In X.-C. Tai, E. Bae, T. F. Chan, M. Lysaker (Eds.): Energy Minimization Methods in
Computer Vision and Pattern Recognition. Lecture Notes in Computer Science,
Springer, Vol. 8932, 85-98, Berlin, 2015. The final publication is available at
link.springer.com.

Why Does Non-binary Mask Optimisation Work for Diffusion-based Image Compression?

Laurent Hoeltgen and Joachim Weickert

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

Abstract Finding optimal data for inpainting is a key problem for image-compression with partial differential equations. Not only the location of important pixels but also their values should be optimal to maximise the quality gain. The position of important data is usually encoded in a binary mask. Recent studies have shown that allowing non-binary masks may lead to tremendous speedups but comes at the expense of higher storage costs and yields prohibitive memory requirements for the design of competitive image compression codecs. We show that a recently suggested heuristic to eliminate the additional storage costs of the non-binary mask has a strong theoretical foundation in finite dimension. Binary and non-binary masks are equivalent in the sense that they can both give the same reconstruction error if the binary mask is supplemented with optimal data which does not increase the memory footprint. Further, we suggest two fast numerical schemes to obtain this optimised data. This provides a significant building block in the conception of efficient data compression schemes with partial differential equations.

Keywords: Laplace Interpolation, Inpainting, Convex Optimisation

1 Introduction

A major challenge in data analysis is the reconstruction of a function, for example a 1D signal or an image, from a few data points. In image processing this interpolation problem is called inpainting [?,?]. Often 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 data used for the reconstruction. For instance, in recent approaches related to image compression [?,?,?,?,?,?,?,?,?,?] the authors selected suitable interpolation data for reconstructions via partial differential equations (PDEs). Köstler et al. demonstrated in [?] that PDEs can also be used to compress video sequences. Let us emphasise that finding good data sets for interpolation is by no means a simple task. Choosing for example 5% of the pixels from a 256×256 pixel large image offers more than 10^{5000} possible combinations.

Besides a good selection for the position of the interpolation data, one can also consider an optimisation of corresponding data values in the co-domain. Schmaltz et al. [?] used direct searching strategies to find good tonal values for the reconstruction of their nonlinear diffusion process. Mainberger et al. [?] presented a solid mathematical foundation of tonal optimisation and emphasised the benefits of a good spatial and tonal data selection. Since their inpainting was based on the Laplace equation, the optimal grey values could be found by solving a least squares approach. A related optimal control based model to find good inpainting masks was considered by Hoeltgen et al. [?]. This model, however, uses a regularised formulation that does not require the mask to be binary. It reduces an unfeasible combinatorial problem to a series of convex optimisation problems that can be solved in an highly efficient way. Similar models were also discussed in [?] by Chen et al., whereas Ochs et al. suggested fast numerics in [?,?]. The approaches of Mainberger et al. [?], Hoeltgen et al. [?], and Chen et al. [?] achieve a similar high level of reconstruction quality. The benefits of the control based approach of Hoeltgen et al. [?] over the Mainberger method [?] is its significantly lower runtime. Unfortunately, storing non-binary masks is expensive in terms of memory requirements, especially in the context of image compression. As a remedy, Hoeltgen et al. [?] suggested a heuristic to reduce the storage requirements. They proposed to binarise the mask and to apply the tonal optimisation of Mainberger et al. [?] as a postprocessing step. Interestingly this heuristic yielded a intriguing phenomenon: *The error with optimal mask values and original data were almost identical to the errors with binary masks and optimised grey values.*

Our Contribution. The goal of our paper is to show that the similarity in the error measures discovered in [?] is no coincidence. We show that in a finite dimensional setting the reconstruction error with an optimal non-binary mask and original image data is always identical to the error with a binary mask combined with tonal optimisation. Thus, we provide a mathematically sound foundation for the development of a image compression codec based on the Laplace equation. Furthermore, we also propose two highly efficient algorithms to handle the latter tonal value optimisation on the CPU and the GPU.

Structure of the Paper. Our paper is organised as follows. In Section 2 we briefly introduce the underlying inpainting scheme as well as the related optimisation tasks that will be analysed in this paper. Section 3 shows the main result of this work, namely the equivalence between the optimisation problems from the first section. Next, Section 4 demonstrates two new numerical schemes that allow a fast and efficient optimisation on both the CPU and GPU. Finally, the paper is closed in Section 5 with a summary and an outlook on future challenges.

2 Inpainting with Homogeneous Diffusion

Inpainting with homogeneous diffusion (sometimes also called Laplace interpolation) is a rather simple reconstruction method that is well suited for highly scattered data in arbitrary dimensional settings. It can be modelled as follows. Let $f : \Omega \to \mathbb{R}$ be a smooth function on some bounded domain $\Omega \subset \mathbb{R}^n$ with a sufficiently regular boundary $\partial\Omega$. Throughout this work, we will restrict ourselves to the case $n = 2$ (grey value images) even though many of the results hold for arbitrary $n \geqslant 1$. Moreover, let us assume that there exists a closed nonempty set of known data $\Omega_K \subsetneqq \Omega$ that will be interpolated by the underlying diffusion process. Homogeneous diffusion inpainting considers the following partial differential equation with mixed boundary conditions.

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

where $\partial_n u$ denotes the derivative of u in the outer normal direction. We assume that both boundary sets $\partial\Omega_K$ and $\partial\Omega \setminus \partial\Omega_K$ are non-empty. Equations of this type are commonly referred to as mixed boundary value problems and sometimes also as Zaremba's problem named after Stanislaw Zaremba who studied such equations already in 1910 [**?**]. The existence and uniqueness of solutions has been extensively studied during the last century. Showing that (1) is indeed solvable is by no means a trivial feat. We refer to [**?**] for an extensive study of linear elliptic partial differential equations. A particularly easy case is the 1-D setting, where the solution can obviously be expressed using piecewise linear splines interpolating data on $\partial\Omega_K$.

Following [**?**], we introduce the *confidence function* $c \colon \Omega \to \mathbb{R}$ which states whether a point is known or not. It is defined by

$$c\left(\boldsymbol{x}\right) := \begin{cases} 1 & \text{for } \boldsymbol{x} \in \Omega_K, \\ 0 & \text{for } \boldsymbol{x} \in \Omega \setminus \Omega_K \ . \end{cases} \tag{2}$$

The confidence function lets us rewrite (1) as a more compact functional equation of the form

$$\begin{aligned} c\left(\boldsymbol{x}\right)\left(u\left(\boldsymbol{x}\right) - f\left(\boldsymbol{x}\right)\right) - \left(1 - c\left(\boldsymbol{x}\right)\right)\Delta u\left(\boldsymbol{x}\right) &= 0, \quad \text{on } \Omega \\ \partial_n u\left(\boldsymbol{x}\right) &= 0, \quad \text{on } \partial\Omega \setminus \partial\Omega_K \ . \end{aligned} \tag{3}$$

As shown in [**?**,**?**], the choice of c has a substantial influence on the solution. For most parts of this text we will prefer the formulation (3), as it is more comfortable to work with. Further, this formulation also makes sense when c is not binary-valued but takes arbitrary values. This observation was also exploited in [?] where the authors complemented (3) by a convex energy to obtain a sparse set of optimal values for c.

A discrete framework corresponding to (3) is easily obtained by a straightforward discretisation of the functions c, u and f on a regular grid of size $n_1 \times n_2$ and by placing the corresponding entries in vectors $\boldsymbol{c}$, $\boldsymbol{u}$ and $\boldsymbol{f}$ respectively. If $\boldsymbol{A}$ represents the symmetric $N \times N$ matrix (N being the total numbers of pixels on our grid, e.g. $N = n_1 n_2$) of the discrete Laplace operator $\mathit{\Delta}$ with homogeneous Neumann boundary conditions on $\partial\mathit{\Omega} \setminus \partial\mathit{\Omega}_K$ we obtain

$$\operatorname{diag}(\boldsymbol{c})(\boldsymbol{u} - \boldsymbol{f}) + (\boldsymbol{I} - \operatorname{diag}(\boldsymbol{c}))(-\boldsymbol{A})\,\boldsymbol{u} = \boldsymbol{0} \tag{4}$$

where $\boldsymbol{I}$ is the identity matrix, $\operatorname{diag}(\boldsymbol{c})$ is a diagonal matrix with the sampled values from $\boldsymbol{c}$ as its entries on the main diagonal. By a simple reordering of the terms, (4) can be rewritten as the following linear system,

$$\Big(\operatorname{diag}(\boldsymbol{c}) + \Big(\boldsymbol{I} - \operatorname{diag}(\boldsymbol{c})\Big)(-\boldsymbol{A})\Big)\boldsymbol{u} = \operatorname{diag}(\boldsymbol{c})\,\boldsymbol{f} \tag{5}$$

If the vector $\boldsymbol{c}$ contains as its entries only the values 0 or 1 and if it is not the zero vector, then it has been shown in [?] that this linear system of equations has a unique solution and that it can be solved efficiently by using bidirectional multigrid methods. Further, Mainberger et al. demonstrated in [?] that a careful tuning of the data $\boldsymbol{f}$ can lead to large quality gains in the reconstruction, e.g. one seeks data $\boldsymbol{g}$ such that solutions of

$$\Big(\operatorname{diag}(\boldsymbol{c}) + \Big(\boldsymbol{I} - \operatorname{diag}(\boldsymbol{c})\Big)(-\boldsymbol{A})\Big)\boldsymbol{u} = \operatorname{diag}(\boldsymbol{c})\,\boldsymbol{g} \tag{6}$$

are as close to our desired output $\boldsymbol{f}$ as possible. Related investigations can also be found in [?], where the authors present subdivision strategies that exploit non-linear PDEs. If the underlying diffusion process is based on a linear operator, then the optimisation can be formulated as a linear least squares problem by considering

$$\boldsymbol{g} = \operatorname*{arg\,min}_{\boldsymbol{x} \in \mathbb{R}^N} \left\{ \frac{1}{2} \left\| \Big(\operatorname{diag}(\boldsymbol{c}) + \Big(\boldsymbol{I} - \operatorname{diag}(\boldsymbol{c})\Big)(-\boldsymbol{A})\Big)^{-1} \operatorname{diag}(\boldsymbol{c})\,\boldsymbol{x} - \boldsymbol{f} \right\|_2^2 \right\} \tag{7}$$

We refer to [?] for the original presentation of this model. In the context of nonlinear diffusion it is not possible to consider such a convex optimisation problem. Schmaltz et al. suggested in [?] to use clever searching strategies in this case.

To alleviate the upcoming discussion we introduce two definitions related to the just mentioned linear system needed for the reconstruction and the least squares problem required for the optimisation. We call *inpainting matrix* the following $N \times N$ matrix

$$\boldsymbol{B}(\boldsymbol{c}) := \operatorname{diag}(\boldsymbol{c}) + (\boldsymbol{I} - \operatorname{diag}(\boldsymbol{c}))(-\boldsymbol{A}) \quad .$$

Further, if we have a mask $\boldsymbol{c}$ to our avail for which the inpainting matrix is invertible, then we call the following $N \times N$ matrix *reconstruction matrix*

$$\boldsymbol{M}(\boldsymbol{c}) := \boldsymbol{B}^{-1}(\boldsymbol{c})\,\mathrm{diag}(\boldsymbol{c}) \quad .$$

The exact requirements for the existence of $\boldsymbol{B}^{-1}(\boldsymbol{c})$ will be covered in future work. For the moment we simply assume that this matrix exists. Using these definitions, we can rewrite the linear system (5) as

$$\boldsymbol{B}\,(\boldsymbol{c})\boldsymbol{u} = \mathrm{diag}(\boldsymbol{c})\boldsymbol{f} \quad \Leftrightarrow \quad \boldsymbol{u} = \boldsymbol{M}(\boldsymbol{c})\boldsymbol{f} \tag{8}$$

and the grey value optimisation problem from (7) takes the form

$$\boldsymbol{g} = \operatorname*{arg\,min}_{\boldsymbol{x}\in\mathbb{R}^N} \left\{ \frac{1}{2} \left\| \boldsymbol{M}(\boldsymbol{c})\,\boldsymbol{x} - \boldsymbol{f} \right\|_2^2 \right\} \tag{9}$$

In order to quantify the quality of the results obtained from the inpainting we introduce the *reconstruction error* which simply measures the ℓ_2 distance between the reconstruction and the initially specified data. We denote it by

$$E\,(\boldsymbol{c}, \boldsymbol{g}) := \frac{1}{2} \left\| \boldsymbol{M}(\boldsymbol{c})\,\boldsymbol{g} - \boldsymbol{f} \right\|_2^2 \tag{10}$$

Note that the reconstruction error is simply a rescaled variant of the popular mean square error frequently used for error measures. We will use the reconstruction error as it is more directly related to the optimisation problem to be analysed in this paper.

3 Optimisation in the Co-Domain

Let us introduce some further notations and definitions relevant for the forthcoming paragraphs. For the sake of simplicity we assume that all N pixels in our image have been labelled by a single index. Thus, the individual pixel locations are given by the set $J := \{1, \ldots, N\}$. Further, we assume that the mask positions have been fixed beforehand and cannot be altered anymore. Also we require that the mask is not empty. We denote the set of mask positions by $K \subseteq J$. Clearly, it follows that $c_i = 0$ for all $i \in J \setminus K$. For $i \in K$ we are left with three possibilities. Either we fix the mask value c_i for all $i \in K$ and manipulate the pixel value g_i to improve the reconstruction, or we fix g_i and optimise the value of c_i. Lastly we could also try to optimise both g_i and c_i for all $i \in K$. In this paper we are interested in the first two special cases. Setting $c_i = 1$ for all $i \in K$ and optimising $\boldsymbol{g}$ yields the tonal optimisation problem described in [**?**]. Fixing $\boldsymbol{g} = \boldsymbol{f}$ and optimising $\boldsymbol{c}$ is related to the strategies from [**?**], even though the approach there did not require the support of $\boldsymbol{c}$ to be specified beforehand. The question arises which of these two frameworks yields the smaller error. Both methods can only influence the reconstruction at locations indicated by the set K. Both optimisation strategies can be reduced to a system of $|K|$ equations although these are only linear if we optimise $\boldsymbol{g}$. In order to analyse these problems let us denote by $\overline{\boldsymbol{c}}$ the following mask:

$$\overline{c}_i := \begin{cases} 1, & i \in K \\ 0, & i \notin K \end{cases} \tag{11}$$

Then we can reformulate the two previously described settings as the following optimisation problems.

$$\boldsymbol{g} = \mathop{\arg\min}_{\boldsymbol{x} \in \mathbb{R}^N} \left\{ E\left(\overline{\boldsymbol{c}}, \boldsymbol{x}\right) \right\} \quad \text{and} \quad \tilde{\boldsymbol{c}} = \mathop{\arg\min}_{c_i, i \in K} \left\{ E\left(\boldsymbol{c}, \boldsymbol{f}\right) \right\} \tag{12}$$

Let us emphasise, that the optimisation is always to be understood as unconstrained. We do not restrict the range of values that the mask or the data takes. The necessary conditions for a minimum of E with respect to $\boldsymbol{g}$ (resp. $\boldsymbol{c}$) are given by

$$\frac{\partial}{\partial g_i} E\left(\overline{\boldsymbol{c}}, \boldsymbol{g}\right) = 0, \quad \forall i \in J \quad \text{resp.} \quad \frac{\partial}{\partial c_i} E\left(\boldsymbol{c}, \boldsymbol{f}\right) = 0, \quad \forall i \in K \tag{13}$$

In order to analyse the potential benefits of optimising the mask values or the grey values we need analytic representations of the gradient of E with respect to each of its variables. To this end we adapt a result from Ochs et al. [**?**] (Lemma 9). There, the authors stated it for the case $\boldsymbol{x} = \boldsymbol{f}$. We refer to the original work for the proof.

Proposition 1 (Gradients of the Reconstruction Error). *The gradients of the reconstruction error with respect to its two arguments are given by*

$$\boldsymbol{\nabla}_{\boldsymbol{c}} E\left(\boldsymbol{c}, \boldsymbol{x}\right) = \operatorname{diag}\big(\boldsymbol{x} - (\boldsymbol{I} + \boldsymbol{A}) \boldsymbol{M}\left(\boldsymbol{c}\right) \boldsymbol{x}\big) \boldsymbol{B}^{-\top}(\boldsymbol{c}) \left(\boldsymbol{M}(\boldsymbol{c})\, \boldsymbol{x} - \boldsymbol{f}\right) \;, \tag{14}$$

$$\boldsymbol{\nabla}_{\boldsymbol{x}} E\left(\boldsymbol{c}, \boldsymbol{x}\right) = \boldsymbol{M}^{\top}(\boldsymbol{c}) \left(\boldsymbol{M}(\boldsymbol{c})\, \boldsymbol{x} - \boldsymbol{f}\right) \;. \tag{15}$$

Note that both gradients of E have a certain similarity. If we denote

$$\boldsymbol{T} := \boldsymbol{B}^{-\top}(\boldsymbol{c}) \left(\boldsymbol{B}^{-1}(\boldsymbol{c}) \operatorname{diag}(\boldsymbol{c})\, \boldsymbol{x} - \boldsymbol{f}\right) \;, \tag{16}$$

then we have

$$\boldsymbol{\nabla}_{\boldsymbol{c}} E\left(\boldsymbol{c}, \boldsymbol{x}\right) = \operatorname{diag}\big(\boldsymbol{x} - \left(\boldsymbol{I} + \boldsymbol{A}\right) \boldsymbol{B}^{-1}\left(\boldsymbol{c}\right) \operatorname{diag}(\boldsymbol{c})\, \boldsymbol{x}\big)\, \boldsymbol{T} \;, \tag{17}$$

$$\boldsymbol{\nabla}_{\boldsymbol{x}} E\left(\boldsymbol{c}, \boldsymbol{x}\right) = \operatorname{diag}(\boldsymbol{c})\, \boldsymbol{T} \;. \tag{18}$$

Assume now that for fixed mask positions K we have found the optimal mask values $\tilde{\boldsymbol{c}}$ for the reconstruction with respect to the original data $\boldsymbol{f}$. This means we have

$$\left(\boldsymbol{\nabla}_{\boldsymbol{c}} E\left(\boldsymbol{c}, \boldsymbol{f}\right)\big|_{\boldsymbol{c}=\tilde{\boldsymbol{c}}}\right)_i = 0 \quad \forall i \in K \;. \tag{19}$$

Inserting the expression from (14) into (19) yields

$$\left(\operatorname{diag}\left(\boldsymbol{f} - \left(\boldsymbol{I} + \boldsymbol{A}\right) \boldsymbol{M}\left(\tilde{\boldsymbol{c}}\right) \boldsymbol{f}\right) \boldsymbol{B}^{-\top}(\tilde{\boldsymbol{c}}) \left(\boldsymbol{M}\left(\tilde{\boldsymbol{c}}\right) \boldsymbol{f} - \boldsymbol{f}\right)\right)_i = 0 \quad \forall i \in K \;. \tag{20}$$

The previous equation is a product between a diagonal matrix and a vector. This comes down to a componentwise multiplication between the diagonal entries of the matrix and the vectors entries. Therefore, at least one of the two following equations must hold:

$$\left(\boldsymbol{f} - (\boldsymbol{I} + \boldsymbol{A})\, \boldsymbol{M}\,(\tilde{\boldsymbol{c}})\, \boldsymbol{f}\right)_{i \in K} = 0 \ , \tag{21}$$

$$\left(\boldsymbol{B}^{-\top}(\tilde{\boldsymbol{c}})\,(\boldsymbol{M}\,(\tilde{\boldsymbol{c}})\,\boldsymbol{f} - \boldsymbol{f})\right)_{i \in K} = 0 \ . \tag{22}$$

Our goal is to show that the second equation actually always holds for all $i \in K$. If for a certain entry $i \in K$, the first equation equation differs from 0, then the second one must be 0. Thus, we only need to show that the first equation can never hold. To his end, note that $\boldsymbol{u} := \boldsymbol{M}(\tilde{\boldsymbol{c}})\,\boldsymbol{f}$ solves by definition the equation

$$\operatorname{diag}(\tilde{\boldsymbol{c}})\,(\boldsymbol{u} - \boldsymbol{f}) - (\boldsymbol{I} - \operatorname{diag}(\tilde{\boldsymbol{c}}))\,\boldsymbol{A}\boldsymbol{u} = \boldsymbol{0} \tag{23}$$

and that (21) is equivalent to

$$\operatorname{diag}(\tilde{\boldsymbol{c}})\,(\boldsymbol{f} - (\boldsymbol{I} + \boldsymbol{A})\,\boldsymbol{M}\,(\tilde{\boldsymbol{c}})\,\boldsymbol{f}) = \boldsymbol{0} \ . \tag{24}$$

From (23) it follows that

$$\operatorname{diag}(\tilde{\boldsymbol{c}})\,(\boldsymbol{u} - \boldsymbol{f} + \boldsymbol{A}\boldsymbol{u}) = \boldsymbol{A}\boldsymbol{u} \ . \tag{25}$$

Plugging (25) into (24) yields the requirement $-\boldsymbol{A}\boldsymbol{u} = \boldsymbol{0}$. Thus, if (21) would hold, then the reconstruction $\boldsymbol{u} = \boldsymbol{M}\,(\tilde{\boldsymbol{c}})\,\boldsymbol{f}$ would also solve $\boldsymbol{A}\boldsymbol{u} = \boldsymbol{0}$. This would contradict our assumption that the inpainting mask $\tilde{\boldsymbol{c}}$ is nonempty. Therefore, (21) can never hold.

Similarly as for (24), we note that (22) can be extended to all indices $i \in K$ by multiplying it from the left with $\operatorname{diag}(\tilde{\boldsymbol{c}})$. This gives us

$$\operatorname{diag}(\tilde{\boldsymbol{c}})\,\boldsymbol{B}^{-\top}(\tilde{\boldsymbol{c}})\,(\boldsymbol{M}\,(\tilde{\boldsymbol{c}})\,\boldsymbol{f} - \boldsymbol{f}) = 0$$

which implies that

$$\boldsymbol{\nabla}_{\boldsymbol{x}} E\,(\tilde{\boldsymbol{c}}, \boldsymbol{x})\big|_{\boldsymbol{x}=\boldsymbol{f}} = \boldsymbol{0} \ . \tag{26}$$

The previous equation implies that if we have found optimal mask values, then all necessary optimality conditions with respect to the mask values and with respect to the data values are fulfilled.

Conversely, we could also set $c_i = 1$ for all $i \in K$ to obtain a mask $\overline{\boldsymbol{c}}$ and optimise the grey values for reconstruction. This yields the requirement

$$\begin{aligned}
& \boldsymbol{\nabla}_{\boldsymbol{x}} E\,(\overline{\boldsymbol{c}}, \boldsymbol{x}) = \boldsymbol{0} \\
\Leftrightarrow \quad & \operatorname{diag}(\overline{\boldsymbol{c}})\,\boldsymbol{B}^{-\top}(\overline{\boldsymbol{c}})\left(\boldsymbol{B}^{-1}(\overline{\boldsymbol{c}})\operatorname{diag}(\overline{\boldsymbol{c}})\,\boldsymbol{x} - \boldsymbol{f}\right) = \boldsymbol{0}, \\
\Leftrightarrow \quad & \left(\boldsymbol{B}^{-\top}(\overline{\boldsymbol{c}})\left(\boldsymbol{B}^{-1}(\overline{\boldsymbol{c}})\operatorname{diag}(\overline{\boldsymbol{c}})\,\boldsymbol{x} - \boldsymbol{f}\right)\right)_i = 0, \quad \forall i \in K
\end{aligned} \tag{27}$$

Assume that we are in possession of optimal data $\boldsymbol{g}$ for given $\overline{\boldsymbol{c}}$ such that (27) holds. In combination with (16), it follows then that we have

$$\left(\boldsymbol{\nabla}_{\boldsymbol{c}} E(\boldsymbol{c}, \boldsymbol{g})\big|_{\boldsymbol{c}=\overline{\boldsymbol{c}}}\right)_i = 0 \quad \forall i \in K$$

Thus, if we have a binary mask to our avail with optimised tonal values, then it follows again that all necessary optimality conditions are fulfilled. We summarise the previous results in the following theorem.

Theorem 1 (Fulfilment of Optimality Conditions). *Non-binary optimisation of the mask values while keeping the grey values fixed at the original data yields a pair of data that fulfils all necessary optimality conditions for minimising the error of the reconstruction. Similarly, fixing a binary sparsity pattern for the inpainting mask and optimising the grey values also returns a pair of data that fulfils all necessary optimality conditions for minimising the error of the reconstruction.*

Ultimately we would like to show that the reconstruction error is the same regardless of whether we optimise the mask $\boldsymbol{c}$ and keep the data fixed or whether we optimise the data and enforce a binary inpainting mask. In order to show this, we need to prove that

$$E\left(\tilde{\boldsymbol{c}}, \boldsymbol{f}\right) = E\left(\overline{\boldsymbol{c}}, \boldsymbol{g}\right) \quad . \tag{28}$$

To this end let an optimal mask $\tilde{\boldsymbol{c}}$, such that $E\left(\tilde{\boldsymbol{c}}, \boldsymbol{f}\right)$ is minimal, be given and assume that there exists a vector $\overline{\boldsymbol{g}}$ such that the reconstruction is the same with the binary mask $\overline{\boldsymbol{c}}$ corresponding to $\tilde{\boldsymbol{c}}$. Thus, we have

$$\boldsymbol{M}(\overline{\boldsymbol{c}})\,\overline{\boldsymbol{g}} = \boldsymbol{M}(\tilde{\boldsymbol{c}})\,\boldsymbol{f} \quad . \tag{29}$$

By applying the definition of $\boldsymbol{M}(\overline{\boldsymbol{c}})$ we obtain the following analytic expression for $\overline{\boldsymbol{g}}$:

$$\overline{\boldsymbol{g}} = \operatorname{diag}(\overline{\boldsymbol{c}})\,\boldsymbol{B}(\overline{\boldsymbol{c}})\,\boldsymbol{M}(\tilde{\boldsymbol{c}})\,\boldsymbol{f} \tag{30}$$

For a given mask $\tilde{\boldsymbol{c}}$ the right-hand side can always be computed provided that $\boldsymbol{B}^{-1}(\tilde{\boldsymbol{c}})$ exists. In order to show that grey value optimisation comes with no loss compared to mask optimisation we have to show that the pair $(\overline{\boldsymbol{c}}, \overline{\boldsymbol{g}})$ from (30) satisfies the normal equations (15). Thus, we have to show that

$$\boldsymbol{M}^{\top}(\overline{\boldsymbol{c}})\left(\boldsymbol{M}\left(\overline{\boldsymbol{c}}\right)\overline{\boldsymbol{g}} - \boldsymbol{f}\right) = \boldsymbol{0} \tag{31}$$

An essential observation in the verification of (31) is that $\overline{\boldsymbol{c}}$ and $\tilde{\boldsymbol{c}}$ have the same sparsity pattern, i.e. $\overline{\boldsymbol{c}}_i = 1 \Leftrightarrow \tilde{\boldsymbol{c}}_i \neq 0$ and $\overline{\boldsymbol{c}}_i = 0 \Leftrightarrow \tilde{\boldsymbol{c}}_i = 0$. This implies that for the kernels we obtain $\ker\left(\operatorname{diag}(\overline{\boldsymbol{c}})\right) = \ker\left(\operatorname{diag}(\tilde{\boldsymbol{c}})\right)$ and thus $\ker\left(\boldsymbol{M}\left(\overline{\boldsymbol{c}}\right)\right) = \ker\left(\boldsymbol{M}\left(\tilde{\boldsymbol{c}}\right)\right)$, too. Further, we note that for any linear operator $\boldsymbol{K}$ from $\mathbb{R}^n$ to $\mathbb{R}^n$ we have $\ker\left(\boldsymbol{K}^{\top}\right) = \operatorname{ran}\left(\boldsymbol{K}\right)^{\perp}$, where ran denotes the range of the operator. Combining this identity with the first isomorphism theorem yields

$$\begin{aligned} \ker\left(\boldsymbol{M}^{\top}\left(\overline{\boldsymbol{c}}\right)\right) &= \left(\operatorname{ran}\left(\boldsymbol{M}\left(\overline{\boldsymbol{c}}\right)\right)\right)^{\perp} \simeq \left(\mathbb{R}^n / \ker\left(\boldsymbol{M}\left(\overline{\boldsymbol{c}}\right)\right)\right)^{\perp} \\ &= \left(\mathbb{R}^n / \ker\left(\boldsymbol{M}\left(\tilde{\boldsymbol{c}}\right)\right)\right)^{\perp} \simeq \left(\operatorname{ran}\left(\boldsymbol{M}\left(\tilde{\boldsymbol{c}}\right)\right)\right)^{\perp} = \ker\left(\boldsymbol{M}^{\top}(\tilde{\boldsymbol{c}})\right) \end{aligned} \tag{32}$$

The importance of this identity will become clear in a moment. By assumption, $\tilde{\boldsymbol{c}}$ was chosen optimal. This implies $\boldsymbol{\nabla}_{\boldsymbol{c}} E\left(\tilde{\boldsymbol{c}}, \boldsymbol{f}\right) = \boldsymbol{0}$. Because of Theorem 1 it follows that $\boldsymbol{\nabla}_{\boldsymbol{x}} E\left(\tilde{\boldsymbol{c}}, \boldsymbol{f}\right) = \boldsymbol{0}$ is also true. Expanding this equation and using (29) gives us

$$\boldsymbol{M}^{\top}\left(\tilde{\boldsymbol{c}}\right)\left(\boldsymbol{M}\left(\overline{\boldsymbol{c}}\right)\overline{\boldsymbol{g}} - \boldsymbol{f}\right) = \boldsymbol{0} \ . \tag{33}$$

Two possibilities exist. Either $\boldsymbol{M}\left(\overline{\boldsymbol{c}}\right)\overline{\boldsymbol{g}} - \boldsymbol{f} = \boldsymbol{0}$ in which case (31) holds trivially, or $\boldsymbol{0} \neq \boldsymbol{M}\left(\overline{\boldsymbol{c}}\right)\overline{\boldsymbol{g}} - \boldsymbol{f} \in \ker\left(\boldsymbol{M}^{\top}\left(\tilde{\boldsymbol{c}}\right)\right)$. From (32) it follows that $\boldsymbol{M}\left(\overline{\boldsymbol{c}}\right)\overline{\boldsymbol{g}} - \boldsymbol{f} \in \ker\left(\boldsymbol{M}^{\top}\left(\overline{\boldsymbol{c}}\right)\right)$ and thus (31) is fulfilled, too. We conclude that our vector $\overline{\boldsymbol{g}}$ contains the optimal grey values for a binary mask. We summarise our findings in the following theorem.

Theorem 2 (Equivalence between Tonal and Spatial Optimisation). *Let $\tilde{\boldsymbol{c}}$ be a solution of*

$$\min_{c_i, i \in K} \left\{E\left(\boldsymbol{c}, \boldsymbol{f}\right)\right\} \tag{34}$$

and assume that $\boldsymbol{B}^{-1}\left(\tilde{\boldsymbol{c}}\right)$ exists. Then the vector $\overline{\boldsymbol{g}}$ given by (30) *solves*

$$\min_{\boldsymbol{x}} \left\{E\left(\overline{\boldsymbol{c}}, \boldsymbol{x}\right)\right\} \tag{35}$$

where $\overline{\boldsymbol{c}}$ is the binary mask corresponding to $\tilde{\boldsymbol{c}}$ and $E\left(\tilde{\boldsymbol{c}}, \boldsymbol{f}\right) = E\left(\overline{\boldsymbol{c}}, \overline{\boldsymbol{g}}\right)$, i.e. the reconstruction error is the same in each case.

We note that the preceding theory also gives us an analytic expression for the optimal grey values in (30) in terms of optimal mask values.

4 Fast and Efficient Tonal Optimisation

In the previous section we have shown that a tonal optimisation comes with no loss compared to non-binary mask optimisation. Nevertheless, finding the best mask values is a tedious non-convex optimisation task whereas the grey value optimisation problem is a convex least squares problem. The latter family of problems is well studied and many highly efficient strategies exist. In this section we present two fast methods that allow an efficient computation of the perfect tonal values without having to resort to (30) and optimal mask values. Let us remark that our cost function $E\left(\boldsymbol{c}, \cdot\right)$ is convex but not strictly convex. Indeed the reconstruction matrix $\boldsymbol{M}\left(\boldsymbol{c}\right)$ is only invertible if $c_i = 1$ for all i. Further, it is easy to see that usually there exist infinitely many minimisers of $E\left(\boldsymbol{c}, \cdot\right)$. If $\boldsymbol{g}$ is a minimiser, than we can arbitrarily change any entry i of $\boldsymbol{g}$ where $c_i = 0$.

In the following we present two strategies. The first one is well suited for implementations on a CPU, whereas the second one exploits the massive parallelism provided by modern GPUs.

4.1 LSQR Approach

The venerable LSQR algorithm [?,?] is a highly efficient method to solve general least squares problems of the form

$$\underset{x \in \mathbb{R}^n}{\arg\min} \left\{\left\|\boldsymbol{K}\boldsymbol{x} - \boldsymbol{b}\right\|_2\right\} \tag{36}$$

with a large, sparse and unsymmetric matrix $\boldsymbol{K}$. The underlying iterative process applies the bidiagonalisation process of Golub and Kahan [?] and decreases the norm of the residual in each step. Although the algorithm generates a sequence of iterates that has the same properties as those from standard conjugate gradient methods it tends to behave much better in numerically ill-posed situations. Further, it is easy to implement, and it only requires the matrix $\boldsymbol{K}$ for computing matrix vector products of the form $\boldsymbol{K}\boldsymbol{u}$ and $\boldsymbol{K}^\top\boldsymbol{v}$ for various vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. In presence of routines capable of computing these products efficiently, it is not even necessary to know the matrix explicitly. This fact makes the algorithm attractive for solving (12). The adaptation is straightforward. It suffices to to set $\boldsymbol{K} = \boldsymbol{M}(\boldsymbol{c})$ in (36). For our setting we have

$$\begin{aligned} \boldsymbol{y} = \boldsymbol{M}(\boldsymbol{c})\,\boldsymbol{x} \quad &\Leftrightarrow \quad \boldsymbol{B}(\boldsymbol{c})\,\boldsymbol{y} = \operatorname{diag}(\boldsymbol{c})\,\boldsymbol{x}\ , \\ \boldsymbol{y} = \boldsymbol{M}^\top(\boldsymbol{c})\,\boldsymbol{x} \quad &\Leftrightarrow \quad \boldsymbol{B}^\top(\boldsymbol{c})\,\boldsymbol{z} = \boldsymbol{x}, \quad \boldsymbol{y} = \operatorname{diag}(\boldsymbol{c})\,\boldsymbol{z}\ . \end{aligned} \tag{37}$$

The linear systems $\boldsymbol{B}(\boldsymbol{c})\,\boldsymbol{y} = \operatorname{diag}(\boldsymbol{c})\,\boldsymbol{x}$ and $\boldsymbol{B}^\top(\boldsymbol{c})\,\boldsymbol{z} = \boldsymbol{x}$ can be solved in a highly efficient manner with either the multigrid methods from [?] or the multifrontal sparse LU decomposition from [?,?,?]. For the sparse LU solver the decomposition of the matrix $\boldsymbol{B}(\boldsymbol{c})$ needs only be done once during the first iteration of the LSQR algorithm. Forthcoming iterations can then be computed at almost no additional cost. This yields an extremely fast strategy. The complete algorithm is depicted in Algorithm 1.

Algorithm 1: Tonal optimisation with the LSQR Algorithm.

```
Input      : Reconstruction matrix $\boldsymbol{M}(\boldsymbol{c})$, data $\boldsymbol{f}$, number of iteration $N$
Output     : Solution of the least squares problem (12) $\boldsymbol{x}_N$
Initialise : $\bar{\boldsymbol{u}}_1 = \boldsymbol{b},\ \ \beta_1 = \|\bar{\boldsymbol{u}}_1\|,\ \ \boldsymbol{u}_1 = \beta_1^{-1}\bar{\boldsymbol{u}}_1,\ \ \bar{\boldsymbol{v}}_1 = \boldsymbol{M}^\top(\boldsymbol{c})\,\boldsymbol{u}_1,\ \ \alpha_1 = \|\bar{\boldsymbol{v}}_1\|,$
             $\boldsymbol{v}_1 = \alpha_1^{-1}\bar{\boldsymbol{v}}_1,\ \ \boldsymbol{w}_1 = \boldsymbol{v}_1,\ \ \boldsymbol{x}_0 = \boldsymbol{0},\ \ \bar{\phi}_1 = \beta_1,\ \ \bar{\rho}_1 = \alpha_1$
for $k$ from 1 to $N$ do
    $\bar{\boldsymbol{u}}_{k+1} = \boldsymbol{M}(\boldsymbol{c})\,\boldsymbol{v}_k - \alpha_k\boldsymbol{u}_k,\ \ \beta_{k+1} = \|\bar{\boldsymbol{u}}_{k+1}\|,\ \ \boldsymbol{u}_{k+1} = \beta_{k+1}^{-1}\bar{\boldsymbol{u}}_{k+1}$
    $\bar{\boldsymbol{v}}_{k+1} = \boldsymbol{M}(\boldsymbol{c})^\top\boldsymbol{u}_{k+1} - \beta_{k+1}\boldsymbol{v}_k,\ \ \alpha_{k+1} = \|\bar{\boldsymbol{v}}_{k+1}\|,\ \ \boldsymbol{v}_{k+1} = \alpha_{k+1}^{-1}\bar{\boldsymbol{v}}_{k+1}$
    $\rho_k = \sqrt{\bar{\rho}_k^2 + \beta_{k+1}^2},\ \ c_k = \bar{\rho}_k/\rho_k,\ \ s_k = \beta_{k+1}/\rho_k$
    $\theta_{k+1} = s_k\alpha_{k+1},\ \ \bar{\rho}_{k+1} = -c_k\alpha_{k+1},\ \ \phi_k = c_k\bar{\phi}_k,\ \ \bar{\phi}_{k+1} = s_k\bar{\phi}_k$
    $\boldsymbol{x}_{k+1} = \boldsymbol{x}_{k-1} + \phi_k/\rho_k\boldsymbol{w}_k$
    $\boldsymbol{w}_{k+1} = \boldsymbol{v}_{k+1} - \theta_{k+1}/\rho_k\boldsymbol{w}_k$
end
```

4.2 Primal Dual Method

Alternatively to the LSQR algorithm, we may also apply primal dual approaches that have enjoyed an increasing popularity in the previous years, especially in the domain of image processing. Starting from (12) we rewrite the optimisation problem by introducing a dummy variable $\boldsymbol{d}$ and enforce that $\boldsymbol{d}$ coincides with our reconstruction $\boldsymbol{M}(\boldsymbol{c})\,\boldsymbol{x}$. Using the indicator function $\iota_{\{\mathbf{0}\}}$ defined as

$$\iota_{\{\mathbf{0}\}}(\boldsymbol{x}) := \begin{cases} 0, & \boldsymbol{x} = \mathbf{0} \\ \infty, & \boldsymbol{x} \neq \mathbf{0} \end{cases} \tag{38}$$

we can reformulate our task in the following way:

$$\operatorname*{arg\,min}_{\boldsymbol{x},\boldsymbol{d} \in \mathbb{R}^N} \left\{ \frac{1}{2} \left\| \boldsymbol{d} - \boldsymbol{f} \right\|_2^2 + \iota_{\{\mathbf{0}\}} \left(\boldsymbol{d} - \boldsymbol{M}(\boldsymbol{c})\,\boldsymbol{x} \right) \right\} \ . \tag{39}$$

Note that $\boldsymbol{d} = \boldsymbol{M}(\boldsymbol{c})\,\boldsymbol{x}$ if and only if $\boldsymbol{B}(\boldsymbol{c})\,\boldsymbol{d} = \operatorname{diag}(\boldsymbol{c})\,\boldsymbol{x}$. Thus, (39) is equivalent to

$$\operatorname*{arg\,min}_{\boldsymbol{x},\boldsymbol{d} \in \mathbb{R}^N} \left\{ \frac{1}{2} \left\| \boldsymbol{d} - \boldsymbol{f} \right\|_2^2 + \iota_{\{\mathbf{0}\}} \left(\boldsymbol{B}(\boldsymbol{c})\,\boldsymbol{d} - \operatorname{diag}(\boldsymbol{c})\,\boldsymbol{x} \right) \right\} \ . \tag{40}$$

The benefit of the latter equation is that we have eliminated the inverse $\boldsymbol{B}^{-1}(\boldsymbol{c})$ by introducing $\boldsymbol{B}(\boldsymbol{c})$ at another position. Equation (40) can be efficiently handled with the algorithm presented in [?]. Applying the primal dual method from [?] only requires the evaluation of $\boldsymbol{B}(\boldsymbol{c})\,\boldsymbol{u}$ and $\boldsymbol{B}^\top(\boldsymbol{c})\,\boldsymbol{v}$ for vectors $\boldsymbol{u}$ and $\boldsymbol{v}$. Since the matrix $\boldsymbol{B}(\boldsymbol{c})$ is structured and extremely sparse, these computations can be handled in an efficient manner, leading to a high performing grey value optimisation strategy. A straightforward application of Algorithm 1 from [?] with $G(\boldsymbol{x}) := \frac{1}{2}\left\|\boldsymbol{x} - \boldsymbol{f}\right\|^2$ and $F(\boldsymbol{x}) = \iota_{\{\mathbf{0}\}}(\boldsymbol{x})$ gives us the simple iterative strategy shown in Algorithm 2.

Algorithm 2: Tonal optimisation with primal dual methods.

```
Input      : $N$ the number of iterations.
Output     : Vectors $\boldsymbol{x}^{N+1}$ and $\boldsymbol{d}^{N+1}$ solving (40)
Initialise : $\tau,\ \sigma > 0$ such that $\sigma\tau \left\| \left(\boldsymbol{B}(\boldsymbol{c}) \quad - \operatorname{diag}(\boldsymbol{c})\right) \right\|_2^2 < 1,\ \ \theta \in [0,1]$,
             $\boldsymbol{u}^0,\ \boldsymbol{c}^0,\ \boldsymbol{y}^0$ arbitrary, $\hat{\boldsymbol{u}}^0 = \boldsymbol{u}^0$ and $\hat{\boldsymbol{c}}^0 = \boldsymbol{c}^0$
for $k$ from $1$ to $N$ do
    $\boldsymbol{y}^{k+1} = \boldsymbol{y}^k + \sigma\left(\boldsymbol{B}(\boldsymbol{c})\,\hat{\boldsymbol{d}}^k - \operatorname{diag}(\boldsymbol{c})\,\hat{\boldsymbol{x}}^k\right)$
    $\boldsymbol{d}^{k+1} = (1+\tau)^{-1}\left(\boldsymbol{d}^k - \tau\left(\boldsymbol{B}(\boldsymbol{c})^\top \boldsymbol{y}^{k+1} - f\right)\right)$
    $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \tau \operatorname{diag}(\boldsymbol{c})\,\boldsymbol{y}^{k+1}$
    $\hat{\boldsymbol{d}}^{k+1} = \boldsymbol{d}^{k+1} + \theta\left(\boldsymbol{d}^{k+1} - \boldsymbol{d}^k\right)$
    $\hat{\boldsymbol{x}}^{k+1} = \boldsymbol{x}^{k+1} + \theta\left(\boldsymbol{x}^{k+1} - \boldsymbol{x}^k\right)$
end
```

This algorithm is better suited for parallel implementations than Algorithm 1 since almost all operations are pointwise and do not depend on each other. Further it does not have to solve any linear systems of equations. Let us also remark that additional optimisations like preconditioning strategies, as presented in [?], could further improve the performance of Algorithm 2.

Table 1. Speed comparison between the different algorithms for tonal optimisation on the CPU and GPU. The approach of Mainberger et al. from [?] performs worst on every image size and its runtime increases much faster for larger images than for the other two algorithms. The LSQR approach has the best runtimes on the CPU whereas the primal dual method excels on the GPU. The runtime for computing the mask positions is not included as it is the same for every method.

Image Size	Runtime CPU (seconds)			Runtime GPU
	Method from [?]	Algorithm 1	Algorithm 2	Algorithm 2
32 × 32	7.99	0.44	1.37	1.04
48 × 48	32.57	1.23	2.90	1.35
64 × 64	156.33	2.69	5.82	1.28
80 × 80	360.42	4.63	8.50	1.47
96 × 96	783.87	7.72	14.89	2.30
112 × 112	1633.82	12.02	35.86	2.60
128 × 128	3116.70	18.73	52.57	3.33
256 × 256	95832.64	113.07	260.26	9.02

4.3 Performance Comparison

We compare the performance with respect to speed of our LSQR solver, the primal dual solver and the stochastic tonal optimisation method from [?]. The algorithms were implemented in Fortran and C and all the tests were done on a standard desktop PC with an Intel Xeon processor (3.2GHz) and 24GB of memory. We also used a Nvidia GeForce GTX 460 for the GPU experiments. The runtimes are depicted in Table 1. The represented timings are the averages of three runs for each test case. We used different sizes of the *trui* test image (see Figure 1). Due to spatial constraints we only give results for a single image. The performance for other images are analogous. For each image size we computed a binary inpainting mask using the optimal control framework from [?]. All masks have a density within the range of $5.0 \pm 0.1\%$. We used the algorithm from [?] as a reference method and compared how our algorithms perform in terms of speed. All algorithms converged towards the same solution. The method from [?] uses a multigrid solver for the computation of the inpainting echos. It stopped when the error between two iterates dropped below 10^{-3}. Algorithm 1 stopped when the increment in the solution dropped in norm below 10^{-10} whereas Algorithm 2 halted its execution when the update in any variable was smaller than 10^{-15} in norm. These tolerances were chosen such that the resulting images always had the same reconstruction error. The exceptional performance of the LSQR algorithm stems from the fact that it reached a convergent state within 10 to 30 iterations which implies that it requires less than 100 inpaintings, whereas the method from [?] has to compute an inpainting for every mask pixel during each iteration. While Algorithm 1 is well suited for CPU implementations, the fact that most of the computations in Algorithm 2 can be done in parallel and that no linear systems must be solved render this algorithm attractive for GPUs.

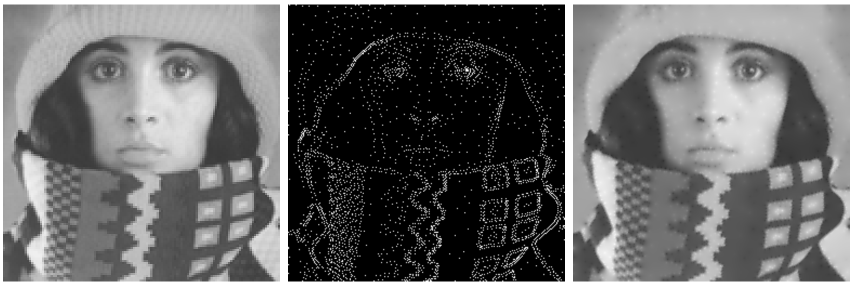

Figure 1. Data used for the experimental setup with a corresponding reconstruction. Left: original (256 × 256), Center: binary mask, Right: reconstruction after tonal optimisation.

5 Summary and Conclusions

We have shown an equivalence result for inpainting with the Laplace equation when the data positions are fixed: Grey value optimisation with binary masks is equivalent to non-binary mask optimisation. This finding justifies the postprocessing step proposed in [**?**] where the optimal mask values were exchanged with optimal data values. Our results show that this strategy comes with no loss in the reconstruction quality. Further, it significantly reduces the amount of data to be stored for compression purposes and marks a significant step towards a fast PDE based data compression codec. Finally, we have suggested two efficient algorithms to solve the tonal optimisation problem on the CPU and on the GPU.

It remains an open question whether a combined and simultaneous optimisation of the mask and the interpolation data can yield an even better reconstruction. The analysis of this problem as well as the development of a competitive image compression codec will be the subject of future work.