In A. Heyden, F. Kahl, C. Olsson, M. Oskarsson, X.-C. Tai (Eds.):
Energy Minimization Methods in Computer Vision and Pattern Recognition.
Lecture Notes in Computer Science, Vol. 8081, pp. 151–164, Springer, Berlin, 2013.
The final publication is available at link.springer.com.

An Optimal Control Approach to Find Sparse Data for Laplace Interpolation

Laurent Hoeltgen, Simon Setzer, and Joachim Weickert

Mathematical Image Analysis Group,
Faculty of Mathematics and Computer Science, Campus E1.7
Saarland University, 66041 Saarbrücken, Germany,
{hoeltgen, setzer, weickert}@mia.uni-saarland.de

Abstract. Finding optimal data for inpainting is a key problem in the context of partial differential equation-based image compression. We present a new model for optimising the data used for the reconstruction by the underlying homogeneous diffusion process. Our approach is based on an optimal control framework with a strictly convex cost functional containing an L_1 term to enforce sparsity of the data and non-convex constraints. We propose a numerical approach that solves a series of convex optimisation problems with linear constraints. Our numerical examples show that it outperforms existing methods with respect to quality and computation time.

Keywords: Laplace Interpolation, Optimal Control, Inpainting, Non-convex Optimisation

1 Introduction

A major challenge in data analysis is to reconstruct a function, for example a 1-D signal or an image from a few data points. In image processing this interpolation problem is called inpainting [2,18]. In most applications one has no influence on the given data and thus improvements can only be made by introducing more powerful reconstruction models. In some interesting applications however, one has the freedom to choose the interpolation data. For instance, in recent approaches to image compression [12,16,22] the authors choose suitable interpolation data for reconstructions via partial differential equations (PDEs). A related approach can also be found in [8,9].

In this paper we present an energy-based approach to optimise the interpolation data. Since this is a very challenging task, we restrict ourselves to a simple reconstruction method, namely homogeneous diffusion inpainting [1,17] which is

also called Laplace interpolation. Further, we will solely focus on the optimisation with respect to its reconstruction quality. Whether the methods presented herein are competitive to other image compression algorithms will be the subject of future work. Homogeneous diffusion inpainting is a fast and very flexible way to reconstruct large scattered datasets, especially in higher dimensions. It considers the following boundary value problem:

$$\begin{aligned} -\Delta u &= 0, \quad \text{on } \Omega \setminus \Omega_K \\ u &= f, \quad \text{on } \Omega_K \\ \partial_n u &= 0, \quad \text{on } \partial\Omega \setminus \partial\Omega_K \end{aligned} \tag{1}$$

where f is a smooth function on some bounded domain $\Omega \subset \mathbb{R}^n$ with a sufficiently regular boundary $\partial\Omega$. The set Ω_K is assumed to be a subset of Ω with positive measure. It represents the known data that will be interpolated by the underlying diffusion process. Finally $\partial_n u$ denotes the derivative of u in outer normal direction. Following [17], we use a binary valued mask c to rewrite (1) in the form

$$\begin{aligned} c(x)\,(u(x) - f(x)) - (1 - c(x))\,\Delta u(x) &= 0, \quad \text{on } \Omega \\ \partial_n u(x) &= 0, \quad \text{on } \partial\Omega \setminus \partial\Omega_K \end{aligned} \tag{2}$$

where c is the characteristic function of the set Ω_K, i.e. $c(x) = 1$ if $x \in \Omega_K$ and 0 otherwise. It was shown in [17] that a careful tuning of the data can lead to tremendous quality gains in the reconstruction. Related investigations can also be found in [12], where the authors present subdivision strategies that exploit non-linear PDEs. We also note that the problem considered in this paper is closely related to the domain of shape, or more generally, topology optimisation. We refer to [14,23] for general introductions to this topic and to [1] for a more specific analysis in the context of image inpainting with homogeneous inpainting.

We will use (2) as a starting point and relax the restrictions placed upon the function c by allowing it to take a continuous range of values. Secondly, we will add a strictly convex energy to the PDE to penalise poor reconstructions and non-sparse sets of interpolation data. Our complete framework can then be regarded as an optimal control problem and presents a large-scale optimisation task with a strictly convex but non-differentiable objective and non-convex constraints. The concrete formulation of the model will be derived in Section 2. In Section 3 we will present a strategy to handle the occurring difficulties in this optimisation problem. The underlying idea of our solver will be to replace the original problem by a series of simpler convex optimisation problems that can be solved efficiently. Section 4 will provide some insight in our framework by stating optimality conditions and requirements for a monotonic convergence towards a solution. Finally, in Section 5, we will present experiments that show the general usefulness of our model both in the 1-D and 2-D setting.

2 A New Model for Finding Optimal Interpolation Data

If $c(x) \in \{0, 1\}$ for all $x \in \Omega$, then (1) and (2) represent equivalent formulations of the same boundary value problem. Interestingly, the latter equation makes also

sense if c is allowed to take a continuous range of values such as $\mathbb{R}$. One may regard continuously-valued functions c as a relaxation of the initial formulation which is much easier to handle. Our goal will be to optimise such masks c with respect to the accuracy of the reconstruction and to the sparsity of the interpolation data. Note that these two objectives cannot be perfectly fulfilled at the same time. If $c\,(x) \equiv 1$, then the reconstruction obtained by solving (2) is perfect. On the other hand, the sparsest possible choice would be $c\,(x) \equiv 0$ which does not allow any reconstruction at all. Therefore, we suggest to complement (2) by an energy that reflects exactly this trade-off between the quality of the reconstruction and the amount of interpolation data. The constrained optimisation problem that we will consider is

$$\begin{aligned} \underset{u,c}{\arg\min}\ E_{\lambda,\varepsilon}\,(u,c) &:= \int_\Omega \frac{1}{2}\,(u\,(x) - f\,(x))^2 + \lambda\,|c\,(x)| + \frac{\varepsilon}{2} c\,(x)^2\,\mathrm{d}x \\ \text{such that:}\quad c\,(x)\,(u\,(x) - f\,(x)) - (1 - c\,(x))\,\Delta u\,(x) &= 0 \quad \text{on } \Omega \\ \partial_n u &= 0 \quad \text{on } \partial\Omega \setminus \partial\Omega_K \end{aligned} \tag{3}$$

with positive parameters λ and ε. The first term of the energy penalises deviations of the reconstruction from the original data f. As in many other imaging applications such as image segmentation [20], we encourage a sparse mask by penalising the L_1 norm of c. The choice of λ lets us steer the sparsity of the mask. For $\lambda = 0$, $c(x) \equiv 1$ is the optimal solution. If λ increases, the mask will become sparser. Finally, $\lambda \to \infty$ will require $c(x) \equiv 0$.

As we will see in the forthcoming section, our numerical solver will require us to solve intermediate problems with a linear instead of non-convex constraint. These problems are related to optimal control problems of the form

$$\begin{aligned} \underset{u,c}{\arg\min}\ \int_\Omega \frac{1}{2}\,(u\,(x) - g\,(x))^2 + \lambda\,|c\,(x)| + \frac{\varepsilon}{2} c\,(x)^2\,\mathrm{d}x \\ \text{such that:}\quad Lu = f + c \end{aligned} \tag{4}$$

with a second-order elliptic and linear differential operator L, a state u, a control variable c and given data g and f. Existence and regularity of such formulations is analysed in [7,24,26]. The problem in (4) may not necessarily have a solution c if $\varepsilon = 0$, unless one resorts to measures [7]. In order to avoid such ill-posed formulations, one has to fix ε at a small positive value. A convergence analysis for $\varepsilon \to 0$ is presented in [26]. Although the analytical discussion of our model in (3) is out of the scope of this paper, we remark that we include the penaliser on the squared L_2 norm of c for the same regularity reasons. Alternatively one could also introduce box constraints of the form $a \leqslant u(x) \leqslant b$ on u as discussed in [24].

3 A Solution Strategy

Our optimal control model proposed in (3) is challenging for two reasons. First of all, the energy contains a non-differentiable term and secondly, the occurring

mixed products $c(x)u(x)$ and $c(x)\Delta u(x)$ render the set of points (u, c) that fulfil the PDE non-convex. In order to devise a solution strategy, we opt for a discretise-first-then-optimise approach. We discretise the data on a regular grid and reorder it into a vector of length n. The discrete analogue of (3) is given by

$$\begin{aligned} &\underset{(u,c)\in\mathbb{R}^n\times\mathbb{R}^n}{\arg\min} \quad \frac{1}{2}\|u-f\|_2^2 + \lambda\|c\|_1 + \frac{\varepsilon}{2}\|c\|_2^2 \\ &\text{such that: } \operatorname{diag}(c)\,(u-f) - (I - \operatorname{diag}(c))\,Du = 0 \end{aligned} \tag{5}$$

where the vectors u, f and c from $\mathbb{R}^n$ denote our reconstruction, the initial data and the corresponding discretisation of the confidence function, respectively. The operator $\operatorname{diag}(\cdot)$ represents the diagonal matrix in $\mathbb{R}^{n\times n}$ with its argument as the main diagonal. Finally, I is the identity matrix and D the discrete version of the Laplace operator with Neumann boundary conditions.

In order to tackle (5) numerically, we will replace it by a series of simpler *convex* optimisation problems. This idea is related to several well-known methods from the literature. One of the simplest strategies is known as *sequential linear programming (SLP)* [13]. SLP methods replace a single nonlinear optimisation problem by a sequence of linear programs. These linear programs are obtained through a first-order Taylor approximation of the objective and the constraints. Even though this method sounds appealing because it significantly reduces the complexity of the problem, it has a major drawback. In order to achieve an accurate result, the solution must lie at a vertex of the linearised constraint space. This requirement is usually not fulfilled. As an alternative, one may consider *linearly constrained Lagrangian* methods *(LCL)* [11,19]. They differ from SLP formulations by the fact that they do not linearise the objective function. They only consider a linear approximation of the constraints and try to minimise the (augmented) Lagrangian of the original problem. In [21] it is shown that under suitable conditions one can achieve quadratic convergence rates with LCL methods.

The main difference between these methods and ours will be the treatment of the objective function. We keep the original energy and merely augment it by an additional penalty term. This way we can circumvent the need to differentiate the objective and provide an alternative approach to LCL methods that often require the involved data to be differentiable. A similar strategy to ours is also briefly mentioned in [25] as a possibility to derive optimality conditions for nonlinear optimal control problems. Furthermore, similar approaches that exploit a partial linearisation of the considered optimisation problem have recently been analysed in [27].

As already mentioned, our goal is to replace the problem in (5) by a series of convex problems that are easier to solve. Therefore, we will replace the constraints by linear counterparts that approximate the original conditions. We define a mapping T which evaluates the constraints for given vectors u and c.

$$T : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}^n, \quad (u, c) \mapsto \operatorname{diag}(c)\,(u-f) - (I - \operatorname{diag}(c))\,Du \tag{6}$$

Its first order approximation around some point $(\bar{u}, \bar{c})$ can be written as

$$T(u,c) \approx T(\bar{u},\bar{c}) + D_u T(\bar{u},\bar{c})(u-\bar{u}) + D_c T(\bar{u},\bar{c})(c-\bar{c}) \tag{7}$$

where $D_u T(\bar{u},\bar{c})$ and $D_c T(\bar{u},\bar{c})$ describe the Jacobi matrices for the differentiation with respect to u and c at position $(\bar{u},\bar{c})$. It is easy to check that

$$D_u T(\bar{u},\bar{c}) = \operatorname{diag}(\bar{c}) - (I - \operatorname{diag}(\bar{c}))\, D \tag{8}$$

$$D_c T(\bar{u},\bar{c}) = \operatorname{diag}(\bar{u} - f + D\bar{u}) \tag{9}$$

If $(\bar{u},\bar{c})$ is a feasible solution of the constraints in (5), then $T(\bar{u},\bar{c}) = 0$ and our initial problem is approximated by

$$\begin{aligned} &\underset{(u,c)\in\mathbb{R}^n\times\mathbb{R}^n}{\arg\min} \ \frac{1}{2}\|u-f\|_2^2 + \lambda\|c\|_1 + \frac{\varepsilon}{2}\|c\|_2^2 \\ &\text{such that: } D_u T(\bar{u},\bar{c})(u-\bar{u}) + D_c T(\bar{u},\bar{c})(c-\bar{c}) = 0 \end{aligned} \tag{10}$$

However, the previous formulation is only reliable for pairs (u,c) in a neighbourhood of $(\bar{u},\bar{c})$. Therefore, we additionally penalise large deviations from $(\bar{u},\bar{c})$ by adding an additional proximal term $\frac{\mu}{2}\|c-\bar{c}\|_2^2$ with a positive weight μ.

$$\begin{aligned} &\underset{(u,c)\in\mathbb{R}^n\times\mathbb{R}^n}{\arg\min} \ \frac{1}{2}\|u-f\|_2^2 + \lambda\|c\|_1 + \frac{\varepsilon}{2}\|c\|_2^2 + \frac{\mu}{2}\|c-\bar{c}\|_2^2 \\ &\text{such that: } D_u T(\bar{u},\bar{c})(u-\bar{u}) + D_c T(\bar{u},\bar{c})(c-\bar{c}) = 0 \end{aligned} \tag{11}$$

Note that u depends continuously on c. Therefore, there is no need to introduce an additional proximal term for this variable. For the sake of brevity, we define $A := D_u T(\bar{u},\bar{c})$ and $B := D_c T(\bar{u},\bar{c})$. A simple computation yields $A\bar{u} + B\bar{c} = \operatorname{diag}(\bar{c})(I+D)\,\bar{u} =: g$. This leads us to the final form of our discrete approximation of (3):

$$\begin{aligned} &\underset{(u,c)\in\mathbb{R}^n\times\mathbb{R}^n}{\arg\min} \ \frac{1}{2}\|u-f\|_2^2 + \lambda\|c\|_1 + \frac{\varepsilon}{2}\|c\|_2^2 + \frac{\mu}{2}\|c-\bar{c}\|_2^2 \\ &\text{such that: } Au + Bc = g \end{aligned} \tag{12}$$

Equation (12) is a constrained optimisation problem with a strictly convex cost and linear constraints. Such problems are well studied and many highly efficient algorithms exist. Note that A is the system matrix one would obtain by discretising (2) with known and fixed c and sought u. In [16] it was shown that this matrix is invertible if all the c_i take only the values 0 or 1 with at least one $c_i = 1$. However, it is easy to see that the same argument carries over to our case with the requirement that $c_i > 0$ for at least one i.

In order to solve the problem occurring in (12), we use a primal-dual algorithm for convex problems from Esser et al. [10] and Chambolle and Pock [5] where it is referred to as PDHGMu and Algorithm 1, respectively. For convex functions F, G and a linear operator K this algorithm solves

$$\underset{x}{\arg\min} \ G(x) + F(Kx) \tag{13}$$

by computing iteratively

$$\begin{aligned} y^{k+1} &= \arg\min_y \frac{1}{2}\|y - \left(y^k + \sigma K \hat{x}^k\right)\|_2^2 + \sigma F^*(y) \\ x^{k+1} &= \arg\min_x \frac{1}{2}\|x - \left(x^k - \tau K^\top y^{k+1}\right)\|_2^2 + \tau G(x) \\ \hat{x}^{k+1} &= x^{k+1} + \theta\left(x^{k+1} - x^k\right) \end{aligned} \tag{14}$$

where F^* is the Fenchel conjugate of F. It was shown in [5] that if $\tau\sigma\|K\|_2^2 < 1$, $\theta \in [0,1]$ and if the primal problem from (13) as well as its dual have a solution, the sequences $(x^k)_k$, $(y^k)_k$ generated by the above algorithm converge to a solution of the primal and dual problem, respectively. We apply this algorithm with

$$G(u,c) := \frac{1}{2}\|u - f\|_2^2 + \lambda\|c\|_1 + \frac{\varepsilon}{2}\|c\|_2^2 + \frac{\mu}{2}\|c - \bar{c}\|_2^2 \tag{15}$$

$$F(Au + Bc) := \iota_g(Au + Bc) \tag{16}$$

where the indicator function ι_g is given by

$$\iota_g(x) := \begin{cases} 0, & \text{if } x = g, \\ \infty, & \text{else.} \end{cases} \tag{17}$$

We note that $F^*(x) = \langle x, g\rangle$. This gives us the algorithm depicted in Algorithm 1. Note that the updates of y^k and u^k only require matrix-vector products and can be carried out in a very fast way. The update of c^k can also be expressed in

Algorithm 1: Minimisation strategy for solving (12).

Input : N number of iterations.
Output: Vectors u and c solving (12)

Choose τ, $\sigma > 0$ such that $\sigma\tau\|\begin{pmatrix} A & B \end{pmatrix}\|_2^2 < 1$ and $\theta \in [0,1]$.
Set u^0, c^0, y^0 arbitrary and set $\hat{u}^0 = u^0$ and $\hat{c}^0 = c^0$.
for k **from** *1* **to** N **do**
  $y^{k+1} = \arg\min_z \ \frac{1}{2}\|z - \left(y^k + \sigma\left(A\hat{u}^k + B\hat{c}^k\right)\right)\|_2^2 + \sigma\langle z, g\rangle$
  $u^{k+1} = \arg\min_z \ \frac{1}{2}\|z - \left(u^k - \tau A^\top y^{k+1}\right)\|_2^2 + \frac{\tau}{2}\|z - f\|_2^2$
  $c^{k+1} = \arg\min_z \ \frac{1}{2}\|z - \left(c^k - \tau B^\top y^{k+1}\right)\|_2^2 + \tau\left(\frac{\varepsilon}{2}\|z\|_2^2 + \frac{\mu}{2}\|z - \bar{c}\|_2^2 + \lambda\|z\|_1\right)$
  $\hat{u}^{k+1} = u^{k+1} + \theta\left(u^{k+1} - u^k\right)$
  $\hat{c}^{k+1} = c^{k+1} + \theta\left(c^{k+1} - c^k\right)$
end
Set $c \leftarrow c^N$ and $u \leftarrow u^N$.
Return *Optimal solutions c and u.*

closed form using the soft shrinkage operator shrink_α given by

$$\text{shrink}_\alpha(x) := \text{sgn}(x) \cdot \max\left(|x| - \alpha, 0\right) \tag{18}$$

For vector-valued arguments, the shrinkage is performed componentwise. With this definition, updating c comes down to computing

$$c^{k+1} = \operatorname{shrink}_{\frac{\tau\lambda}{1+\tau\varepsilon+\tau\mu}} \left(\frac{c - \tau B^\top y^{k+1} + \tau\mu\bar{c}}{1 + \tau\varepsilon + \tau\mu} \right) \tag{19}$$

which again consists only of matrix vector multiplications and can be carried out in a very fast way.

It is important to note that the pair (u, c) obtained from Algorithm 1 is in general not a feasible point for the problem stated in (3). It only represents a solution of (12). Therefore, we discard the vector u and plug c into the constraint of (3) to obtain a linear system of equations which can be solved to obtain a new feasible pair $(\bar{u}, \bar{c})$. Using these vectors, we compute a new first-order approximation of the previously defined function T and repeat all the steps. This iterative scheme is repeated until convergence. The complete algorithm to solve (5) is given in Algorithm 2. Let us remark at this point that matrix factorisation

Algorithm 2: Minimisation strategy for solving (5).

Input : Initial mask c, parameters λ, ε, μ and N the number of iterations.
Output: Vectors u and c solving (5)

for k **from** *1* **to** N **do**
  Set $\bar{c}^k \leftarrow c$.
  Solve $\left(\operatorname{diag}(\bar{c}^k) - \left(I - \operatorname{diag}(\bar{c}^k)\right) D\right) \bar{u}^k = \operatorname{diag}(\bar{c}^k) f$ for $\bar{u}^k$.
  Compute first order approximation of $T\,(u, c)$ around the $\left(\bar{u}^k, \bar{c}^k\right)$.
  Obtain c by solving (12) with Algorithm 1.
end
Compute u from c.
Return *Optimal solutions c and u.*

and completion problems have a similar non-convex structure as the problems presented in this work. Hence, similar methods as ours were recently proposed in [15,27,28].

4 Theoretical Properties of Our New Scheme

4.1 Optimality Conditions

The linearised problem from (12) seeks the minimiser of a strictly convex energy under linear constraints. It is a well known fact that for such problems the Karush-Kuhn-Tucker conditions yield necessary and sufficient optimality conditions. They are given by the following lemma.

Lemma 1. *The Lagrangian associated with the optimisation problem from* (12) *is given by*

$$L\,(u, c, p) := \frac{1}{2}\|u - f\|_2^2 + \lambda\|c\|_1 + \frac{\varepsilon}{2}\|c\|_2^2 + \frac{\mu}{2}\|c - \bar{c}\|_2^2 + \langle p, Au + Bc - g\rangle \ . \tag{20}$$

The corresponding first-order optimality conditions are given by

$$\begin{aligned} u - f + A^\top p &= 0 \\ \lambda \partial \left(\|\cdot\|_1 \right)(c) + \varepsilon c + \mu \left(c - \bar{c} \right) + B^\top p &\ni 0 \\ Au + Bc - g &= 0 \end{aligned} \tag{21}$$

where $\partial \left(\|\cdot\|_1 \right)(c)$ denotes the subdifferential of $\|\cdot\|_1$ at point c.

In a similar manner, one can also derive necessary optimality conditions for (5). We refer to [3,6,25] for more details.

Lemma 2. *The following system of equations yields necessary optimality conditions for (5). For a solution $(u, c) \in \mathbb{R}^n \times \mathbb{R}^n$ of (5) there must exist a $p \in \mathbb{R}^n$ such that the following relations are fulfilled.*

$$\begin{aligned} u - f + D_u T \left(u, c \right)^\top p &= 0 \\ \lambda \partial \left(\|\cdot\|_1 \right)(c) + \varepsilon c + D_c T \left(u, c \right)^\top p &\ni 0 \\ T \left(u, c \right) &= 0 \end{aligned} \tag{22}$$

Using the previous two lemmas we are able to show that fixed points of our Algorithm 2 fulfil the necessary optimality conditions.

Proposition 1. *If Algorithm 1 has reached a fixed point with respect to its input (e.g. $u = \bar{u}^k$ and $c = \bar{c}^k$ in lines 5 and 7 of Algorithm 2), then u and c also fulfil the necessary optimality conditions from Lemma 2.*

Proof. By construction, u and c fulfil the optimality conditions from (21). Since $c = \bar{c}^k$ and $u = \bar{u}^k$, it follows that

$$A = D_u T \left(\bar{u}^k, \bar{c}^k \right) = D_u T \left(u, c \right) \quad \text{and} \quad B = D_c T \left(\bar{u}^k, \bar{c}^k \right) = D_c T \left(u, c \right) \tag{23}$$

and hence, the first two equations from (21) and (22) coincide. Further, $\bar{u}^k$ and $\bar{c}^k$ are by construction feasible, thus it follows that $T \left(u, c \right) = T \left(\bar{u}^k, \bar{c}^k \right) = 0$. □

The following proposition states under which condition the iterates of Algorithm 2 will monotonically decrease the energy.

Proposition 2. *If we denote the solutions obtained from Algorithm 1 by (u, c) and let $\tilde{u}$ fulfil $T \left(\tilde{u}, c \right) = 0$, then $(\tilde{u}, c)$ will be feasible iterates that decrease the energy in every iteration as long as the following condition is valid.*

$$\frac{1}{2} \left(\|\tilde{u} - f\|_2^2 - \|\bar{u}^k - f\|_2^2 \right) \leqslant \lambda \|\bar{c}^k\|_1 + \frac{\varepsilon}{2} \|\bar{c}^k\|_2^2 - \left(\lambda \|c\|_1 + \frac{\varepsilon}{2} \|c\|_2^2 \right) \tag{24}$$

Proof. By definition of $(\hat{u}, c)$ and $\left(\bar{u}^k, \bar{c}^k \right)$ we have

$$\begin{aligned} \frac{1}{2} \|\hat{u} - f\|_2^2 + \lambda \|c\|_1 + \frac{\varepsilon}{2} \|c\|_2^2 &\leqslant \frac{1}{2} \|\hat{u} - f\|_2^2 + \lambda \|c\|_1 + \frac{\varepsilon}{2} \|c\|_2^2 + \frac{\mu}{2} \|c - \bar{c}^k\|_2^2 \\ &\leqslant \frac{1}{2} \|\bar{u}^k - f\|_2^2 + \lambda \|\bar{c}^k\|_1 + \frac{\varepsilon}{2} \|\bar{c}^k\|_2^2 \,. \end{aligned} \tag{25}$$

Replacing $\hat{u}$ by $\tilde{u}$ and reordering the terms yields the sought expression. □

Equation (24) allows an interesting interpretation. The left-hand side can be seen as the loss in accuracy whereas the right-hand side can be considered as the gain in sparseness. Thus, the energy will be decreasing as long as the gain in sparseness outweighs the loss in precision.

5 Experiments

5.1 Signals in 1-D

To demonstrate the performance of our approach we have chosen the piecewise polynomial and non-continuous signal Piece-Polynomial from the WAVELAB 850 Toolbox [4] and normalised it to the interval $[0, 1]$ to ease the simultaneous visualisation of signal, reconstruction and mask. The result is shown in Figure 1. Note that the obtained mask is sparse and the non-zero entries are placed at positions where one would naturally expect them, e.g. two mask points are used to encode a step in the signal. Also note the excellent quality of the reconstruction.

We initialise our method with u being the original signal and a full mask, i.e. $c_i = 1$ for all i, and use the parameters $\varepsilon = 10^{-9}$, $\mu = 1.0$, $\lambda = 0.02$. For Algorithm 1 we set $\theta = 1$ and $\tau = 0.25$. In order to fulfil the step length constraint $\tau\sigma L^2 < 1$ where $L = \|(A\ B)\|$ we approximate L through power iterations and set $\sigma = \left(\left(L^2 + 0.1\right)\tau\right)^{-1}$. The method aborted when the distance between two consecutive iterates dropped below $3 \cdot 10^{-16}$. In order to reach this precision we required about 225000 iterations of Algorithm 1. After 630 iterations of Algorithm 2 the distance between two consecutive iterates c had dropped below 10^{-15} at which point the algorithm stopped. The whole algorithm was implemented Matlab with Algorithm 1 as a mex function written in ANSI C. All the tests were done on a desktop PC with an Intel Xeon processor (3.2GHz) and 24GB of memory. The total runtime was roughly 10 minutes. The squared Euclidean distance between the input signal and the reconstruction is 0.0377.

5.2 Images in 2-D

To show that our approach performs as well on 2-D data sets as it does on 1-D signals, we apply our algorithm to three different test images and compare our method to the state-of-the-art approach from [17]. In [17] the authors proposed a greedy method, called stochastic sparsification, that iteratively selects a set of candidate points and discards those pixels that yield the smallest increase in the error when removed from the mask. This step is repeated until a desired density is reached. In a second step, called non-local pixel exchange, mask and non-mask pixels are swapped. If the reconstruction error increases, the swap is undone, otherwise it is kept. This latter step is repeated until the desired error or the maximal number of swaps is reached.

The results of our method are depicted in Fig. 2 and a summary with a comparison to the approach from [17] is given in Tab. 1. As an error measure we

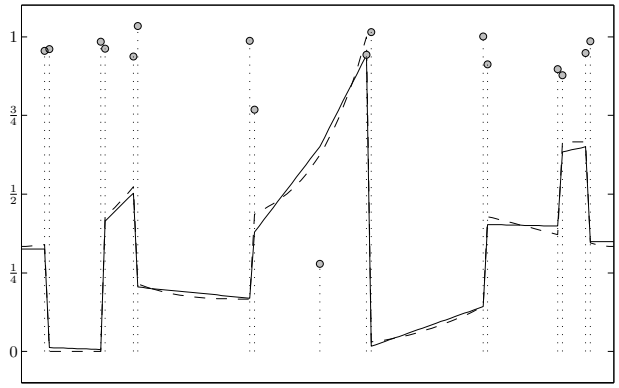

Fig. 1. The original signal (dashed line), the reconstruction (solid line) as well as the used mask (grey dots). As expected, the mask is sparse (17 non-zero entries out of 128) and not binary-valued. The mask point in the middle of the signal with the smallest value allows to better adapt to the curvature of the input signal by blending the diffusion result with the data. Also note that the mask entries neatly align with the discontinuities of the signal.

use the mean squared error (MSE) which is computed by

$$\mathrm{MSE}\,(f,u) := \frac{1}{n}\sum_{i=1}^{n}(f_i - u_i)^2 \tag{26}$$

where the two images f and u to be compared have been reshaped into vectors of length n. For the computation of the MSE we assume that the image values lie in the interval $[0, 255]$.

The third column in Tab. 1 represents the MSE by using the data as it is returned by Algorithm 2. Binarising the mask by thresholding it at 0.01 significantly increases the reconstruction error as we can see in the fourth column of Tab. 1. In [17] an additional optimisation step on the interpolation data was presented. The idea is to decrease the reconstruction error by optimising the image values for a fixed mask. This so called grey value optimisation (GVO) can clearly be used as a postprocessing step in combination with our masks.

The GVO is a least squares formulation to minimise the reconstruction error. It solves

$$\underset{(u,g)\in\mathbb{R}^n\times\mathbb{R}^n}{\arg\min} \|u-f\|_2^2 \quad \text{such that} \quad \mathrm{diag}\,(c)\,(u-g) - (I - \mathrm{diag}\,(c))\,Du = 0 \tag{27}$$

with a fixed c and known image f. Let us define the matrix $M := \mathrm{diag}\,(c) - (I - \mathrm{diag}\,(c))\,D$. Note that this matrix is identical to $D_u T\,(\bar{u}, \bar{c})$ given in (8) for which we know that it is an invertible matrix as long as at least one entry in c is non-zero. Using that $u = M^{-1}\,\mathrm{diag}(c)g$, it follows that the GVO can be formulated as

$$g = \underset{x\in\mathbb{R}^n}{\arg\min}\, \frac{1}{2}\|M^{-1}\,\mathrm{diag}\,(c)\,x - f\|_2^2\,. \tag{28}$$

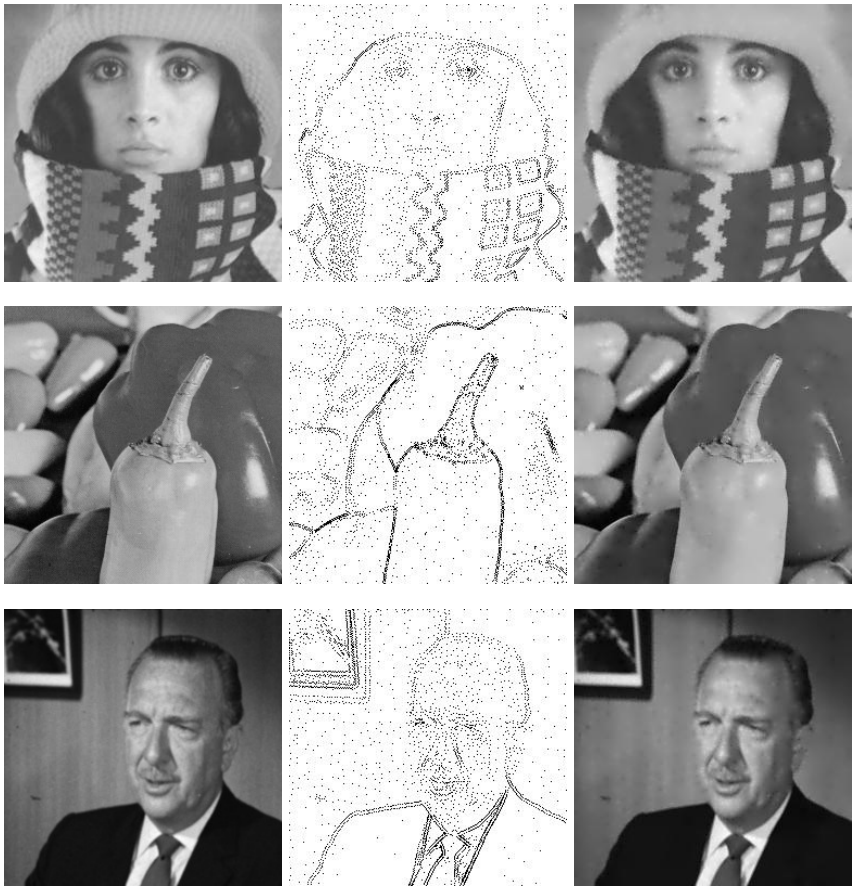

Fig. 2. First row (from left to right): Test image *Trui* (256×256), mask with the locations of the interpolation points (marked in black) with 4.95% of non-zero entries, reconstruction with grey-value optimisation and binarised mask (MSE = 16.95). Note that the bright spots visible in the face are an artifact stemming from the fact that the Laplace operator is used. Parameters: $\lambda = 0.0036$, $\mu = 1.25$, $\varepsilon = 10^{-4}$, 3600 outer iterations and 25000 iterations of Algorithm 1. The parameter choices for the Algorithm 1 are identical to those mentioned in the caption of Figure 1. As initialisation we used $u = f$ and $c \equiv 1$. **Second row (from left to right):** Test image *Peppers* (256×256), mask with the locations of the interpolation points (marked in black) with 5.02% of non-zero entries, reconstruction with grey-value optimisation and binarised mask (MSE = 18.44). Parameters: $\lambda = 0.0034$, $\mu = 0.01$, $\varepsilon = 10^{-9}$, 130 outer iterations and 100000 iterations of Algorithm 1. The remaining parameters are identical as in the experiment with *Trui*. **Third row (from left to right):** Test image *Walter* (256×256), mask with the locations of the interpolation points (marked in black) with 5.00% of non-zero entries, reconstruction with grey-value optimisation and binary mask (MSE = 7.59). Parameters: $\lambda = 0.0018$, 430 outer iterations and 100000 iterations of Algorithm 1. All the other parameters are identical to those used for the *Peppers* experiment.

Table 1. MSE for our experimental results. If combined with a grey value optimisation, our binarised mask is always better than the one obtained in [17]. Note that the mask density in [17] was exactly 5% for each image. For our experiments we had a density of 4.95% for *Trui*, 5.02% for *Peppers* and 5.00% for *Walter*. The best results for each image are marked in boldface. Unavailable results are marked with a —.

Image	Algorithm	Continuous c	Binary c	Continuous c with GVO	Binary c with GVO
Trui	Our method	17.41	47.89	16.98	**16.95**
	Method of [17]	—	23.21	—	17.17
Peppers	Our method	18.91	30.51	**18.44**	**18.44**
	Method of [17]	—	—	—	19.38
Walter	Our method	07.81	19.61	**07.59**	**07.59**
	Method of [17]	—	—	—	08.14

Applying the GVO to the continuous masks only yields minimal gains. However, it completely compensates for the loss of accuracy by binarising the mask. This is a remarkable observation and greatly increases the potential usefulness of this method for image compression tasks since it cuts the necessary amount of data to be stored for the reconstruction in half. Let us emphasise that neither the binarised nor the results with additional GVO represent solutions of our optimal control formulation. Even though the binarised mask fulfils the constraints of our minimisation problem it yields a higher energy. The results with GVO do not, in general, fulfil the constraints of (5). The results from Tab. 1 clearly show the superiority of our approach. Except for *Trui*, our continuous mask already improves the quality from [17]. With the additional GVO we can further reduce our error measures and are able to outperform the results from [17] in every test case.

A Greedy Speed-Up Strategy. In the above experiments we used very large numbers of iterations in order to find the optimal error. This results in computation times of several hours, similarly as the methods in [17]. For practical purposes one can significantly reduce the number of iterations in Algorithm 1. Moreover, the results from Tab. 1 suggest an interesting heuristic to further speed up the computation of the mask. Our algorithm, as it is presented in this paper, starts with a full mask which is made gradually sparser. Since the smallest errors are obtained with a binarised mask combined with a GVO, it is not really necessary to know the exact value of the mask at every position. Only if it will be 0 or not. Therefore, one can threshold the mask during the iterations and check whether this thresholded version differs significantly from the thresholded mask of the previous iterate. If not, one aborts the iteration. Using this heuristic with $\lambda = 0.00325$, $\mu = 0.01$, $\varepsilon = 10^{-9}$, 1250 iterations of Algorithm 1 and 50 outer iterations we obtain a binary mask with a density of 5.01%. In combination with GVO, the MSE for the Peppers image was 19.38, which is identical to the

result from [17]. The total runtime was 272 seconds. Even though the obtained mask yields a slightly larger error when compared to the results from Tab. 1, the runtime was reduced from 15 hours down to less than 5 minutes. This corresponds to a speed-up factor of 180.

6 Conclusion

We present a new model for the determination of optimal interpolation data for homogeneous diffusion inpainting. The formulation contains basically only a single parameter with a very intuitive meaning. An iterative strategy to solve the corresponding discrete optimisation problem is also derived. Numerical experiments show the usefulness of the suggested method. We note that the our method is not restricted to the Laplacian as differential operator. An extension using more powerful operators is possible. Therefore, a thorough theoretical analysis of the framework presented in this work as well as the extension to other PDE-based interpolation methods for image compression tasks will be the subject of future work. We also note that our method is capable of handling non-differentiable penalties on the reconstruction, for example the L_1 norm. This will also be the topic of future work.

References

1. Belhachmi, Z., Bucur, D., Burgeth, B., Weickert, J.: How to choose interpolation data in images. SIAM Journal on Applied Mathematics 70(1), 333–352 (2009)
2. Bertalmío, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proc. 27th Annual Conference on Computer Graphics and Interactive Techniques. pp. 417–424. ACM Press/Addison-Wesley Publishing Company (2000)
3. Bonnans, J.F., Shapiro, A.: Perturbation Analysis of Optimization Problems. Springer Series in Operations Research, Springer (2000)
4. Buckheit, J., Chen, S.S., Donoho, D., Huo, X., Johnstone, I., Levi, O., Scargle, J., Yu, T.: WAVELAB 850 toolbox for matlab (2012), `http://www-stat.stanford.edu/~wavelab/Wavelab_850/download.html`
5. Chambolle, A., Pock, T.: A first order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision 40(1), 120–145 (2011)
6. Clarke, F.H.: Optimization and Nonsmooth Analysis. SIAM (1990)
7. Clason, C., Kunisch, K.: A duality-based approach to elliptic control problems in non-reflexive banach spaces. ESAIM: Control, Optimisation and Calculus of Variations 17(1), 243–266 (2011)
8. Demaret, L., Dyn, N., Iske, A.: Image compression by linear splines over adaptive triangulations. Signal Processing 86(7), 1604–1616 (2006)
9. Demaret, L., Iske, A.: Advances in digital image compression by adaptive thinning. Annals of the MCFA 3, 105–109 (2004)
10. Esser, E., Zhang, X., Chan, T.F.: A general framework for a class of first order primal-dual algorithms for convex optimization in imaging science. SIAM Journal on Imaging Sciences 3(4), 1015–1046 (2010)

11. Friedlander, M.P., Saunders, M.A.: A globally convergent linearly constrained lagrangian method for nonlinear optimization. SIAM Journal on Optimization 15(3), 863–897 (2005)
12. Galić, I., Weickert, J., Welk, M., Bruhn, A., Belyaev, A., Seidel, H.P.: Image compression with anisotropic diffusion. Journal of Mathematical Imaging and Vision 31(2-3), 255–269 (2008)
13. Griffith, R.E., Stewart, R.A.: A nonlinear programming technique for the optimization of continuous processing systems. Management Science 7(4), 379–392 (1961)
14. Haslinger, J., Mäkinen, R.A.E.: Introduction to Shape Optimization: Theory, Approximation, and Computation. SIAM (1987)
15. Lin, C.J.: Projected gradient methods for nonnegative matrix factorization. Neural Computation 19(10), 2756–2779 (2007)
16. Mainberger, M., Bruhn, A., Weickert, J., Forchhammer, S.: Edge-based image compression of cartoon-like images with homogeneous diffusion. Pattern Recognition 44(9), 1859–1873 (2011)
17. Mainberger, M., Hoffmann, S., Weickert, J., Tang, C.H., Johannsen, D., Neumann, F., Doerr, B.: Optimising spatial and tonal data for homogeneous diffusion inpainting. In: Bruckstein, A.M., ter Haar Romeny, B., Bronstein, A.M., Bronstein, M.M. (eds.) Proc. Scale Space and Variational Methods in Computer Vision, LNCS, vol. 6667. Springer, Berlin (2011)
18. Masnou, S., Morel, J.M.: Level lines based disocclusion. In: Proc. of the International Conference on Image Processing. vol. 3, pp. 259–263. IEEE (1998)
19. Murthagh, B.A., Saunders, M.A.: A projected lagrangian algorithm and its implementation for sparse nonlinear constraints. Mathematical Programming Study 16, 84–117 (1982)
20. Pock, T., Schoenemann, T., Graber, G., Bischof, H., Cremers, D.: A convex formulation of continuous multi-label problems. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) Computer Vision - ECCV 2008. LNCS, vol. 5304, pp. 792–805. Springer (2008)
21. Robinson, S.M.: A quadratically-convergent algorithm for general nonlinear programming problems. Mathematical Programming 3, 145–156 (1972)
22. Schmaltz, C., Weickert, J., Bruhn, A.: Beating the quality of JPEG 2000 with anisotropic diffusion. In: Denzler, J., Notni, G., Süße, H. (eds.) Pattern Recognition, LNCS, vol. 5748, pp. 452–461. Springer, Berlin (2009)
23. Sokolowski, J., Zolesio, J.P.: Introduction to Shape Optimization. Springer (1992)
24. Stadler, G.: Elliptic optimal control problems with L_1-control cost and applications for the placement of control devices. Computational Optimization and Applications 44(2), 159–181 (2009)
25. Tröltzsch, F.: Optimale Steuerung Partieller Differentialgleichungen: Theorie, Verfahren und Anwendungen. Vieweg+Teubner, 2nd. edn. (2009)
26. Wachsmuth, G., Wachsmuth, D.: Convergence and regularization results for optimal control problems with sparsity functional. ESAIM: Control, Optimisation and Calculus of Variations 17(3), 858–886 (2011)
27. Xu, Y., Yin, W.: A block coordinate descent method for multi-convex optimization with applications to nonnegative tensor factorization and completion. Rice CAAM Technical Report TR12-15, Rice University (2012)
28. Xu, Y., Yin, W., Wen, Z., Zhang, Y.: An alternating direction algorithm for matrix completion with nonnegative factors. Frontiers of Mathematics in China 7(2), 365–384 (2012)