dependence on the image values; whereas σ and ρ are describing the scale on which details are neglected for the calculation of the direction of the information flow. Finally, κ controls the amount of diffusivity. In view of the last remark in Sect. 4 larger values of κ are allowed for larger radii ϵ .

For inpainting problems with narrow (that is, about 10 px) but elongated inpainting domains, such as Figs. 1, 12, and 13, a good start is made with the *default parameters*

$$(\epsilon, \kappa, \sigma, \rho) = (5 \text{ px}, 25, 1.4 \text{ px}, 4 \text{ px}). \tag{23}$$

Other problems need some interactive experimentation to get a good choice of parameters. However, due to the fast performance of our algorithm, such an experimentation is generally feasible.

Comparison with Telea's [24] Method Recall that our method was constructed from a deeper mathematical understanding of Telea's algorithm with the aim of removing its shortcoming while keeping its computational speed. A comparison of Figs. 1(c) and 3(b) shows the improvement for

¹³To foster further experiments and checks of the reproducibility of our experiments, all the images and a Windows executable of the code with its Matlab driver can be downloaded, for academic purposes only, at the URL <http://www-m3.ma.tum.de/bornemann/InpaintingCodeAndData.zip>. In particular, the directories `Code_and_Images/originals` and `Code_and_Images/compare` contain the original images and the results of the methods we compare with, respectively.

Fig. 11 (Color online) Telea's method vs. our method of coherence based inpainting. Σ is shown in red, ∂D in blue

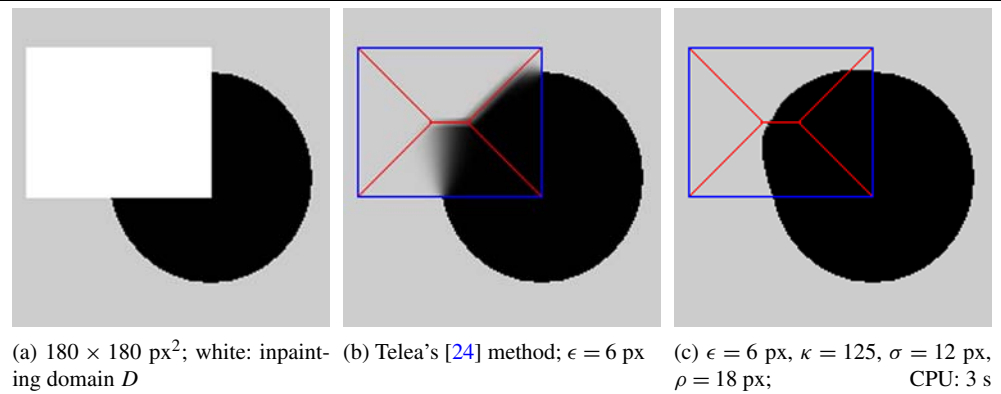


Fig. 12 (Color online) Scratch removal: restoration of an old photograph (compare with [3], Fig. 5)

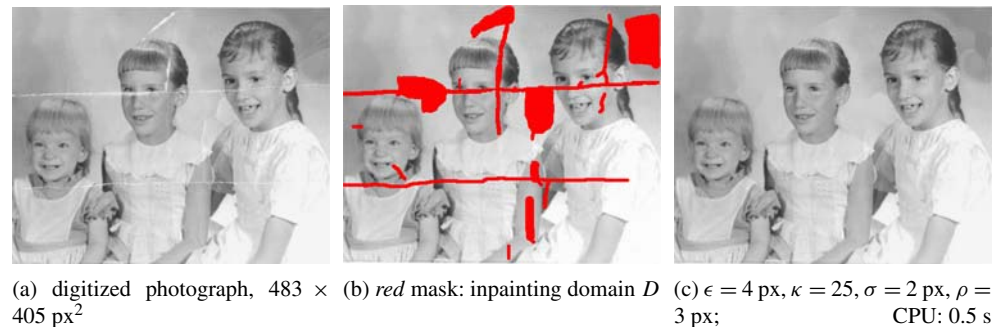
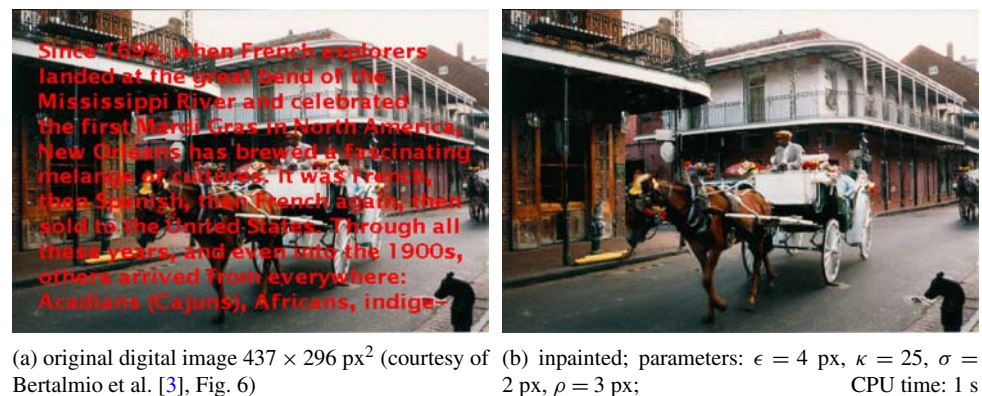


Fig. 13 Removal of superimposed text (compare with [3], Fig. 6)



the restoration of a vandalized natural image. There is virtually no diffusion visible with our algorithm, even the fine flow-like texture of the eyebrow has been inpainted in a visually satisfactory way. In fact, the nonexpert observer might accept Fig. 1(c) as the original image. (However, compare with the actual original one shown in Fig. 1(a).) In contrast, Telea's algorithm introduces strong diffusive artifacts. In this example, the computing time increases by a factor of 5 from Telea's algorithm to our improvement.

Because of the unmatched abilities of the human visual system synthetic inpainting problems are much more unfor- giving. In Sects. 2 and 4 we have already discussed at length the comparison of our method with Telea's for the inpainting of a straight line (see Figs. 4(b) and 8(a)). The inpainting of a curved object is shown in Fig. 11. Even though our method

closes the boundary of the circle not perfectly symmetric, it is reasonable in view of the large inpainting domain of $105 \times 85 \text{ px}^2$. The remarkable sharp closure can be understood as follows: at the inpainting of a point $x \in D$ the modified coherence flow field starts communicating between opposite sides of the yet-to-be-inpainted domain $D \setminus \Omega(x)$ if their distance is below 4ρ .

This result should be compared to the involved variational techniques, such as the Mumford–Shah–Euler model, that have been specifically invented just for the purposes of inpainting curved objects; see the results reported in [11], Fig. 7, or [8], Fig. 6.17, for which no CPU times have been published, however.

Comparison with Bertalmio et al.'s [3] Method We have already discussed the basic principles underlying Bertalmio

Fig. 14 (Color online)
Touch-up: liberate a parrot
(digital image and mask
courtesy of [25], Fig. 6(a))

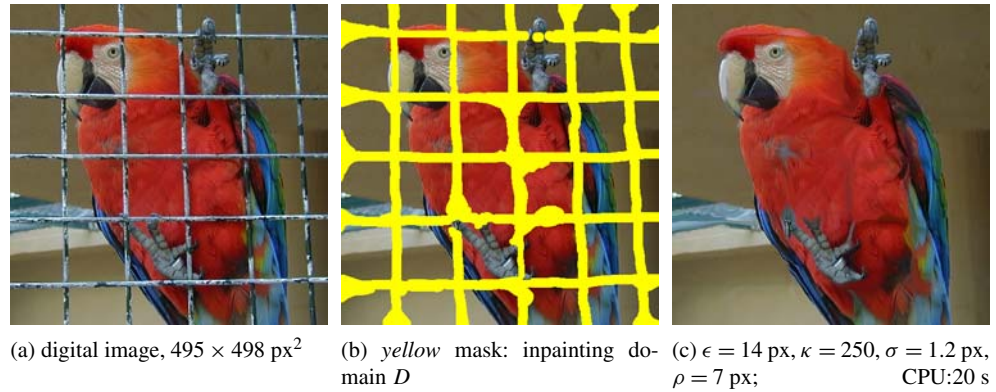
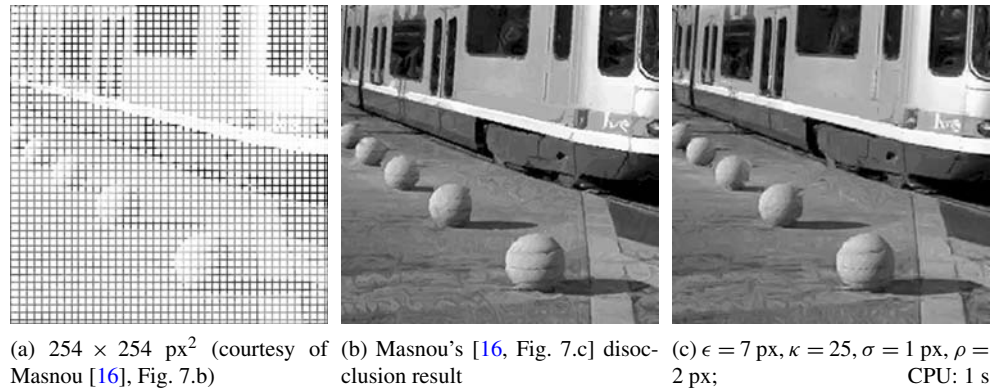


Fig. 15 Disocclusion:
comparison with Masnou’s [16]
variational approach using
isophotes



et al.’s [3] method in the introduction and at the beginning of Sect. 5. Two examples from this celebrated paper have served as a kind of benchmark for several other authors: the removal of scratches in an old photograph [3], Fig. 5, and the removal of superimposed text [3], Fig. 6. Our results are shown in Figs. 12 and 13; their visual quality is on a high level which is absolutely comparable to the results¹⁴ published by Bertalmio et al. These authors report, for a 300 MHz Pentium II PC, CPU times of about 2 min for the first example and about 5 min for the second one. If we concede a factor of 10 due to our faster hardware, our method is nevertheless at least an order of magnitude faster.

Oliveira et al. ([18], Figs. 4 and 5) have also applied their method of repeated convolutions to these two examples. Their result shows considerable artifacts for the text removal. The CPU times are more or less comparable to the ones of our method.

Comparison with Tschumperlé’s [25] Method An interesting iterative method for image inpainting that gives excellent results was recently published by Tschumperlé. Like in the method of Bertalmio et al. [3] the iteration is realized by introducing a formal time variable. His method is related to

ours in its use of the structure tensor. Specifically, in a simplified variant¹⁵ the method realizes the time evolution of a heat equation constrained to the flow lines of the coherence flow,

$$\frac{\partial u(\Phi^s x)}{\partial t} = \frac{\partial^2 u(\Phi^s x)}{\partial s^2},$$

$$u|_{t=0} = \text{some simple interpolation of } u^0,$$

with boundary values given by the image data u^0 . Here, $\Phi^s x$ denotes a parametrization of the flow line through a given point $x \in D$. At each time step the coherence direction is updated by the actual values of u .

Tschumperlé’s method (that also uses at least 4 parameters) was popularized by an open source plug-in to *The GIMP*, the GNU image manipulation program. His example [25], Fig. 5(a) of removing the cage from a digital image of a parrot contributed to this popularization. [25], p. 14 reports a CPU time of 4 min 11 s on a 2.8 GHz PC; our result shown in Fig. 14 is of a similar¹⁶ quality and completes within 20 s—an order of magnitude faster. One could

¹⁴Bertalmio et al.’s results are part of the web material mentioned in Footnote 13.

¹⁵The general variant is a weighted average of diffusion along *all* directions, giving the coherence direction a stronger weight depending on a measure of coherence strength.

¹⁶Tschumperlé’s result is part of the web material mentioned in Footnote 13.

Fig. 16 Details of the comparison in Fig. 15

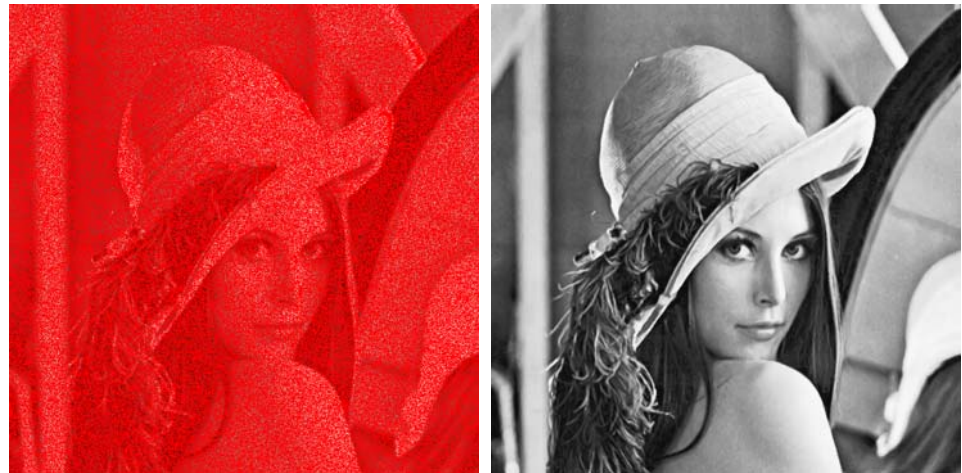


(a) 81×81 px² detail of Fig. 15(a) (b) detail of Fig. 15(b) (c) detail of Fig. 15(c)

Fig. 17 (Color online) Denoising: inpainting of a salt and pepper noise domain, detected by thresholding



(a) immortal digital Lena, 1024×1024 px² (b) Lena with 80% salt & pepper noise



(c) red: inpainting domain D , detected by thresholding (d) inpainted; $\epsilon = 5$ px, $\kappa = 25$, $\sigma = 1$ px, $\rho = 2$ px; CPU time: 20 s

think about using the result of our method as the start image that is iteratively improved by Tschumperlé’s method.

Comparison with Masnou’s [16] Method Masnou and Morel [17] presented a variational approach using level lines (isophotes) by minimizing a discrete version of the func-

tional

$$\int_{\Omega} |\nabla u| \left(\alpha + \beta \left| \operatorname{div} \frac{\nabla u}{|\nabla u|} \right|^p \right) dx, \quad p = 1,$$

subject to the boundary conditions $u|_{\Omega \setminus D} = u^0|_{\Omega \setminus D}$. The relation of this idea to an axiomatic theory of image interpolation has been studied in [4]; details of the theoretical jus-

Fig. 18 Denoising: comparison with the hybrid method suggested by Guichard and Morel [14], §24.4.6



(a) $246 \times 255 \text{ px}^2$ (Guichard and Morel [14], Fig. 1.2) (b) Guichard and Morel's [14, Fig. 1.2] result (c) $\epsilon = 5 \text{ px}, \kappa = 25, \sigma = 1 \text{ px}, \rho = 2 \text{ px}$; CPU: 0.5 s

Fig. 19 (Color online) Details of the comparison in Fig. 18



(a) $81 \times 81 \text{ px}^2$ detail; blue: inpainting domain (b) detail of Fig. 18(b) (c) detail of Fig. 18(c)

tification and the algorithm itself were much later published by Masnou [16]. The discrete version is deeply connected to Gestalt theory, in particular Kanizsa's *amodal completion theory*, that contributes to explaining how the human visual system manages to understand partially occluded objects. The detection of *T-junctions* plays a prominent role in the algorithm; the actual global minimization over all competing level line completions is done by dynamic programming.

Because of its root in image analysis and understanding, Masnou and Morel use the term *disocclusion* instead of *image inpainting*. On the other hand, the above functional with $p = 2$ is basically the *elastica image model* suggested later by Chan et al. [5] for inpainting problems. For the relation of the two approaches see [8], p. 285.

We compare Masnou's method with ours for the partially occluded train shown in Fig. 15(a): only one row resp. column out of six is kept, the rest is turned white. Thus, about 70% of the image have been destroyed. Masnou's [16], Fig. 7(c) result¹⁷ is shown in Fig. 15(b) (no CPU times have been published), ours in Fig. 15(c); details are shown in Fig. 16. The results are of comparable quality (though the

details in Fig. 16 show that straight lines appear less ragged in our method); the original image of the train without occlusion can be found in [16], Fig. 7(a) and as part of the web material mentioned in Footnote 13.

Denoising Salt and Pepper Noise Masnou [16], p. 75 has suggested the use of his disocclusion algorithm for the denoising of images that are affected by salt & pepper (impulse) noise. First, the occluded domain, that is, the inpainting domain D , is detected by tools of mathematical morphology. Then, the image is restored by inpainting. Masnou shows an example with a noise frequency of 10%, see [16], Fig. 5. We will show two experiments with a much larger noise frequency.

In Fig. 17, a $1024 \times 1024 \text{ px}^2$ 8-bit image of Lena (a) has been destroyed by 80% black and white impulse noise (b). Detecting the inpainting domain by masking the 0- and 255-level yields an inpainting domain (c), shown in red, that consists of one large component and five isolated points. The inpainting (d) by our method gives within a CPU time of 20 s a surprisingly good result. One has to zoom-in quite closely to see that the textures of the smooth surfaces look like being truly "painted".

Figure 18(a) shows an 8-bit image taken from [14], Fig. 1.2, reportedly affected by about 75% of salt and pepper noise. Here, we take as inpainting region the domain obtained by thresholding below the 28 and above the 227 level

¹⁷Actually, the uncompressed versions of the occluded and the disoccluded image were extracted from [4], Fig. 10. The electronically published images of [16], Fig. 7, suffer from strongly visible JPEG artifacts.

of gray; this way 56% of the image have been masked. The inpainting result, computed in a CPU time of 0.5 s, is shown in Fig. 18(c). Guichard and Morel suggest in [14], §24.4.6, the following hybrid method for denoising: first they apply a grain filter (that is, the morphological area open-close filter, see [23], §4.4.2 and §8.4) and then a finite difference scheme for the affine invariant morphological scale space equation (AMSS). They report a computing time of about 24 s on a Pentium 200 MHz PC. That is, the computing time is roughly comparable to our suggestion, for which the edges are less frayed, though. However, the surfaces get a slightly brushed texture with our method, which is typical for the use of the coherence flow (see Fig. 19).

References

- Aubert, G., Kornprobst, P.: *Mathematical Problems in Image Processing*. Springer, New York (2002)
- Bertalmio, M., Bertozzi, A.L., Sapiro, G.: Navier-Stokes, fluid dynamics, and image and video inpainting. In: *CVPR'01: Proc. IEEE Int. Conf. on Computer Vision and Pattern Recognition*, Kauai, vol. I, pp. 355–362. IEEE Press, New York (2001)
- Bertalmio, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: *SIGGRAPH '00: Proc. 27th Conf. on Computer Graphics and Interactive Techniques*, New Orleans, pp. 417–424. ACM Press/Addison-Wesley, New York (2000)
- Caselles, V., Masnou, S., Morel, J.-M., Sbert, C.: Image interpolation. In: *Séminaire sur les Équations aux Dérivées Partielles, École Polytech., Palaiseau, 1997–1998. Exp. No. XII, p. 15* (1998), downloadable at the url citeseer.ist.psu.edu/caselles98image.html
- Chan, T.F., Kang, S.H., Shen, J.: Euler's elastica and curvature-based inpainting. *SIAM J. Appl. Math.* **63**(2), 564–592 (2002)
- Chan, T.F., Shen, J.: Inpainting based on nonlinear transport and diffusion. In: *Inverse problems, image analysis, and medical imaging*, New Orleans, LA, 2001. *Contemporary Mathematics*, vol. 313, pp. 53–65. AMS, Providence (2002)
- Chan, T.F., Shen, J.: Mathematical models for local nontexture inpaintings. *SIAM J. Appl. Math.* **62**(3), 1019–1043 (2002)
- Chan, T.F., Shen, J.: *Image Processing and Analysis*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia (2005)
- Chan, T.F., Shen, J.: Variational image inpainting. *Commun. Pure Appl. Math.* **58**(5), 579–619 (2005)
- Dijkstra, E.W.: A note on two problems in connexion with graphs. *Numer. Math.* **1**, 269–271 (1959)
- Esedoglu, S., Shen, J.: Digital inpainting based on the Mumford-Shah-Euler image model. *Eur. J. Appl. Math.* **13**(4), 353–370 (2002)
- Fuchs, F.G.: *Eulers Elastica- und krümmungsbasiertes Inpainting*. Master's thesis, Technische Universität München (2006)
- Gonzalez, R.C., Woods, R.E., Eddins, S.L.: *Digital Image Processing Using Matlab*. Pearson Prentice Hall, Upper Saddle River (2004)
- Guichard, F., Morel, J.-M.: *Image Analysis and PDEs*. Unpublished book. Manuscript version 15/07/2000, 345 pp. (2000), downloadable at the url: citeseer.ist.psu.edu/guichard01image.html
- Kimmel, R.: *Numerical Geometry of Images*. Springer, New York (2004)
- Masnou, S.: Disocclusion: a variational approach using level lines. *IEEE Trans. Image Process.* **11**(2), 68–76 (2002)
- Masnou, S., Morel, J.-M.: Level lines based disocclusion. In: *ICIP'98: Proc. IEEE Int. Conf. on Image Processing*, Chicago, pp. 259–263. IEEE Press, New York (1998)
- Oliveira, M.M., Bowen, B., McKenna, R., Chang, Y.-S.: Fast digital image inpainting. In: *VIP '01: Proc. Int. Conf. on Visualization, Imaging, and Image Processing*, Marbella, Spain, pp. 261–266 (2001)
- Sapiro, G.: *Geometric Partial Differential Equations and Image Analysis*. Cambridge University Press, Cambridge (2001)
- Sethian, J.A.: A fast marching level set method for monotonically advancing fronts. *Proc. Natl. Acad. Sci. U.S.A.* **93**(4), 1591–1595 (1996)
- Sethian, J.A.: *Level Set Methods and Fast Marching Methods*, 2nd edn. Cambridge University Press, Cambridge (1999)
- Soille, P.: Spatial distributions from contour lines: an efficient methodology based on distance transforms. *J. Vis. Commun. Image Rep.* **2**(2), 138–150 (1991)
- Soille, P.: *Morphological Image Analysis*. 2nd edn. Springer, Berlin (2003)
- Telea, A.: An image inpainting technique based on the fast marching method. *J. Graph. Tools* **9**(1), 23–34 (2004)
- Tschumperlé, D.: Fast anisotropic smoothing of multi-valued images using curvature-preserving PDE's. *Int. J. Comput. Vis.* **68**(1), 65–82 (2006)
- Tsitsiklis, J.N.: Efficient algorithms for globally optimal trajectories. *IEEE Trans. Autom. Control* **40**(9), 1528–1538 (1995)
- Weickert, J.: *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart (1998)
- Weickert, J.: Coherence-enhancing diffusion of color images. *Image Vis. Comput.* **17**(3–4), 201–212 (1999)
- Weickert, J.: Coherence-enhancing shock filters. In: Michaelis, B., Krell, G. (eds.) *Pattern Recognition. Lecture Notes in Computer Science*, vol. 2781. Springer, New York (2003)



Folkmar Bornemann received his diploma, Ph.D., and habilitation degrees in mathematics from the Freie Universität Berlin (Germany) in 1989, 1991, and 1997, respectively. He worked as a research scientist at the Zuse Institute in Berlin (Germany) and at the Courant Institute of Mathematical Sciences in New York (USA). Currently, he is full professor of mathematics at the Technische Universität München (Germany). His research interests include numerical analysis and scientific computing, in particular fast methods for nonlinear differential equations with applications to multiscale problems in physics, chemistry, and image science.



Tom März received his diploma degree in mathematics from the Technische Universität München in 2005. Currently, he is a Ph.D. student in the group of Folkmar Bornemann. His research interests include image processing, numerical methods for partial differential equations and hyperbolic functional differential equations.