Image Inpainting based on coherence transport with Adapted distance functions

by

Thomas März
IMAGE INPAINTING BASED ON COHERENCE TRANSPORT WITH ADAPTED DISTANCE FUNCTIONS

THOMAS MÄRZ

Abstract. We discuss an extension of our method Image Inpainting Based on Coherence Transport. For the latter method the pixels of the inpainting domain have to be serialized into an ordered list. Up till now, to induce the serialization we have used the distance to boundary map. But there are inpainting problems where the distance to boundary serialization causes unsatisfactory inpainting results. In the present work we demonstrate cases where we can resolve the difficulties by employing other distance functions which better suit the problem at hand.

Key words. Image Processing, Image Inpainting, Distance Functions

AMS subject classifications. 65D18, 51K99

1. Introduction. Non-texture image inpainting, also termed image interpolation, is the task of determining the values of a digital image for a destroyed, or consciously masked, subregion of the image domain.

The simple idea of the generic single pass method – which forms the basis for our method Image Inpainting Based on Coherence Transport published in [5] – is to fill the inpainting domain by traversing its pixels in an onion peeling fashion from the boundary inwards and thereby setting new image values as weighted means of given or already calculated ones.

Telea has been the first to use such an algorithm in [13]: the pixels are serialized according to their euclidean distance to the boundary of the inpainting domain and the weight is such that image values are propagated mainly along the gradient of the distance map. By his choices of the weight and the pixel serialization, the method of Telea is not adapted to image.

In [5], we addressed the adaption of the weight: our method uses an image dependent weight such that image values are propagated along the estimated tangents of color lines which have been interrupted by the inpainting domain. That way we could improve the quality of the inpainting results compared to Telea (see figure 1.1). Beyond that, we have illustrated in [5] that our method matches the high level of quality of the methods by [4], [10], [9], and [14] while being considerably faster.

But serializing the pixels by their distance to the boundary, as Telea and we in [5] did, is not always a good idea as figure 1.2 illustrates. The image in the middle shows the result which we obtain by using our method with distance to boundary serialization. The diagonal is not restored and the contours of the distance to boundary map indicate that this is due to the bad location of the skeleton which consists of the ridges of the distance map. Before reaching the skeleton, however, the performance is good and the diagonal is continued tangentially. The right image of figure 1.2 shows the result which we get if we use another distance function (for pixel serialization) that suits the problem at hand better (see §4.1, figure 4.1).

The way of serializing the pixels is an important degree of freedom and has already been addressed in patch-based inpainting methods, see e.g. the articles [6], [12], and [3]. For the methods of the latter two papers the user can provide hand-driven geometric constraints and thereby control the filling process.

∗Mathematical Institute, University of Oxford, OX1 3LB, UK. (maerz@maths.ox.ac.uk). Manuscript as of September 16, 2011.
Similarly, the pixel serializations that we discuss here are all induced by a more general type of distance function which respect geometric constraints specified by the user. Our method is still a PDE-based geometric inpainting technique and the new distance functions help to resolve some obstructions which the euclidean distance to boundary serialization entails.

Outline of the Paper. In section 2 we summarize the existing results and give a complete description of the basic algorithm. Section 3 describes the interface of the algorithm. In section 4 we turn to concrete distance functions which are used to serialize the pixels of the inpainting domain. We describe three approaches: distance by harmonic interpolation (§4.1), modified distance to boundary (§4.2), and distance to skeleton (§4.3). The experiments here are restricted to synthetic examples. Section 5 discusses some natural examples.

2. Summary of Existing Results. Our starting point is the generic algorithm for gray tone images. We assume that all gray tone images, seen as functions, take values in the real interval $[0, 255]$ and we assume that gray tones are mapped onto $[0, 255]$ such that the natural order on the interval reflects the order of the shades of gray by their brightness from black ($=0$) to white ($=255$). Moreover, we will distinguish between discrete (digital) notions by using the index $h$ and continuous (analog) notions where we omit the index. Continuous notions are thought of as the high-resolution limit of the corresponding discrete ones. Finally, we identify pixels with their midpoints.

Notation.
a) $\Omega_{0,h}$ is the image domain, the matrix of pixels for the final, restored image $u_h$:
\( \Omega_{0,h} \to [0, 255] \).

b) \( \Omega_h \subset \Omega_{0,h} \) is the inpainting domain whose values of \( u_h \) have to be determined.

c) \( \Omega_{0,h} \setminus \Omega_h \) is the data domain whose values of \( u_h \) are given as \( u_h|_{\Omega_{0,h}\setminus\Omega_h} = u_{0,h} \).

d) \( \partial \Omega_h \subset \Omega_h \) is the discrete boundary, i.e., the set of inpainting pixels that have at least one neighbor in the data domain.

Continuous quantities are defined correspondingly. Finally, we define discrete and continuous \( \epsilon \)-neighborhoods by

\[
B_{\epsilon,k}(x) := \{ y \in \Omega_{0,h} : |y - x| \leq \epsilon \}, \quad B_{\epsilon}(x) := \{ y \in \Omega : |y - x| \leq \epsilon \}.
\]

**Generic Algorithm.** The basic idea is to fill the inpainting domain from its boundary inwards by using weighted means of given or already calculated image values. We assume that the pixels which are to be inpainted have already been serialized, i.e. \( \Omega_h = (x_1, x_2, \ldots, x_N) \) is an ordered list. The following set

\[
B_{\epsilon,2}^<(x_k) := B_{\epsilon,2}(x_k) \setminus \{x_k, \ldots, x_N\}, \quad k = 1, \ldots, N,
\]

denotes the neighborhood of the pixel \( x_k \) consisting only of known or already inpainted pixels. Then, the algorithm reads as follows:

\[
u_h|_{\Omega_{0,h}\setminus\Omega_h} = u_{0,h},
\]

\[
u_h(x_k) = \frac{\sum_{y \in B_{\epsilon,2}^<(x_k)} w(x_k, y) u_h(y)}{\sum_{y \in B_{\epsilon,2}^<(x_k)} w(x_k, y)}, \quad k = 1, \ldots, N. \tag{2.1}
\]

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

\[
\sum_{y \in B_{\epsilon,2}^<(x)} w(x, y) > 0, \quad x \in \Omega_h.
\]

**Pixel Serialization.** In the generic algorithm, any serialization of the pixels can be used which geometrically goes from the boundary inwards in an onion peeling fashion. Now, let \( T : \Omega \to [0, T_{\text{max}}] \) be a map with \( T|_{\Gamma} = 0, \Gamma \subset \partial \Omega \), and which strictly grows into the interior of \( \Omega \). We define the onion peels to be the level lines of \( T \). Using such a map \( T \) or rather a discretized version \( T_h : \Omega_h \to [0, T_{\text{max}}] \), we serialize the pixels of \( \Omega_h \) by

\[
T_h(x_j) < T_h(x_k) \Rightarrow j < k \tag{2.2}
\]

into an ordered list \( \Omega_h = (x_1, x_2, \ldots, x_N) \). Unfortunately, the serialization rule (2.2) cannot order pixels which belong to the same onion peel where the \( T_h \)-values are all equal. Now, in the case of \( T_h(x_k) = T_h(x_j) \) we force the computations of the values \( u_h(x_k) \) and \( u_h(x_j) \) to be independent of each other by redefining the pixel neighborhood in terms of \( T_h \) as

\[
B_{\epsilon,k}^<(x_k) := \{ x \in B_{\epsilon,k}(x_k) : T_h(x) < T_h(x_k) \text{ if } x \in \Omega_h \}.
\]

And hence we can process along an onion peel of \( T_h \) in arbitrary order.
Clearly, there are different possible choices of a suitable $T$. In the article so far this degree of freedom has been fixed by setting $T = d$, where $d$ is the euclidean distance to the boundary $\partial \Omega$,

$$d(x) = \text{dist}(x, \partial \Omega), \quad x \in \Omega.$$ 

As pointed out in the introduction $T = d$ is not always a good choice and in the next section we will use generalized distances $T$.

Weights. Here again, different choices for the weights are possible. We consider weights of the form

$$w(x, y) = \frac{1}{|x - y|} k \left( x, (x - y) \cdot e^{-1} \right)$$

(2.3)

with a kernel $k(x, \eta)$:

a) Normal Transport (Telea's method). Telea, in [13], uses the kernel

$$k(x, \eta) = \frac{|\langle N(x), \eta \rangle|}{|\eta|}, \quad \text{with} \quad N(x) = \frac{\nabla T(x)}{|\nabla T(x)|}.$$ 

Though Telea has taken $T = d$, the kernel carries over to more general $T$.

b) Coherence Transport (our method of [5]). For a given guidance field called $g$ we use the kernel

$$k_{\mu}(x, \eta) = \sqrt{\frac{\pi}{2}} \mu \exp \left( -\frac{\mu^2}{2} \left( g^{\perp}(x), \eta \right)^2 \right).$$

(2.4)

The guidance vector $g(x)$, supposed to be an approximate tangent, is computed by employing structure tensor analysis. The set-up of the structure tensor $S_{\sigma, \rho}(x)$ is as follows:

$$\alpha_{\sigma}(y, x) = \int_{\Omega(x)} K_{\sigma}(y - h) \, dh$$

$$\nu_{\sigma}(y, x) = \frac{1}{\alpha_{\sigma}(y, x)} \cdot \int_{\Omega(x)} K_{\sigma}(y - h) \cdot u(h) \, dh$$

$$S_{\sigma, \rho}(x) = \frac{1}{\alpha_{\rho}(x, x)} \cdot \int_{\Omega(x)} K_{\rho}(x - y) \cdot \nabla y \nu_{\sigma}(y, x) \cdot \nabla y \nu_{\sigma}^T(y, x) \, dy$$

where $\Omega(x) := \{ y \in \Omega : T(y) < T(x) \} \cup \Omega_0 \ \cap \ \Omega$ and $K_{\sigma}$ is a Gaussian kernel. The coherence vector $g(x)$ which enters $k_{\mu}$ as guidance is the eigenvector of $S_{\sigma, \rho}(x)$ w.r.t. the minimal eigenvalue. Since $g$ depends on the image $u$, the weight does here, too.

For more details on the usage of the structure tensor w.r.t. coherence transport inpainting see [5] and [8]. For other applications of structure tensor analysis and the related scale space theory see e.g. [1], [15] and [2].

A Note on the Theory Behind. Because continuous theory is not an issue of this paper, we want at least make some comments. For weights of the form (2.3) we have shown in [5] and [8] that the high-resolution ($h \to 0$) vanishing-viscosity ($\varepsilon \to 0$) limit yields transport/advection PDEs of the form:
a) Normal Transport.
\[ \langle N(x), \nabla u(x) \rangle = 0 \quad \text{in} \quad \Omega \setminus \Sigma, \quad u|_{\partial \Omega} = u_0|_{\partial \Omega}. \]

b) Coherence Transport.
\[ \langle c_\mu(x), \nabla u(x) \rangle = 0 \quad \text{in} \quad \Omega \setminus \Sigma, \quad u|_{\partial \Omega} = u_0|_{\partial \Omega}, \quad \langle c_\mu(x), N(x) \rangle \geq \beta_\mu > 0. \]  

(2.5)

In both cases the set \( \Sigma \) denotes the set where \( N(x) = \nabla T(x) / |\nabla T(x)| \) is singular. The side condition \( \langle c_\mu(x), N(x) \rangle \geq \beta_\mu \) is the continuous counterpart of the pixel serialization: the function \( T \) here plays the role of a Lyapunov function and this condition says that the characteristics of the advection PDE evolve strictly into the interior of \( \Omega \) and stop at \( \Sigma \). The general existence and well-posedness theory for such type of equations and its application to the analog inpainting model above can be found in \([8]\).

Finally, in order to better understand the role of the parameter \( \mu \) of the kernel \( k_\mu \), we cite two results of \([5]\). Firstly, according to \([5]\, \text{theorem 1}\) every weight \( w \) of the form \((2.3)\) corresponds to a transport field \( c \). Secondly, in \([5]\, \text{theorem 2}\), when using the coherence transport kernel \( k_\mu \), we have proved an asymptotic expansion of \( c_\mu(x) \), w.r.t. \( \mu \to \infty \), which implies the following limit behavior

\[
\lim_{\mu \to \infty} c_\mu(x) = \begin{cases} 
  g(x), & \langle g(x), N(x) \rangle > 0 \\
  -g(x), & \langle g(x), N(x) \rangle < 0 \\
  N(x), & \langle g(x), N(x) \rangle = 0 
\end{cases}.
\]  

(2.6)

That means the coherence vector \( g(x) \) guides in fact the coherence transport \((2.5)\) and \( \mu \) is the strength of this guidance.

3. Interface of the Algorithm. In the next section the algorithm will be performed only in its coherence transport version. That is, the weight function \( w \) has the form of \((2.3)\) with the kernel \( k_\mu \) given by equation \((2.4)\). The execution of the coherence transport algorithm depends on the choice of four parameters:
- \( \epsilon \), the averaging radius,
- \( \mu \), the guidance strength of \( k_\mu \),
- \( \sigma \) and \( \rho \), the scale parameters of the smoothing operations in the structure tensor \( S_{\sigma, \rho} \).

Finally, the algorithm is supplied with the data image \( u_0 \) and a sorted list of the pixels which are to be inpainted. Any item of this list is a triple

\[ [i \quad j \quad T_h(i, j)] , \]

whereas \((i, j)\) are the pixel coordinates and the list is sorted in ascending order of the distance values \( T_h(i, j) \). Hence, pixel serialization which means setting up this list is a pre-processing step.

For comparing CPU-times. The experiments in the following sections have been performed using Matlab 7.6 (R2008a) on a MacBook with a 2.33 GHz Intel Core 2 Duo processor and 2 GB of RAM. The codes of the inpainter and that of the Fast Marching based pixel serializers are written in C with a mex-interface to Matlab. For each inpainting experiment we state the overall CPU-time, i.e., serialization plus inpainting.
4. Concrete Distance Functions. The generic algorithm of section 2 depends on a prescribed pixel serialization, which orders the pixels from the boundary inwards. In all previous experiments of [5] the pixels were serialized by their euclidean distance to boundary. The advantage in practice is that for all types of domains the approximate distance to boundary map is easy to generate by the Fast Marching Method (see [11] and [7]). But the disadvantage is that it is not always the best choice if one wants to get a nice looking inpainting result. Here, we present three other ways of setting up a generalized discrete distance function \( T_h \). We show a few synthetic examples where they yield better inpaintings than the distance to boundary.

4.1. Distance by Harmonic Interpolation. The broken diagonal is our first example. Figure 4.1 (b) shows the result when using the distance to boundary \( d_h \). A restored diagonal would be a nicer result but the diagonal is only partly continued correctly. Figure 4.1 (c) – with the contours of \( d_h \) overlaid – demonstrates that the undesired effect is due to the wrong location of the stop set. Figure 4.1 (e) and (h) show the result when using \( T_h \) which we obtain by prescribing a better located stop set (red lines in figure 4.1 (d), (g) ) and harmonic interpolation. Here, the algorithm is able to restore the diagonal. For all three cases we have used the same set of parameters \([\varepsilon, \mu, \sigma, \rho] = [3, 50, 0.5, 5]\).

Now, we describe the construction of the discrete distance function \( T_h \). Here, we can prescribe the location of the stop set arcs and their distance values. Because the boundary is the start set, the discrete distance function \( T_h \) shall equal zero on \( \partial \Omega_h \). In addition, we take at least one or more discrete curves \( \Gamma^k_h \), \( k \in \{1, \ldots, n\} \), which are contained in \( \Omega_h \) and should belong to the stop set \( \Sigma_h \). Moreover, for every \( \Gamma^k_h \) we specify a distance value \( t_k > 0 \). The remainder of \( T_h \) then, is calculated by harmonic interpolation. That is, we solve the discrete Laplace equation

\[
\Delta_h T_h = 0 \quad \text{in} \quad \Omega_h \setminus \Sigma_h , \quad \Sigma_h = \bigcup_{k=1}^n \Gamma^k_h , \quad T_h = 0 \quad \text{on} \quad \partial \Omega_h , \quad T_h = t_k \quad \text{on} \quad \Gamma^k_h , \quad k = 1, \ldots, n .
\]

Hereby, the discretization \( \Delta_h \) of the Laplacian is due to the five-point-stencil

\[
\Delta_h T_h(i, j) = T_h(i-1, j) + T_h(i, j-1) - 4T_h(i, j) + T_h(i, j+1) + T_h(i+1, j) .
\]

The resulting linear system is symmetric, banded, and sparse: we solve it using Matlab’s backslash-operator for sparse matrices.

Since harmonic interpolation provides a minimum and maximum principle, our construction of \( T_h \) can be imagined as setting up a tent roof over the domain \( \Omega_h \) where \( \Gamma^k_h \) are the locations of the tent poles, and every tent pole of \( \Gamma^k_h \) has the length \( t_k \).

In the case of a single curve \( \Gamma_h \) the resulting \( T_h \) is always a valid distance because of the minimum and maximum principle, see figure 4.2. Moreover, the choice the prescribed distance value \( t > 0 \) does not matter in this case: we will of course get different distance functions \( T_h \), but they all induce the same pixel serialization.

Unfortunately, not every choice of curves \( \Gamma^k_h \) with time values \( t_k \), results in a valid distance function: \( T_h \) might have local minima and so would not strictly increase into the interior of \( \Omega_h \). If there are two or more curves \( \Gamma^k_h \), the question whether the resulting \( T_h \) is valid or not depends on the location of the curves in relation to
FIG. 4.1. Broken diagonal. In the first row we have (a) the damaged image with the inpainting domain $\Omega_h$ in white, (b) the result inpainted using euclidean distance to boundary $d_h$, and (c) the result with contours of $d_h$. In the second and third row we have (d), (g) the same damaged image with a specified stop set $\Gamma_h$ in red, (e),(h) the result inpainted using distance by harmonic interpolation $T_h$, and (f),(i) the result with contours of $T_h$. (Image size: $301 \times 301$, damage: 29%).

each other and the differences of their prescribed values $t_k$. Figure 4.3 (b) shows an example with three curves $\Gamma^1_h$, $\Gamma^2_h$ and $\Gamma^3_h$ with

$$t_1 = t_2 = 250 > t_3 = 50.$$ 

Here, all points of $\Gamma^3_h$ are local minima of $T_h$. But, if we keep the geometry of $\Gamma^1_h$, $\Gamma^2_h$ and $\Gamma^3_h$, and change the prescribed distances to

$$t_1 = t_2 = 250 > t_3 = 249,$$

then $T_h$ does not have any minima (see figure 4.3 (c)). Generally speaking, if we have two or more curves $\Gamma^k_h$ and if the values $t_k$ are chosen unfavorably, then the
resulting $T_h$ might possess local minima on some of the $\Gamma^k_h$. Luckily, again owing to the minimum and maximum principle, there is always a set of $t_k$-values such that the resulting $T_h$ is free of local minima.

Let us review our first example. Figure 4.1 (e) illustrates that our $T_h$ – which is that shown in figure 4.2 (b) – works fine. More generally, if we think of $\Sigma_h$ as an initial scratch, which has been dilated to all of $\Omega_h$ over the time $T_h$, then the backward filling-in process, if $\Sigma_h$ is well located, makes the matching opposite sides come together. Figure 4.1 (g)-(i) demonstrates that it is also possible to put $\Sigma_h$ on the diagonal to be restored. Clearly, if we deliberately place $\Sigma_h$ badly – for this example that means $\Sigma_h$ is not on the diagonal and prevents the parts of the diagonal from coming together – then the method cannot produce the diagonal (see figure 4.4, the parameters are $[\varepsilon, \mu, \sigma, \rho] = [3, 50, 0.5, 5]$, as before). This might also be seen as a chance if a user does not want a restored diagonal: choose a $\Sigma_h$ that prevents the edges from coming together but which has a nice shape.

Now, we discuss two further examples. Figure 4.5 (a) shows two broken diagonals with the same inpainting domain as in figure 4.1 (a). We emphasize here that the appearance of an undesired effect depends on how the edge that needs to be continued is located in relation to the inpainting domain. In figure 4.5 (b) – which is inpainted using distance to boundary – the bottom-left-to-top-right diagonal is continued as desired, while the continuation of the top-left-to-bottom-right diagonal

**FIG. 4.2. Single $\Gamma_h$ yields a valid $T_h$**

**FIG. 4.3. (a) shows the domain $\Omega_h$ and the desired stop arcs, red: $\Gamma_1^h$ and $\Gamma_2^h$, blue: $\Gamma_3^h$. For (b) the distances on $\Gamma_k^h$ have been set to $t_1 = t_2 = 250$, $t_3 = 50$ and we can see that the resulting $T_h$ has $\Gamma_3^h$ as local minimum and is thus not valid. For (c) the distances on $\Gamma_k^h$ have been set to $t_1 = t_2 = 250$, $t_3 = 249$ and the resulting $T_h$ is free of minima.**
suffers from a badly located stop set. In Figure 4.5 (d) and (g) we have the damaged image with a single curve \( \Gamma_h \), \( t_1 = 127 \) shown in red (for the latter (g) \( \Gamma_h \) is just a point). Figure 4.5 (e) and (h) show the results each inpainted using the corresponding distance by harmonic interpolation \( T_h \). Again, a good location of \( \Sigma_h \) makes for a good result. For all three cases we have used the same set of parameters \( [\varepsilon, \mu, \sigma, \rho] = [3, 50, 0.5, 5] \).

Figure 4.6 (a) shows a damaged cross junction. A cross junction would, in any case, geometrically be the simplest object for completion. Using the distance to boundary we obtain the result shown in figure 4.6 (b). Here, the stop set which is the central arc of the skeleton (see figure 4.6 (c)), has an unfavorable location because the bar coming from the right-hand side can never reach its counterpart. Setting \( \Gamma_h \) as the center of the cross junction (see figure 4.6 (d)) and using the distance by harmonic interpolation, we are able to restore the cross junction (see figure 4.6 (e)). Which of the bars is closed in the end depends on the coherence strength. The brighter bar has the higher contrast w.r.t. the black background and is thus of stronger coherence. This is the reason why this bar is closed. For both cases we have used the same set of parameters \( [\varepsilon, \mu, \sigma, \rho] = [5, 100, 0.5, 10] \).

4.2. Modified Distance to Boundary. Figure 4.7 (a) shows a damaged stripe pattern. Performing the algorithm with distance to boundary and the set of parameters \( [\varepsilon, \mu, \sigma, \rho] = [5, 100, 0.5, 10] \) yields the result shown in figure 4.7 (b). The difficulty, is that the coherence vector which is tangent to the edges is orthogonal to the lower left and the upper right segment of \( \partial \Omega_h \). Thus, the transport vector \( c \) switches to the normal \( N \) as in the exceptional case of equation 2.6. Figure 4.7 (c) confirms the switch of the transport to \( N \).

To combat this we suggest a modification of the distance map set-up. The euclidean distance to boundary map \( d \) is the viscosity solution of

\[
|\nabla d| = 1 \quad \text{in} \quad \Omega, \quad d|_{\partial \Omega} = 0.
\]

The modification is that we search for the euclidean distance \( d_\ast \) to a subset \( \Gamma \) of the boundary \( \partial \Omega \), i.e.,

\[
|\nabla d_\ast| = 1 \quad \text{in} \quad \Omega, \quad d_\ast|_{\Gamma} = 0. \quad (4.1)
\]
FIG. 4.5. Two broken diagonals. In the first row we have (a) the damaged image with the inpainting domain $\Omega_h$ in white, (b) the result inpainted using euclidean distance to boundary $d_h$, and (c) the result with contours of $d_h$. The result in (b) demonstrates that the undesired effect depends on how edges are located in relation to the inpainting domain. In the second and third row we have (d),(g) the same damaged image with a specified stop set $\Gamma_h$ in red, (e),(h) the result inpainted using distance by harmonic interpolation $T_h$, and (f),(i) the result with contours of $T_h$. (Image size: 301 $\times$ 301, damage: 29%).

We classify the points which shall not belong to $\Gamma$. Assume $x \in \partial \Omega$ satisfies $d_s(x) = 0$, then the boundary normal is given by $N(x) = \nabla d_s(x)$. Now, if we have at $x$

$$\langle g(x), N(x) \rangle^2 = 0,$$

whereas $g$ is the guidance vector, then either there is no guidance ($|g(x)| = 0$) or the guidance vector does not point inwards. Such a boundary point shall not belong to $\Gamma$. In fact, we use the stronger condition

$$0 \leq \langle g(x), N(x) \rangle^2 \leq \gamma$$
with a threshold parameter $0 < \gamma \leq 1$. Complementarily, the set of active boundary points $\Gamma$ is given by

$$\Gamma = \{ x \in \partial \Omega : \langle g(x), N(x) \rangle^2 > \gamma \} .$$

Of course, the new parameter $\gamma$ must be chosen such that $\Gamma$ is not empty.

After having found the active boundary $\Gamma$, we use the Fast Marching Method to solve equation (4.1) for $d_{s,h}$.

We have applied this idea to the stripe pattern: the red lines in figure 4.7 (d) are the active boundary points and the result is shown in figure 4.7 (e). Our standard parameters are $[\varepsilon, \mu, \sigma, \rho] = [5, 100, 0.5, 10]$, while the additional parameter is set to $\gamma = 0.1$. The overlaid contour plot of $d_{s,h}$ in figure 4.7 (f) shows that the guidance vector always points inwards.

4.3. Distance to Skeleton. The third approach to obtaining a serialization is to use the distance to a prescribed stop part of the skeleton. Let $S^k$, $k \in \{1, \ldots, n\}$ be curves in the image domain $\Omega_0$. Those curves $S^k$ which are contained in $\Omega$ will later belong to the skeleton $S$. Let, then, $T_s$ be the viscosity solution of

$$|\nabla T_s| = 1 \quad \text{in} \quad \Omega_0 , \quad T_s = 0 \quad \text{on} \quad S^k , \quad k \in \{1, \ldots, n\} ,$$
and let

\[ T_{s,\text{max}} = \max_{x \in \Omega} T_s(x). \]

The desired distance function is defined by

\[ T(x) = T_{s,\text{max}} - T_s(x), \quad x \in \Omega. \]

Here again, we use the Fast Marching Method to get the approximation \( T_{s,h} \).

Warning: as in the case of harmonic interpolation (see §4.1) one must check if \( T \) is admissible, i.e., if \( T \) is free of local minima.

Figure 4.8 shows the result for the example of the broken diagonal. The red curve in figure 4.8 (a) defines \( S_1^1 \) (discrete) which shall belong to the skeleton. The set of parameters is \([\varepsilon, \mu, \sigma, \rho] = [3, 50, 0.5, 5] \), the same as we used for figure 4.1 (e). The inpainted result here (figure 4.8 (b)) is the same as in figure 4.1 (e), but the distance function has changed (compare figures 4.8 (c) and 4.1 (f)). If there is only one curve \( S_1^1 \) which is completely contained in \( \Omega \), then distance by harmonic interpolation (with \( \Gamma_1^1 = S_1^1 \)) and distance to skeleton will yield very similar results. If there are two or more curves, then distance by harmonic interpolation allows for different distance values on \( \Gamma_k^1 \), while distance to skeleton has exactly one distance value on all of the curves \( S_k^1 \). In contrast to distance to skeleton, the distance by harmonic interpolation method requires \( T_h|_{\partial \Omega_h} = 0 \).
Moreover, since $T_h$ is defined on $\Omega_0$ (the full image domain), we can place $S^k$ outside of the inpainting domain $\Omega$. We use this possibility to restore the stripe pattern in figure 4.9. The parameters are $[\epsilon, \mu, \sigma, \rho] = [5, 100, 0.5, 10]$, the same as we used for figure 4.7. It is obvious from the level lines of $T_h$ (see 4.9 (c)) that the guidance vector always points inwards. Thus, we get the desired result again here.

Remark: The construction here, as well as that of §4.2 produces distance functions whereas not the whole boundary belongs to the start set. Besides that, it is possible that parts of the boundary belong in fact to the stop set.

5. Natural Inpainting Examples. In the previous section we have demonstrated that there are inpainting problems where a distance to boundary induced serialization of the pixels is not favorable and we have constructed other distances which are able to resolve the difficulties which appear.

So far we have considered only synthetic images, because their image geometry is easy to understand. Thus, we were able to construct distances which are adapted to the image or rather to an expected result.

Of course, there are inpainting problems, as shown in figures 1.1 (left) and 5.1 (a), where it is not as easy to set up an adapted distance function. This is because
• the geometry of the image is harder to understand,
• the damaged region is complicated.

Moreover, if the damaged region $\Omega$ consists of many connected components and one wants to use our distance by harmonic interpolation technique (§4.1), one must prescribe arcs $\Gamma^k_h$ for every single component of connectivity: for the example of figure 5.1 (a) that means for every single letter. This can be tedious.

In contrast, the distance to boundary map can be computed fast and easily for every type of inpainting domain. And, if the damage is such that color lines have been broken by scratches (By scratches we mean rather narrow and lengthy damages), the skeleton of $\Omega_h$, being a simplified version of the scratch, is well placed. So, the filling-in process using distance to boundary makes the matching opposite sides come together and produces results of high quality (see figures 1.1 (right) and 5.1 (b)).

But there are also natural inpainting examples with a simple inpainting domain and where we can use the ideas presented in the previous section. Figures 5.2 (taken from [12]) and 5.3 (taken from [6]) show two such examples.

Our method is PDE-based and geometric and thus it does not perform very well on textures in contrast to the patch-based methods of Sun et al. [12] and Criminisi et al. [6]. Therefore, we have chosen examples where the image data is geometry dominated. The second row of figure 5.2 shows the unfavorable result of the distance to boundary serialization, while in the third row we have a nice result from using distance by harmonic interpolation for serializing the pixels. The special choice of the stop set helps to restore the window cross. Similarly, for example 5.3, we can help the inpainter to restore the pole, again employing the distance by harmonic interpolation serialization, see figure 5.4 second and third rows. Criminisi et al. in [6] have already reported on the problem of the distance to boundary map where only the shape of the inpainting domain determines the pixel serialization. In this case our method as well as Criminisi’s cannot restore the pole, see first row of 5.4 and compare to [6, Fig. 11 f]. Criminisi et al. manage this problem with a ”structure-first” fill order. Our modified distance to boundary approach (§4.2) is in a similar vein: the structure of the surrounding data image determines the active boundary. Figure 5.5 shows the result of using this technique.
Fig. 5.2. Remove the pumpkin from the original image (a). In the second row we have (b) the masked image with the inpainting domain $\Omega_h$ in white, (c) the masked image with contours of the euclidean distance to boundary map $d_h$, and (d) the result inpainted using $d_h$. In the third row we have (e), the same masked image with a specified stop set $\Gamma_h$ in red, (f) the masked image with contours of the distance by harmonic interpolation $T_h$, and (g) the result inpainted using $T_h$. Parameters in both cases $[\varepsilon, \mu, \sigma, \rho] = [4, 15, 0.7, 4]$. (Image size: $334 \times 473$, damage: 4%).

Fig. 5.3. Task: remove the lower sign from the original image (both images courtesy of Criminisi et al. [6, Fig. 11]). Results shown in figures 5.4 and 5.5. (Image size: $290 \times 195$, damage: 7%).
Fig. 5.4. Remove the lower sign from the original image shown in figure 5.3 (a). In the first row we have (a) the masked image with the inpainting domain $\Omega_h$ in white, (b) the masked image with contours of the euclidean distance to boundary map $d_h$, and (c) the result inpainted using $d_h$. In the second row we have (d), the same masked image with a horizontal straight line as stop set $\Gamma_h$ in red, (e) the masked image with contours of the distance by harmonic interpolation $T_h$, and (f) the result inpainted using $T_h$. In the third row we have (g), the same masked image with a vertical straight line in the upper part as stop set $\Gamma_h$ in red, (h) the masked image with contours of the distance by harmonic interpolation $T_h$, and (i) the result inpainted using $T_h$. Parameters in all three cases $[\epsilon, \mu, \sigma, \rho] = [7, 25, 0.7, 3]$. (Image size: $290 \times 195$, damage: 7%).
Fig. 5.5. Remove the lower sign from the original image shown in figure 5.3 (a). (a) the masked image with the inpainting domain $\Omega_h$ in white and the active part of the boundary in red (short straight line at the bottom of $\Omega_h$), (b) the masked image with contours of the modified distance to boundary map $d_{\star,h}$, and (c) the result inpainted using $d_{\star,h}$. Parameters $[\epsilon, \mu, \sigma, \rho, \gamma] = [7, 25, 1.25, 5, 0.9]$. (Image size: $290 \times 195$, damage: 7%).

6. Acknowledgements. The author would like to thank Folkmar Bornemann for his advice and the inspiring discussions. This work was supported in part by the Graduiertenkolleg Angewandte Algorithmische Mathematik (GKAAM) funded by the Deutsche Forschungsgemeinschaft (DFG) at the Technische Universität München (TUM), and by Award No KUK-C1-013-04 made by King Abdullah University of Science and Technology (KAUST).

REFERENCES


<table>
<thead>
<tr>
<th>Date</th>
<th>Title</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>29/11</td>
<td>Floating carpets and the delamination of elastic sheets</td>
<td>Wagner</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vella</td>
</tr>
<tr>
<td>30/11</td>
<td>Numerical Study of Liquid Crystal Elastomers by a Mixed Finite</td>
<td>Luo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Calderer</td>
</tr>
<tr>
<td>31/11</td>
<td>The indentation of pressurized elastic shells: From polymeric</td>
<td>Vella</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ajdari</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vaziri</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Boudaoud</td>
</tr>
<tr>
<td>32/11</td>
<td>Wrinking of pressurized elastic shells</td>
<td>Vella</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Ajdari</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Vaziri</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Boudaoud</td>
</tr>
<tr>
<td>33/11</td>
<td>Data assimilation using bayesian filters and B-spline geological</td>
<td>Duan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Farmer</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hoteit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Lu</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Moroz</td>
</tr>
<tr>
<td>34/11</td>
<td>Review of nonlinear Kalman, ensemble and particle filtering with</td>
<td>Luo</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Hoteit</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Duan</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wang</td>
</tr>
<tr>
<td>35/11</td>
<td>Modelling a Tethered Mammalian Sperm Cell undergoing Hyper-</td>
<td>Curtis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Kirkman-Brown</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Connolly</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gaffney</td>
</tr>
<tr>
<td>36/11</td>
<td>A simple mathematical model for investigating the effect of cluster</td>
<td>Zyggalakis</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Roose</td>
</tr>
<tr>
<td>37/11</td>
<td>Frequency jumps in the planar vibrations of an elastic beam</td>
<td>Neukirch</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Frelat</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Goriely</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Maurini</td>
</tr>
<tr>
<td>38/11</td>
<td>Ice-lens formation and cement-induced supercooling in soils and other</td>
<td>Style</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cocks</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Peppin</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wettlaufer</td>
</tr>
<tr>
<td>39/11</td>
<td>An asymptotic theory for the re-equilibration of a micellar</td>
<td>Griffiths</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Bain</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Breward</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Chapman</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Howell</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Water</td>
</tr>
<tr>
<td>40/11</td>
<td>Higher-order numerical methods for stochastic simulation of chemical</td>
<td>Székely</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Burrage</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Erban</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Zyggalakis</td>
</tr>
<tr>
<td>41/11</td>
<td>On the modelling and simulation of a high pressure shift freezing</td>
<td>Smith</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Peppin</td>
</tr>
</tbody>
</table>
44/11 A prototypical model for tensional wrinkling in thin sheets Davidovitch Schroll Vella Adda-Bedia Cerda

45/11 A fibrocontractive mechanochemical model of dermal wound closure incorporating realistic growth factor Murphy Hall Maini McCue McElwain

46/11 A two-compartment mechanochemical model of the roles of transforming growth factor $\beta$ and tissue tension in dermal wound healing Murphy Hall Maini McCue McElwain

47/11 Effects of demographic noise on the synchronization of a metapopulation in a fluctuating environment Lai Newby Bressloff

48/11 High order weak methods for stochastic differential equations based on modified equations Abdulle Cohen Vilmart Zygalakis

49/11 The kinetics of ice-lens growth in porous media Style Peppin

50/11 Wound healing angiogenesis: the clinical implications of a simple mathematical model Flegg Byrne Flegg McElwain

51/11 Wound healing angiogenesis: the clinical implications of a simple mathematical model Du Gunzburger Lehoucq Zhou

Copies of these, and any other OCCAM reports can be obtained from:

Oxford Centre for Collaborative Applied Mathematics
Mathematical Institute
24 - 29 St Giles’
Oxford
OX1 3LB
England
www.maths.ox.ac.uk/occam