Universität des Saarlandes

Fachrichtung 6.1 – Mathematik

Preprint Nr. 373

Evaluating the True Potential of Diffusion-Based Inpainting in a Compression Context

Pascal Peter, Sebastian Hoffmann, Frank Nedwed,
Laurent Hoeltgen and Joachim Weickert

Saarbrücken 2016

Fachrichtung 6.1 – Mathematik
Universität des Saarlandes

Preprint No. 373
submitted: May 31, 2016

Evaluating the True Potential of Diffusion-Based Inpainting in a Compression Context

Pascal Peter
Mathematical Image Analysis Group
Faculty of Mathematics and Computer Science, Campus, E1.7
Saarland University, 66041 Saarbrücken, Germany
peter@mia.uni-saarland.de

Sebastian Hoffmann
Mathematical Image Analysis Group
Faculty of Mathematics and Computer Science, Campus, E1.7
Saarland University, 66041 Saarbrücken, Germany
hoffmann@mia.uni-saarland.de

Frank Nedwed
Mathematical Image Analysis Group
Faculty of Mathematics and Computer Science, Campus, E1.7
Saarland University, 66041 Saarbrücken, Germany
nedwed@mia.uni-saarland.de

Laurent Hoeltgen
Applied Mathematics and Computer Vision Group, Brandenburg University of Technology, Platz der Deutschen Einheit 1, 03046 Cottbus, Germany
hoeltgen@b-tu.de

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

Edited by
FR 6.1 – Mathematik
Universität des Saarlandes
Postfach 15 11 50
66041 Saarbrücken
Germany

Fax: + 49 681 302 4443
e-Mail: preprint@math.uni-sb.de
WWW: http://www.math.uni-sb.de/

Abstract

Partial differential equations (PDEs) are able to reconstruct images accurately from a small fraction of their image points. These inpainting capabilities allow compression codecs with sophisticated anisotropic PDEs to compete with transform-based methods like JPEG 2000. For simple linear PDEs, optimal known data can be found with powerful optimisation strategies. However, the potential of these linear methods for compression has so far not yet been determined.

As a remedy, we present a compression framework with homogeneous, biharmonic, and edge-enhancing diffusion (EED) that supports different strategies for data selection and storage: On the one hand, we find exact masks with optimal control or stochastic sparsification and store them with a combination of PAQ and block coding. On the other hand, we propose a new probabilistic strategy for the selection of suboptimal known data that can be efficiently stored with binary trees and entropy coding.

This new framework allows us a detailed analysis of the strengths and weaknesses of the three PDEs. Our investigation leads to surprising results: At low compression rates, the simple harmonic diffusion can surpass its more sophisticated PDE-based competitors and even JPEG2000. For high compression rates, we find that EED yields the best result due to its robust inpainting performance under suboptimal conditions.

1 Introduction

Image compression with partial differential equations (PDEs) exploits the fact that diffusion processes can give very faithful reconstructions of images from a small amount of image points. In such an inpainting process, the respective PDE describes the propagation of known data to missing image areas. Each successful PDE-based codec has to address three key elements of PDE-based compression: choosing the right PDE for image reconstruction, selecting known data, and storing this data efficiently.

Both the position and the value of the known image points can have a large influence of the reconstruction quality. For homogeneous and biharmonic inpainting, powerful optimisation strategies for the selection of known data exist [1–4]. Even these simple PDEs can reach impressive inpainting results if these methods are applied. However, the specific requirements of compression lead to a trade-off situation: Optimal known data is in most cases expensive to store, while suboptimal data that can be significantly cheaper. In codec

design, the right balance between reconstruction quality and storage cost has to be found.
The current state of the art in compression with PDEs, the R-EED codec by Schmaltz et al. [5], relies on edge-enhancing anisotropic diffusion (EED) [6]. It combines this powerful PDE with known data obtained with a heuristic subdivision strategy. These known points are suboptimal, but can be efficiently stored by a binary tree representation.
In contrast, homogeneous diffusion has so far only been used successfully for specific types of images such as cartoons [7] or depth maps [8–10]. The sophisticated optimisation strategies mentioned above have so far not been used for compression. Instead, semantic approaches like edge detection are the main tool for finding known data in this area. Moreover, there is no codec that relies on biharmonic diffusion at all.
So far, different PDEs have only been compared in a pure inpainting context without considering the need for efficient storage [1–4] or w.r.t. their overall performance in a specific codec [5]. Since no comprehensive comparison that allows different strategies for data selection and efficient storage has been conducted so far, the true potential of these PDE needs yet to be discovered.

Our Contributions. We *evaluate and compare* harmonic, biharmonic and EED inpainting in a compression context. In particular, we analyse how vital compression strategies, namely quantisation and selection of known data, impact the performance of each PDE. For such an evaluation, we need a common compression framework that allows a comparison on equal footing. To this end, we propose two different *new codecs*:

1. *Compression with exact masks* combines unrestricted choice of known image points with efficient entropy encoding. For linear diffusion, we use an optimal control scheme [3, 4] to find the position of known data. Since this approach only works for linear PDEs, we rely on the stochastic sparsification approach of Mainberger et al. [2] for EED instead. For the first time, we embed these powerful data optimisation strategies into a complete compression codec.

2. *Stochastic tree densification* restricts the location of known data to a locally adaptive grid that is represented by a binary tree. This new strategy combines successful tree-based ideas from heuristic algorithms [5, 11] with thorough stochastic optimisation [2, 10]. In post-processing, a *nonlocal node exchange* avoids local minima of this stochastic optimisation.

Our paper extends on preliminary results of a conference paper by the same authors [12]. In the present work, however, we do not restrict ourselves to linear methods, but also integrate anisotropic diffusion into our framework. This also requires the use of different optimisation strategies in both codecs. Moreover, we focus on a detailed experimental analysis of the strengths and weaknesses of the three PDEs. In contrast, the main focus of the previous publication was the competitive performance of linear PDEs compared to transform-based codecs.

Related Work. The codecs presented in this section differ primarily by their inpainting PDE and their strategy for selection of known data. On the one hand, there are *optimisation-driven* methods that carefully optimise the locations of scattered data points to minimise the difference to the original. On the other hand, *feature-driven* codecs store semantic structural components of the image such as edges and try to get the maximum quality out of this given data.

Linear diffusion has so far only be used in feature-driven methods for certain classes of images. Since there is a long history of representing image content purely by edges [13–24], it is not surprising that many PDE-based compression codecs also rely on this feature. This choice can also be motivated by the theoretical results of Belhachmi et al. [1], which suggest to choose known data at locations with large Laplacian magnitude. One obvious advantage over the optimisation-driven encoders is run time: edges can be cheaply detected, e.g. with a Canny edge detector [25] or more sophisticated methods.

Mainberger et al. [7, 26] have shown that simple homogeneous diffusion can beat JPEG 2000 on cartoon-like images. The availability of exact edge data circumvents the main drawback of homogeneous diffusion, its inability to reconstruct sharp contrast changes. There are several codecs that rely on variations of the same core idea and all employ homogeneous diffusion: Carlsson [27] proposed an early sketch-based approach with linear homogeneous diffusion which was later modified and extended by Desai et al. [28]. Wu et al. [29] use JPEG2000 to store known data at thickened edges, while Bastani et al. [30] use source points as given locations. Zhao and Du [31] employ a modified Perona-Malik for presmoothing before edge extraction.

Depth maps are also particularly well-suited for this kind of compression, because this data is naturally composed of piecewise smooth image regions. Both Gautier et al. [8] and Li et al. [9] use edge features and homogeneous diffusion similarly as the approach of Mainberger et al. [26] with specific adaptations to depth maps. Hoffmann et al. [10] go beyond pure storage of edge data and partition the depth map into non-overlapping regions. Between these regions, sharp edges are preserved.

PDE-based codecs for general image content are predominantly optimisation-driven and rely on EED [6]. Initially, this class of methods was proposed by

Galić et al. [11], while the current state of the art is the R-EED codec by Schmaltz et al. [5]. Modifications and extensions of R-EED include colour codecs [32], 3-D data compression [5], and progressive modes [33].
In addition, there are several works that are closely related to compression, but do not consider actual encoding [1–4, 34]. Instead, they deal with optimal reconstruction from small fractions of given data. We directly use results from the optimal control scheme for harmonic PDEs by Hoeltgen et al. [3] and its biharmonic extension. Our densification approach on restricted point sets is inspired by the approach of Mainberger et al. [2, 34]. They consider a stochastic sparsification on unrestricted point sets which has the advantage that it can be applied for any inpainting method without the need for modifications.

Organisation of the Paper. We begin with a review of different optimisation techniques for image inpainting in Section 2. This covers both the selection and representation of locations for known pixels, as well as the optimisation of pixel values. In Section 3 we review different approaches for efficient storage of known data that act as the foundation for our new codecs. These approaches include entropy coding as well as additional preprocessing steps such as prediction. We introduce our two new compression codecs in Section 4. They are specifically designed to allow a practically relevant evaluation of different inpainting PDEs in the compression context. Based on these codecs, we perform a detailed experimental analysis in Section 5. This allows us to evaluate both the overall performance of PDEs in compression, as well as the influence and interaction of individual compression steps. Section 6 concludes our paper with a summary and outlook on future work.

2 Optimising and Representing Known Data

Image Inpainting. First we briefly recapitulate PDE-based image reconstruction. The greyscale image $f : \Omega \to \mathbb{R}$ is known on the *inpainting mask* $K \subset \Omega$, and we want to reconstruct the missing data in the *inpainting domain* $\Omega \setminus K$. The general inpainting equation

$$(1 - c(\boldsymbol{x}))Lu - c(\boldsymbol{x})(u - f) = 0 \tag{1}$$

uses a confidence function $c(\boldsymbol{x}) : \Omega \to \mathbb{R}$ to balance closeness to the original data and the smoothness constraint imposed by a suitable differential operator L. For a binary confidence function that is 1 on K, and 0 on $\Omega \setminus K$, we can use a parabolic PDE-formulation instead: We obtain the missing image parts u as the steady state ($t \to \infty$) of the image evolution that is described by the PDE

$$\partial_t u = Lu \quad \text{ on } \Omega \setminus K. \tag{2}$$

Here, we impose reflecting boundary conditions at the image boundary $\partial\Omega$. In addition, the known data is fixed on K, thus creating Dirichlet boundary conditions $u = f$. In the following, we consider two different parameter-free choices for the differential operator L. In the simplest case, we apply *homogeneous diffusion* [35]:

$$Lu = \Delta u = \operatorname{div}(\boldsymbol{\nabla} u). \tag{3}$$

Since experiments suggest that the *biharmonic operator*

$$Lu = -\Delta^2 u \tag{4}$$

may give better reconstructions [4, 5, 11], it is also considered. Both operators propagate known information equally in all directions and behave consistently throughout the whole image evolution. Finally, we also use edge-enhancing anisotropic diffusion (EED) [6] with

$$Lu = \operatorname{div}(\boldsymbol{D}\boldsymbol{\nabla} u) \tag{5}$$

and an anisotropic tensor $\boldsymbol{D}$ that adapts to the local image structure. The first eigenvector $\boldsymbol{v}_1$ of $\boldsymbol{D}$ is parallel to the gradient and thus points across edges. The second eigenvector $\boldsymbol{v}_2$ is perpendicular to $\boldsymbol{v}_1$ and thereby gives the direction along the edge. We allow full diffusion along edges by setting the eigenvalue $\mu_2 = 1$. Across edges, we want to inhibit diffusion. To this end, we use a Charbonnier diffusivity [36] to define our first eigenvalue:

$$\mu_1 = \left(1 + \frac{|\boldsymbol{\nabla} u_\sigma|^2}{\lambda^2}\right)^{-1}. \tag{6}$$

Here, we use convolution with a Gaussian kernel K_σ with standard deviation σ to obtain a presmoothed image $u_\sigma := K_\sigma * u$. The gradient magnitude $|\boldsymbol{\nabla} u_\sigma|^2$ acts as an edge detector.

In principle, the anisotropic diffusion tensor allows EED to obtain more accurate reconstructions from the same amount of known data [5, 11]. However, the price for this increase in quality are algorithms with higher computational complexity and the need to optimise the contrast parameter λ. Therefore, successful compression codecs like R-EED [5] rely on heuristics to optimise known data. There is a lot of evidence [1–4] that good reconstruction quality depends highly on the known data. For an inpainting mask that contains a fixed fraction of all image points, two optimisation strategies can be applied: *spatial optimisation* selects the location of known data, while *tonal optimisation* chooses the corresponding grey or colour value. In the following we provide a review of optimisation methods that are relevant for our new codecs.

2.1 Spatial Optimisation

Finding optimal positions for a fixed amount of mask points is nontrivial. Let us consider a simple example: We have a discrete image with a resolution of 256 × 256 and want a sparse representation that contains only 5% of the total number of pixels. Combinatorics tell us that there are

$$\binom{65536}{3277} \approx 1.7 \cdot 10^{5648} \tag{7}$$

possibilities to select this amount of known data. This large search space increases both with the resolution of the original image and the percentage of selected pixels. In the following, we review the three most successful approaches to solve this problem in image compression.

Optimal Control. Hoeltgen et al. [3] and Chen et al. [4] optimise the locations of the inpainting mask by solving a constrained optimisation problem of the form

$$\underset{u,\,c}{\operatorname{argmin}} \underbrace{\frac{1}{2}\int_{\Omega} \left((u(\boldsymbol{x}) - f(\boldsymbol{x}))^2 + \delta|c(\boldsymbol{x})| + \varepsilon c(\boldsymbol{x})^2 \right) d\boldsymbol{x}}_{=:E(u,c)}, \tag{8}$$

$$\text{such that } c(\boldsymbol{x})(u(\boldsymbol{x}) - f(\boldsymbol{x})) - (1 - c(\boldsymbol{x}))Lu(\boldsymbol{x}) = 0. \tag{9}$$

Note that at the image boundaries, reflecting boundary conditions still apply. Here the goal is to find simultaneously the reconstruction u and the real-valued *confidence function* $c(\boldsymbol{x})$ by minimising the energy $E(u, c)$. The general inpainting equation (1) acts as a side-constraint. The term $(u(\boldsymbol{x})-f(\boldsymbol{x}))^2$ penalises deviations of the reconstruction u from the original f, while the term $|c(\boldsymbol{x})|$ imposes sparsity of the confidence function. Thereby, the parameter $\delta > 0$ can be used to determine the amount of known data that influences the reconstruction. The other parameter, $\varepsilon > 0$, is fixed to a small positive value, since the existence of a solution for efficient solvers is not guaranteed for $\varepsilon = 0$ (see Hoeltgen et al. [3]).

In fact, minimising the energy $E(u, c)$ is challenging due to two facts: the sparsity term is non-differentiable and the problem is non-convex. Nevertheless, efficient algorithms are possible by considering a series of related linear problems from optimal control theory. It has been shown that fast primal-dual schemes can be employed for finding a solution [3, 4]. While Hoeltgen et al. [3] focus on homogeneous diffusion, biharmonic inpainting is considered in addition by Chen et al. [4].

Note that for compression, a continuous confidence function is disadvantageous, since the coding cost of real numbers is much higher than for the integer case. However, Hoeltgen and Weickert [37] have shown that there is no drawback, if the continuous confidence function is reduced to a binary mask by thresholding. The same reconstruction quality as with a continuous confidence function can be achieved, if one optimises not only the location, but also the value of the known pixels. We discuss such *tonal optimisation strategies* at the end of this section.

Stochastic Optimisation. While optimal control approaches are mathematically well-founded and can be implemented efficiently, they also have two drawbacks: On the one hand, there is no straightforward extension to nonlinear anisotropic diffusion, so far. On the other hand, the parameter δ controls the amount of known data only indirectly. It has to be tuned to achieve a specific density of the inpainting mask.

Instead, one can employ the *stochastic sparsification* approach of Mainberger et al. [2]. It starts with a full mask that contains all image points. From this mask, we remove a fixed percentage α of known data. After inpainting with the smaller mask, we add a fraction β of the removed pixels with the highest reconstruction error back to the mask. This sparsification algorithm iterates the aforementioned steps until the target mask density is reached.

However, there is a substantial risk that this algorithm is caught in a local minimum. To avoid this problem, Mainberger et al. [2] propose a *nonlocal pixel exchange* for post processing: First, they remove n randomly selected points from the mask and reconstruct the image. Afterwards, a candidate set of $m > n$ non-mask pixels is selected randomly. From this candidate set, the algorithm adds the n points that have the largest reconstruction error back to the mask. If the new mask yields a better reconstruction it is kept, otherwise the change is reverted. Optimal control and stochastic sparsification with nonlocal pixel-exchange yield results with comparable reconstruction quality. However, the stochastic approach is usually significantly slower.

Tree-based Subdivision. The R-EED codec in [5] uses another approach to spatial optimisation: A *subdivision* algorithm partitions the original image into rectangular subimages. For each subimage, it selects the corner points and its midpoint as known data. If the local reconstruction error in a subimage exceeds a user-defined error threshold, the image is subdivided,

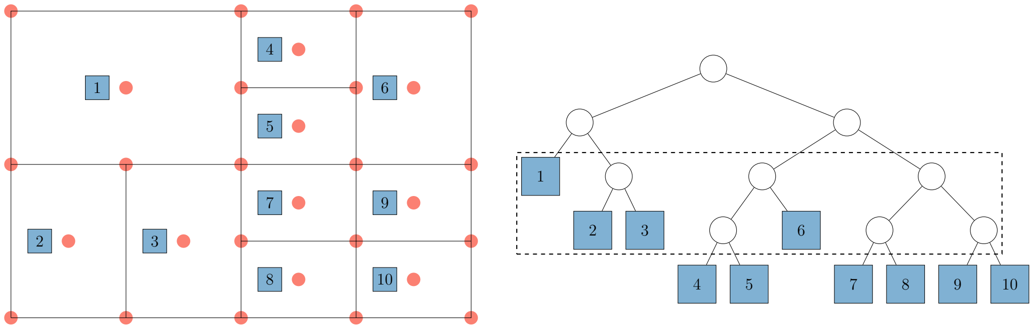

Figure 1: **Tree Representation of Inpainting Masks.** Each square-shaped leaf node in a tree corresponds to a subimage with the same number. Red circles represent the locations stored by each subimage. The subdivision tree has minimum depth 2 and maximum depth 4. Therefore, only the levels 3 and 4 (marked by dashed box) need to be stored: each leaf node as a 0, each inner node as a 1. This yields the overall binary representation 0111001011 with 4 bits for level 3 and 6 bits for level 4.

thus adding more known data in regions that are not reconstructed well. The advantages of this method are twofold: On one hand, it reduces the coding cost of the locations since the subdivision can be efficiently represented by a binary tree. On the other hand, it also reduces the size of the search space. The obvious drawback of *tree-based approaches* is that the restriction to a locally adaptive grid might also reduce the reconstruction quality. In this paper, we also want to investigate how such restrictions affect different inpainting PDEs. We cannot rely on a heuristic subdivision strategy, since this might skew the comparison to *exact masks* that arise from optimal control or stochastic sparsification. Therefore, we propose a new tree-based stochastic approach in Section 4. We discuss the tree representation that we need for this purpose in more detail in the following paragraphs.
For a tree T consisting of nodes $t_0, \ldots, t_n$, the root node t_0 stands for cutting the original image in half in its largest dimension. By adding nodes to the tree, one of the two subimages corresponding to the parent node is split further. In order to encode the tree efficiently in a binary sequence, we exploit that *leaf nodes* are indicators for termination, i.e. the subimage corresponding to a leaf node is not split any further. Thereby, we can represent the tree as a bit sequence (0 for leaf nodes, 1 for other nodes) that results from traversing the tree level by level. We can reduce the coding cost even further by storing the *minimum and maximum tree depth.* All nodes on tree levels up to the minimum depth are split, and all nodes on levels above the maximum tree depth are leaf nodes. Thus, only the tree structure for the levels

in-between these depths needs to be encoded. Fig. 1 provides a visualisation of tree representation. In particular, it also gives a concrete example for a conversion of a tree to a binary sequence.

2.2 Tonal Optimisation

Tonal optimisation is the task of choosing optimal pixel values for an inpainting mask with fixed locations. Intuitively, tonal optimisation can be understood as introducing a small error to the sparse known data to achieve a more significant improvement in the inpainting domain.

In the linear case, tonal optimisation can be formulated as a least squares problem [34]. Let $\boldsymbol{f} \in \mathbb{R}^{n_x n_y}$ denote the original image in vector notation, $\boldsymbol{c} \in \{0,1\}^{n_x n_y}$ the corresponding binary mask, and $\boldsymbol{r}(\boldsymbol{c},\boldsymbol{g})$ the reconstruction that one obtains with linear diffusion inpainting from the mask $\boldsymbol{c}$ and the known data $\boldsymbol{g}$. Then, optimal known data can be found by the minimisation

$$\underset{\boldsymbol{g}}{\operatorname{argmin}}\, |\boldsymbol{f} - \boldsymbol{r}(\boldsymbol{c},\boldsymbol{g})|^2. \tag{10}$$

Hoeltgen et al. [34] have shown that this problem has a unique solution, if the mask is not empty and a linear PDE is used for reconstruction. There are many ways to find this solution. Originally, Mainberger et al. [2] have proposed a randomised Gauß-Seidel scheme that relies on so-called inpainting echoes. Setting a single point of the inpainting mask to 1 and all other mask points to 0 yields the influence of this mask point on the reconstruction, its *echo*. Mainberger et al. [2] have shown that the full inpainting can then be represented as a weighted sum of these echoes, which makes inpainting extremely fast, once these echoes have been computed. However, computing the individual echoes takes time and they are only valid for a specific mask configuration.

More recently, other methods have been proposed that circumvent the costly computation of the echoes. For example, Chen et al. [4] and Hoeltgen and Weickert [37] apply primal-dual methods to solve the problem directly. Hoeltgen et al. [34] propose a gradient descent algorithm that is accelerated with fast explicit diffusion (FED) [38]. In Section 4 we argue that for our specific use in compression, the original echo-based approach is still viable compared to these newer, more efficient methods. Moreover, note that the least squares formulation yields continuous optimal grey values that are just as costly to store as the continuous confidence function from the previous section. We address this problem also in Section 4.

Unfortunately, the aforementioned solvers are not directly applicable to nonlinear anisotropic diffusion. Instead we apply a straightforward iterative algorithm that is also used in R-EED [5]: We visit all mask pixels in random order and check if increasing or decreasing the pixel value by fixed steps yields an improvement. If it does, we keep the new pixel value, otherwise we revert to the original. We iterate these random walks over the whole mask. Similar to the probabilistic algorithm, this method is rather slow, but universally applicable to a wide range of inpainting operators.

3 Storing Data Efficiently

In this section we discuss general requirements and restrictions that the compression setting imposes on optimisation algorithms. This enables us to design codecs that offer a good trade-off between file size and inpainting quality. The most important ingredient for efficient storage of known data in PDE-based compression is entropy coding.

3.1 Entropy Coding

All entropy coders share the common goal of removing redundancy from data. Thereby, they store information losslessly, but with a reduced file size. Huffman coding [39], adaptive arithmetic coding [40], and PAQ [41] have all been successfully used in PDE-based compression [5]. So far, the primary task of these encoders has been to encode the known pixel values. Pixel locations that are represented by a binary tree have only very little potential for further lossless compression. Scheer [42] has shown that even with considerable effort, reductions of the file size are small.

Schmaltz et al. [5] have already conducted an extensive evaluation of the aforementioned entropy coders. They have concluded that arithmetic coding and PAQ offer the best compression results. PAQ is a highly evolved version of prediction by partial matching: It compresses a binary stream by predicting with a very high accuracy if the next bit is a 0 or a 1. To this end, it relies on a large number of complex context models that track how often certain patterns occur in the file. All of these context models are then mixed in a neural network that adapts to the local content of a file during encoding. Such context mixing allows to compress files with varying content very efficiently. The adaptation of context weights is performed by a gradient descent on the coding cost that is computed after a bit is encoded: At this point it becomes clear if the prediction was right or wrong.

In our setting, PAQ appears to be particularly interesting due to its ability to adapt to the local content of a file. If we want to store pixel-accurate locations, this comes down to storing a binary image in addition to the sequence of grey values. In contrast to the tree representation in R-EED, such a binary image contains much more redundancy. Therefore, PAQ can be directly applied as an efficient container format for both positional and brightness data.

3.2 Storing Binary Images

We have tried many different methods that are specialised on the encoding of sparse, binary patterns. In the following we give a short overview of these methods. First, there are off-the-shelf codecs that specialise in binary images.

JBIG uses lossless compression by prediction and arithmetic coding [43]. It considers a fixed configuration of image points that optionally also includes a single pixel with variable position as a context. Depending on the frequency of occurrence of black or white pixels in this context, it predicts the colour of the next pixel. Arithmetic coding is the most efficient entropy coder supported by JBIG.

JBIG2 extends the ideas of JBIG by introducing pattern matching and dictionary approaches [44]. It is specifically tailored to text and halftone images. To this end, it creates a dictionary of repetitive patterns that occur frequently in the image. This dictionary is then used for efficient lossless or lossy compression. For the storage of patterns that do not fit the dictionary, JBIG2 falls back to JBIG encoding.

DjVu is a collection of different compression algorithms that work in tandem [45]: For encoding of mixed content (e.g. images and text), it decomposes images in a binary image foreground and a grey- or colour-valued background part. It compresses the background with wavelet-based compression (IW44 codec) and the foreground with an encoder called JB2. JB2 is a modified version of JBIG2 that relies on the same core concepts.

In PDE-based compression, the aforementioned codecs have been already compared in the context of cartoon compression [26]. Nevertheless, we perform a new evaluation for our compression framework in Section 4, since the nature of our binary images is different. Mainberger et al. [26] compress edge images that feature a lot of connected lines while our optimised masks consist mainly of scattered individual pixels (see e.g. Fig. 2). For the same reason, we also consider different approaches for storing binary images, namely block coding schemes [46–49] and coordinate coding [48]. While we have tried all of the aforementioned approaches, our experiments in Section 4.1 show that

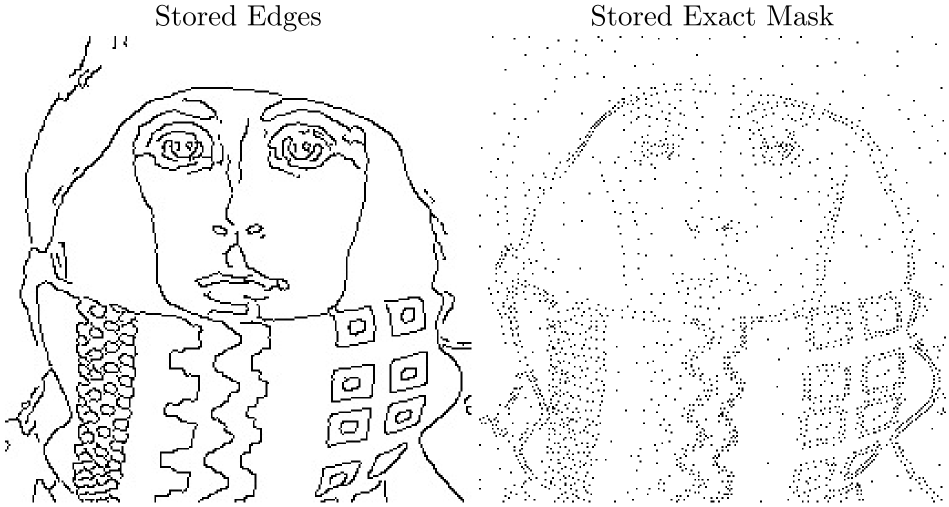

Figure 2: Examples of edges stored in semantic approaches (left, courtesy of Mainberger et al. [26]) and exact masks (right) for the test image *trui*. Since the edge-image contains a lot more connected components and is well-structured, efficient encoding with JBIG is possible. In contrast, the scattered data in the exact mask is more challenging to compress.

ZA Block Coding by Zeng and Ahmed [46] is the best choice for our specific context. This algorithm is specifically designed for sparse binary patterns: It transforms the image into a sequence of binary values by traversing the image row by row. This sequence is then divided into blocks of length b. During encoding, it separates each block by a 0-bit and encodes the relative position of each 1 in the block. The number of bits necessary to encode these positions is given by the block length. Furthermore, each stored coordinate has a leading 1-bit such that the end of block can be detected.

3.3 Quantisation

In addition to the positions of the inpainting mask, we also need to store the pixel values. An important tool for lossy compression of images is the reduction of brightness or colour values from a real-valued domain to a small number of integer values, the so-called *quantisation*. Since we deal with discrete, digital input images, the grey values of the ground truth are usually already quantised. In the following, we assume input images with 8bit grey value depth. Thus, there are initially $2^8 = 256$ grey values $(0, ..., 255)$.

Reducing the number of grey values to some $q \in \mathbb{N}$, $q < 256$ is one of the easiest ways to reduce the file size. It has the added benefit that the human visual system can only distinguish a limited amount of grey values. Therefore, quantisation can be used for *perceptive coding*.

In this paper we perform only *uniform quantisation*. In order to reduce the initial number of $p \in \mathbb{N}$ grey values to a coarser quantisation with only $q < n$ values, we partition the grey value domain in q subintervals of length p/q. This defines a quantisation mapping of the original range $G_p = \{0, \ldots, p-1\}$ to the new grey value range $G_q = \{0, \ldots, q-1\}$: Every value from a given subinterval is mapped to the same grey value from G_q. Obviously, this mapping is lossy. A transformation of a value $x \in G_q$ back to the original dynamic range yields a *reconstruction value* y from the original range G_p according to

$$y := \left\lfloor \frac{xp}{q} + \frac{1}{2} \right\rfloor . \tag{11}$$

Nevertheless, this backtransformation only yields q different values from G_n. Since this introduces an error to the known data, it is a lossy preprocessing step to the lossless entropy coding. In a PDE-based setting, the benefits of tonal optimisation can be diminished, if such a quantisation is applied afterwards. Therefore, in Section 4, we perform tonal optimisation under the constraint of the coarse quantisation.

Note that quantisation has a long tradition in signal processing and many more sophisticated quantisation techniques exist. Non-uniform quantisation allows to distribute quantised values over the full range of original values in such a way that a given error criterion is minimised. These ideas have already been pursued since the dawn of information theory (see e.g. Max [50], Lloyd [51]). For image compression, virtually all lossy methods apply some kind of quantisation. For example, JPEG and JPEG2000 quantise transform coefficients in a non-uniform way. In the case of colour images, one can even consider to quantise vectorial values directly. The drawback of these more complex methods is the need for additional optimisation and potential overhead, since the details of the non-uniform quantisation are needed for decompression. Gersho and Gray [52] provide a detailed overview of both scalar and vector quantisation.

4 Evaluating Inpainting Operators for Compression

Our goal is to evaluate three different diffusion-based inpainting operators with respect to their viability in compression: homogeneous, biharmonic, and edge-enhancing anisotropic inpainting. In particular, we are also interested in how the optimisation methods from Section 2 and the compression steps from Section 3 affect each operator.

Therefore, to assess the true potential of these operators, we have to design codecs that allow them to show their potential in a comparable setting. Note that the compression frameworks which we propose below work in a discrete setting. To this end, we consider the finite difference approximations of the inpainting equation in the same way as Mainberger et al. [2] for linear diffusion and use the standard discretisation for EED as in the R-EED codec [5] .

We propose two different codecs depending on their selection and representation of known data: First we select exact (pixel-accurate) masks with optimal control schemes and stochastic sparsification in Section 4.1. Then we restrict ourselves to locally adaptive grids with an efficient tree representation in Section 4.2.

4.1 Exact Masks with Optimal Control

In the previous sections we have established all building blocks that are necessary to build a codec based on high quality, pixel-accurate masks. In particular, we discuss in the following, how we can store these masks efficiently.

We want our codec to have the following general structure: First, we find an optimal mask that contains a certain percentage of image points. This *mask density* acts as a quality parameter in the range 1 to 100, as in JPEG. We find this optimal mask by the algorithms from Section 2.1: For linear diffusion we use optimal control [3, 4] and for EED we employ stochastic sparsification with nonlocal pixel exchange [2]. This yields a binary image that needs to be stored. Furthermore, we have to decide how to integrate tonal optimisation and quantisation into our codec.

Storing Mask Locations. In order to store the binary image containing the locations of optimal known data efficiently, we have conducted a detailed evaluation of the compression techniques from Section 3.2. We found that block coding schemes [46–49] and coordinate coding [48] are outperformed significantly by encoders for binary images such as JBIG [43], JBIG2 [44]

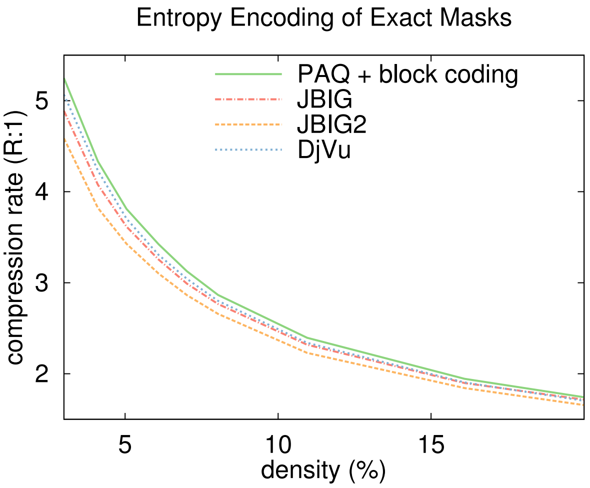

Figure 3: **Encoding of exact masks.** Comparison of different compression methods for exact masks obtained with an optimal control scheme for homogeneous inpainting of the image *peppers*.

and DjVu [45]. However, they are viable as preprocessing steps for entropy coders such as PAQ [41].

Fig. 3 shows compression experiments conducted on exact masks of different density. They were obtained with an optimal control scheme for homogeneous diffusion on the test image *peppers*. On first glance, the results are surprising: JBIG2 performs consistently worse than its predecessor JBIG, even though it features a more sophisticated pattern matching approach. However, one has to keep in mind that JBIG2 is designed for repeating patterns that occur for example in text from a scanned document. In contrast, our masks contain a lot of scattered points that make the creation of a dictionary with repeating patterns difficult.

Similarly, JBIG and DjVu are primarily designed for text documents and natural binary images that contain a lot of connectivity and regular patterns. This explains why the simple block coding scheme by Zeng and Ahmed [46] in combination with PAQ is the best choice. It reduces the file size by up to 10% in comparison to JBIG and DjVu depending on the mask density.

Storing and Optimising Grey Values. In order to store the grey value data associated with the mask efficiently, we have to choose the number q of quantised grey values. As mentioned in Section 3.3, performing such a coarse quantisation after tonal optimisation can affect the results negatively. The grey value optimisation algorithm from R-EED already considers the restriction to a set of coarse grey values. However, the least squares approaches for

```
Input: Original image $\boldsymbol{f}$, admissible set of quantised grey values $Q := \{q_1, ..., q_n\}$,
       inpainting mask $\boldsymbol{c}$.

Initialisation: $\boldsymbol{u} := r(\boldsymbol{c}, \boldsymbol{f})$ and $\boldsymbol{g} := \boldsymbol{f}$.

Compute:
    For all $i \in K$:
        Compute the inpainting echo $\boldsymbol{b_i}$.
    Do
        For all $i \in K$:
          1. Compute the correction term $\alpha := \frac{\boldsymbol{b_i}^\top(\boldsymbol{f}-\boldsymbol{u})}{|\boldsymbol{b_i}|^2}$.
          2. Set $\boldsymbol{u}^{\text{old}} := \boldsymbol{u}$.
          3. Update the grey value $g_i := g_i + \alpha$.
          4. Apply coarse quantisation: $g_i' := \text{argmin}_{q \in Q} |g_i - q|$
          5. Update reconstruction $\boldsymbol{u} := \boldsymbol{u} + \alpha' \cdot \boldsymbol{b_i}$ with $\alpha' = g_i' - u_i$.
    while $|\text{MSE}(\boldsymbol{u}, \boldsymbol{f}) - \text{MSE}(\boldsymbol{u}^{\text{old}}, \boldsymbol{f})| > \varepsilon$.

Output: Optimised quantised grey values $\boldsymbol{g}$.
```

Algorithm 1: Quantisation-aware grey value optimisation.

linear diffusion do not respect this constraint.

As a remedy, we propose *quantisation-aware grey value optimisation* in Algorithm 1. We can express the inpainting solution of the harmonic and biharmonic operator by a superposition of the inpainting echoes [2] from Section 2.2. For a given mask $\boldsymbol{c}$ and corresponding grey values $\boldsymbol{f}$, we denote the associated inpainting result from Section 2 as $\boldsymbol{r}(\boldsymbol{c}, \boldsymbol{f})$. During optimisation, a Gauss-Seidel scheme successively updates the grey values at mask positions one by one. The crucial difference to the tonal optimisation algorithm of Mainberger et al. [2] is that we directly quantise the grey values after every update. Note that the most time-consuming part is the computation of the inpainting echoes. However, since the inpainting mask remains constant, the echoes can be reused for arbitrary quantisation parameters q. Therefore, we are able to optimise q thoroughly and efficiently for the linear operators. For EED, this is more costly.

The choice of q influences the overall file size, since the entropy coding of the grey values becomes more efficient for smaller numbers of different grey values. Therefore, decreasing the parameter q also reduces the file size in general. Simultaneously, the error increases, since the optimised grey values are misrepresented. This negative affect can be somewhat attenuated by the quantisation aware grey value optimisation, but is still present.

Consequently, for a given mask, a suitable parameter q must be found that offers the best trade-off between file size and reconstruction quality. This means that both the inpainting error and the file size have to be minimised simultaneously. For a given quantisation parameter $q \in \{0, ..., 255\}$, let $s : \{0, ..., 255\} \to \mathbb{N}$ be the file size in byte and $e : \{0, ..., 255\} \to \mathbb{R}$ the corresponding mean square error. By normalising both quantities to the range $[0, 1]$ and combining them additively, we define the *trade-off coefficient* μ as

$$\mu := \frac{s(q)}{s(255)} + \frac{e(q)}{e(255)}. \tag{12}$$

The smaller this coefficient, the better the trade-off for a given q. Our goal is to find the best q for a given mask. To this end, we minimise μ with respect to q in combination with quantisation-aware grey value optimisation. This implies a three-step codec for exact masks:

1. Select a fraction d of total pixels with the optimal control approach [3] (harmonic/biharmonic) or stochastic sparsification [2] (EED).

2. Perform quantisation-aware grey value optimisation and select the quantisation parameter q with optimal trade-off between file-size and reconstruction quality.

3. Optimise block size for optimal compression with PAQ. Concatenate header, positional, and grey value data and apply PAQ to the total file.

The reconstruction is straightforward. All the entropy-coded data is recovered and a single inpainting reconstructs the image.

4.2 Stochastic Tree-Building

In this section, we pursue an approach that restricts known data to a regular adaptive grid. In order to lower the coding cost of these locations, we use the *binary subdivision tree* representation from Section 2.1. Unfortunately, there is no straightforward extension of the optimal control approaches for exact masks from the previous section. Moreover, only the heuristic subdivision scheme from R-EED [5] has been used so far to obtain tree-based masks. For a fair comparison, we want to stay as close to the exact codec as possible with respect to the optimisation strategies. Therefore, we extend the stochastic approach of Mainberger et al. [2] to subdivision trees in the following.

```
Input: Original image $\boldsymbol{f}$, fraction $m$ of tree nodes used as candidates for densification, fraction $n$ of candidate nodes that are added in each iteration, desired final mask density $d$.

Initialisation: Splitting tree $T$ containing only the root node $t_0$. Initial leaf node set $L := \{t_1, t_2\}$ containing child nodes of $t_0$.

Compute:
    Do
        1. Compute reconstruction $\boldsymbol{u}$ from mask $C(T)$ and image data $\boldsymbol{f}$.
        2. Choose randomly a candidate set $A \subset L$ containing $m \cdot |L|$ nodes.
        3. For all $t_i \in A$ compute the subimage error $e(t_i)$.
        4. Add a subset of $n \cdot |A|$ candidate nodes $t_i$ with the largest errors $e(t_i)$ to the tree $T$.
        5. Update $L$ to contain all children of leaf nodes from $T$.
    while $|C(T)| < d \cdot |\Omega|$.

Output: Tree $T$ with corresponding mask $C(T)$ of density $d$.
```

Algorithm 2: Stochastic tree densification.

If we want to transfer the basic concepts of stochastic sparsification from Section 2.1 to a binary tree representation, there are some key differences: We have experimentally determined that densification is more efficient for tree structures than sparsification. Therefore, we start with a small amount of data and iteratively add more points at locations with large error until the target density is reached.

In addition, we consider to add nodes to the tree instead of dealing with mask points directly. Since we want to perform a single additional subdivision, the tree structure tells us that only subimages corresponding to leaf nodes may be split. Such a split is equivalent to adding two child nodes to the leaf node (see Fig. 1). Note that several mask points might be added by a single subdivision (the corners and the midpoint of the corresponding subimage). These mask points might also be contained in several of the neighbouring subimages.

Furthermore, the error computation must be adapted. In order to avoid a distortion of the influence of each node, we do not consider the mean square

```
Input: Original image $\boldsymbol{f}$, binary tree $T$, parameters $n < m$.

Compute: Repeat

    1. Create a backup copy $T_{\text{old}}$ of the splitting tree $T$.
    2. Compute reconstruction $\boldsymbol{u}_{\text{old}}$ from mask $C(T_{\text{old}})$ and image data $\boldsymbol{f}$.
    3. Remove the children of $n$ randomly chosen terminal nodes from $T$.
    4. Randomly select a set $A$ containing $m$ leaf nodes from $T$.
    5. For all $t_i \in A$ compute the subimage error $e(t_i)$.
    6. Add the children of the $n$ nodes with the largest error $e(t_i)$ to $T$.
    7. Compute reconstruction $\boldsymbol{u}$ from mask $C(T)$ and image data $\boldsymbol{f}$.
    8. If $\text{MSE}(\boldsymbol{u}, \boldsymbol{f}) > \text{MSE}(\boldsymbol{u}_{\text{old}}, \boldsymbol{f})$
           Reset changes, i.e. $T = T_{\text{old}}$.

  until number of maximum iterations is reached.

Output: Optimised tree $T$.
```

Algorithm 3: Nonlocal node exchange.

error in each subimage, but the sum $e(t_k)$ of unnormalised squared differences

$$e(t_k) = \sum_{(i,j) \in \Omega_k} (f_{i,j} - u_{i,j})^2 \qquad (13)$$

where Ω_k denotes the image domain of the subimage corresponding to the tree node t_k. Without this unnormalised error measure, the same per-pixel-error in small subimages would be weighted higher than in large subimages. Taking all these differences into account, we define *stochastic tree densification* in Algorithm 2. For a target density d, it produces an optimised tree T with a corresponding pixel mask $C(T) \subset \Omega$.

Just as for the original sparsification approach, there is a risk that Algorithm 2 is trapped in a local minimum. To avoid this problem, we propose Algorithm 3, an adapted version of the nonlocal pixel exchange of Mainberger et al. [2] that we have described in Section 2: They first remove random points from the inpainting mask. Then they replace them with potentially better non-mask pixels.

In the following we transfer this concept to our subdivision trees. Most importantly, we have to respect the tree structure in order to define a *nonlocal node exchange*. In the first step, we want to remove n randomly selected

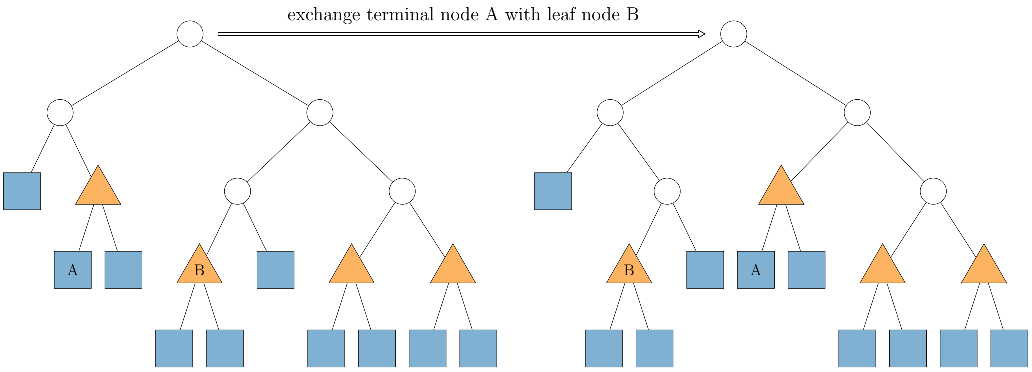

Figure 4: **Nonlocal Node Exchange.** This Figure visualises the concept of exchanging terminal and leaf nodes. Terminal nodes are not split any further, which means that both of their children are leaf nodes. Leaf nodes are marked as blue rectangles, terminal nodes as orange triangles. A single exchange corresponds to reversing the split of a terminal node and splitting a leaf node instead. In our example, we exchange the position of the leaf node A and the terminal node B.

nodes from the tree. However, this set underlies some restrictions: It can only consist of nodes that are split exactly once. This is the case if and only if both children of a node are leaf nodes. We call these nodes *terminal nodes*. The reversion of their associated image split comes down to removing their leaf nodes. Thereby, we convert the terminal node to a leaf node.

In the second step of the nonlocal node exchange, we want to add back nodes to the tree. First, we select a candidate set of m leaf nodes. From these candidates, we select the n nodes that correspond to the subimages with the highest reconstruction error w.r.t. the initial mask. We split these subimages by adding both children to the tree, thus converting leaf nodes into terminal nodes.

These modifications lead to Algorithm 3. The example in Fig. 4 illustrates that the modifications applied to the tree by our algorithm can be interpreted as swapping the positions of pairs consisting of a leaf node and a terminal node.

Finally, the binary trees obtained from the densification and nonlocal node exchange can be stored as a sequence of bits. As in the example from Fig. 1, we store a maximum and minimum tree depth and only save the node-structure explicitly inbetween. The only additional required header data are the image size and the number q of quantised grey values. We combine the tree densification with the same strategies for grey-value optimisation and quantisation as in the previous section and obtain the following four-step compression pipeline:

1. Select a fraction d of total pixels with tree densification (Algorithm 2).
2. Optimise the splitting tree with nonlocal node exchange (Algorithm 3).
3. Perform grey value optimisation and optimise the quantisation parameter w.r.t. the trade-off between file size and reconstruction quality.
4. Concatenate header, positional, and grey value data and apply PAQ.

For reconstruction, we decode the PAQ container, extract positional data from the tree, combine it with the grey values and perform a single inpainting.

5 Experiments

In the following we evaluate the capabilities of harmonic, biharmonic, and EED inpainting for the two compression methods from the previous sections. Our experiments rely on a set of widely used test images. First, we evaluate the sensitivity of the individual operators under different optimisation and compression steps. Then we compare their overall performance to R-EED, which marks the current state of the art in PDE-based compression, and to the transform-based coders JPEG and JPEG2000.

Influence of Data Selection Strategies. In Fig. 6 we compare inpainting results with three different masks that contain 5% known data of the 256×256 image *peppers*: A random mask containing the same uniformly distributed locations for all three algorithms, an exact mask obtained with optimal control or stochastic sparsification, and a restricted mask from tree sparsification (see Fig. 5). The optimised masks are different for each inpainting operator and we do not apply coarse quantisation. For all three methods we have performed tonal optimisation.

EED shows to be far less sensitive to the restriction to an adaptive grid and provides the overall best reconstruction quality for all cases. Biharmonic inpainting performs better than harmonic inpainting in general and also less sensitive to tree-based known data. Also, the results with biharmonic inpainting are more visually pleasing in general: The smoothness constraints of the biharmonic operator avoid the typical singularities that occur at known data for harmonic inpainting. Hoffmann et al. [53] have explained this phenomenon in terms of Green's functions. These singularities are particularly obvious in Fig. 6 (a), the harmonic inpainting with random known data. EED also avoids these singularities and is able to produce the best result consistently. Especially its robust performance on the suboptimal tree-based grid is remarkable. However, it should also be noted that for exact masks, the difference between the three diffusion models is small and would not justify to use the complex nonlinear model instead of the efficient linear ones.

Influence of Mask Density. In order to asses the influence of the mask density on the inpainting results, we optimise exact and restricted masks with different densities, perform grey-value optimisation and compare the mean square error (MSE) at the same mask density. The results in Fig. 7 (a) show that, in general, biharmonic performs better than harmonic inpainting given the same amount of known data. This is consistent with previous results [4]. EED can outperform both methods, but the difference is only significant for low densities. Note, however, that sparse masks are exactly the requirement for efficient compression. The graph also shows again that the restriction of the mask to an adaptive grid has a significant negative impact on the quality. This affects harmonic inpainting more than its biharmonic counterpart and EED.

Influence of Quantisation. The most interesting and surprising results come from a comparison w.r.t. the influence of quantisation. In Fig. 8 we compare results with exact masks and grey value optimisation. This time, we apply a coarse quantisation to 64, 32, and 16 individual grey values. Interestingly, the grey value optimisation is able to compensate for this negative effect very well in the case of harmonic diffusion and EED: The increase of the error is almost negligible compared to the results without quantisation from Fig. 6. However, the higher-order biharmonic inpainting suffers a lot more. As we will see in the following, this affects compression performance significantly.

Compression Performance. An evaluation of the actual compression performance with the codecs from Sections 4.1 and 4.2 in Fig. 7(b) shows a significantly different ranking than in the density comparison. For exact masks, harmonic inpainting can even surpass its biharmonic counterpart. The coding cost for the known data is similar in both cases, but since harmonic inpainting is less sensitive to a coarse quantisation of the grey values, it performs overall better than biharmonic inpainting. The drawbacks of the restrictions in the tree-based approach are attenuated by the reduced positional coding cost. After a break-even point around ratio 20:1, the biharmonic tree-based method outperforms both exact approaches. Since EED does not have distinct advantages at high mask densities, it does not outperform the linear methods at low compression rates. At high compression

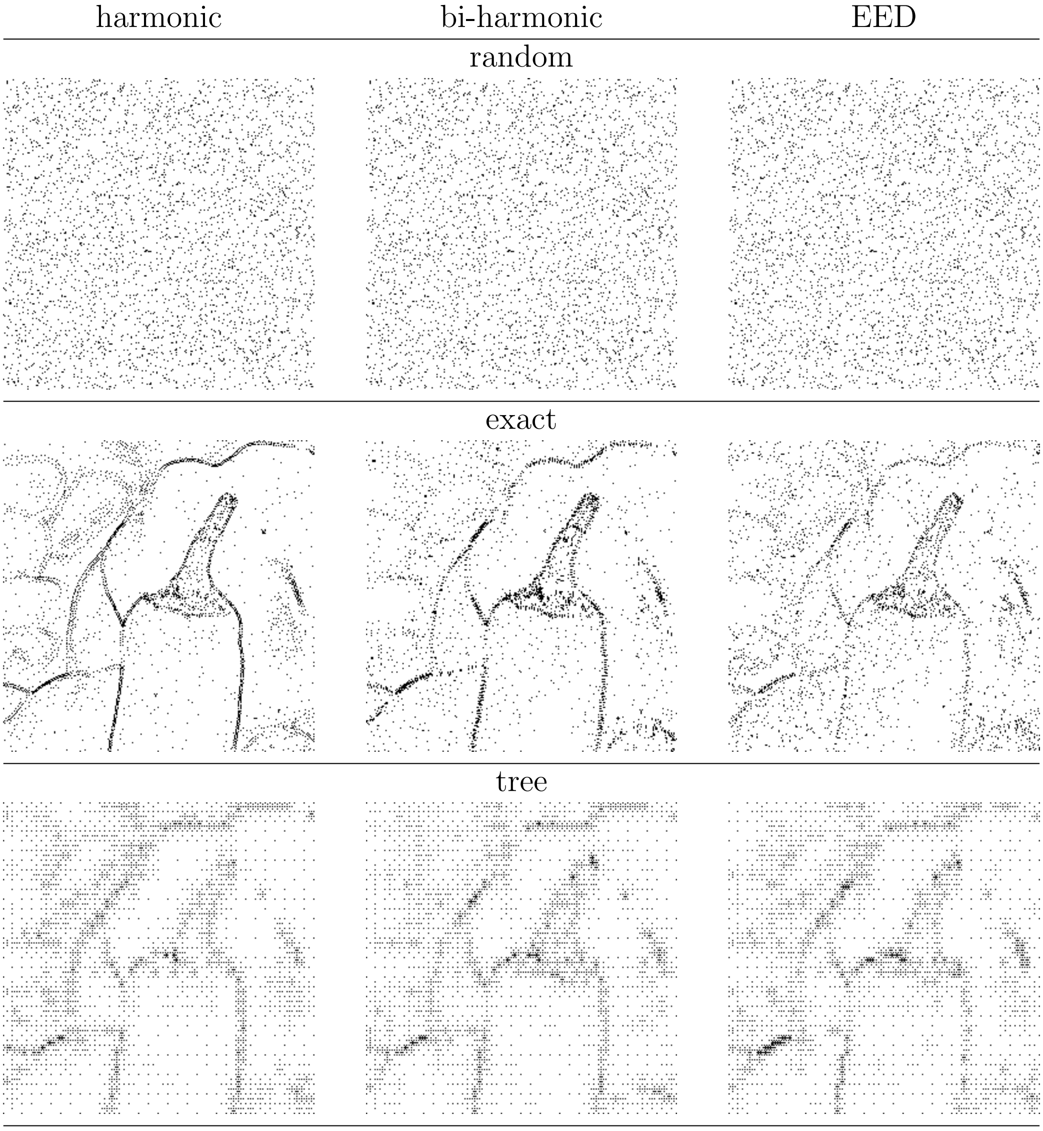

Figure 5: **Optimal Inpainting Masks for Different PDEs.** All masks contain 5% of the total image points of the test image *peppers*. The locations of known data are marked in black. **(a) Top row:** The same uniformly distributed random mask is used for all three diffusion types. **(b) Middle row:** For the harmonic mask, points are stored left and right of edges. In the biharmonic case, the structure remains similar, but points either spread out more or cluster closer together to store a whole region verbatim. The EED mask is spread out much more. **(c) Bottom row:** The masks from tree densification with nonlocal node exchange follow a similar pattern like the exact ones. However, the differences between the choice of locations is less pronounced than in the exact case due to the reduced number of possible choices.

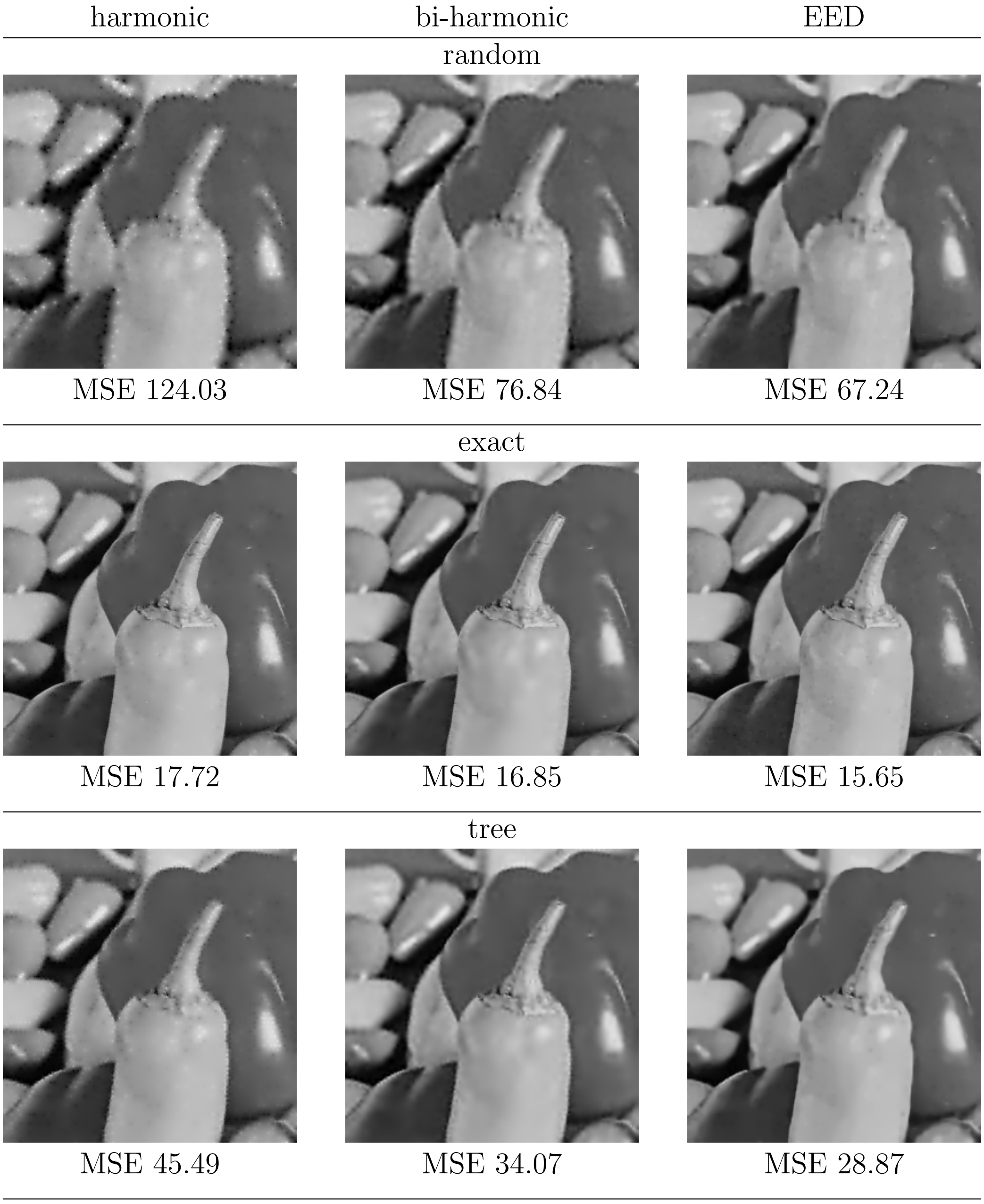

Figure 6: **Influence of Data Selection Strategies on Inpainting.** Reconstruction from the 5% masks from Fig. 5 with different inpainting operators. **(a) Top row:** Both biharmonic and EED perform much better than harmonic on random data. This already indicates their higher robustness to suboptimal known locations. Moreover, harmonic suffers from severe singularities. **(b) Middle row:** The difference between the three inpainting operators is much less pronounced for optimal spatial and tonal data. The singularities are still there for harmonic inpainting, but they are hardly visible in print. **(c) Bottom row:** Harmonic inpainting suffers the most from the restriction to a locally adaptive grid. Biharmonic and EED lose less quality compared to optimal locations.

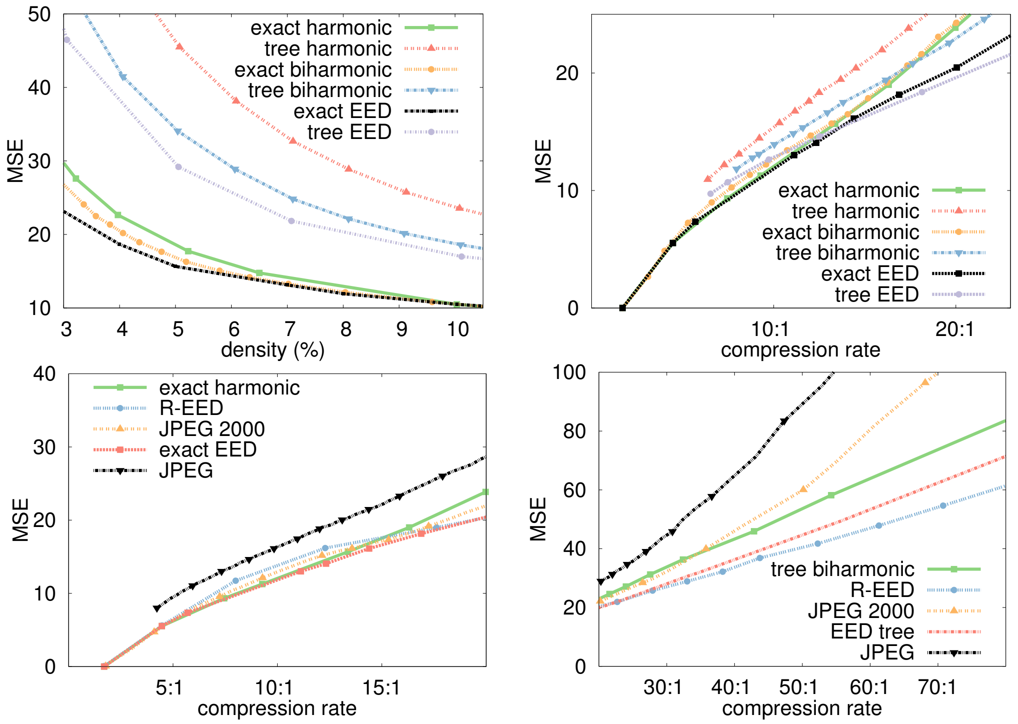

Figure 7: Comparisons for the 256 × 256 image *peppers*. The top row compares harmonic and biharmonic versions of our codecs, and the bottom row compares our best methods to transform coders and R-EED. **(a) Top Left:** Comparison at same mask density. **(b) Top Right:** Comparison at same compression ratio. **(c) Bottom Left:** Low to medium compression ratios. **(d) Bottom Right:** High compression ratios.

rates however, its robustness under both restricted locations and coarsely quantised grey values allow it to outperform its competitors.

Comparison to Other Encoders. In relation to transform-based coders, the tree-based method performs consistently better than JPEG and in many cases also outperforms JPEG2000 for compression rates larger than 35:1. Surprisingly, for very high compression rates, the heuristic approach of R-EED outperforms the more sophisticated stochastic tree densification with EED. While this seems counter-intuitive at first glance, there is a simple explanation: R-EED treats entropy coding and quantisation in a different fashion. Our method first selects a single tree and then optimises q for the best trade-off between file size and reconstruction quality. In contrast, R-EED defines a target compression ratio first and then builds a lot of different trees that fit to this ratio. Thus, it already incorporates coding costs into

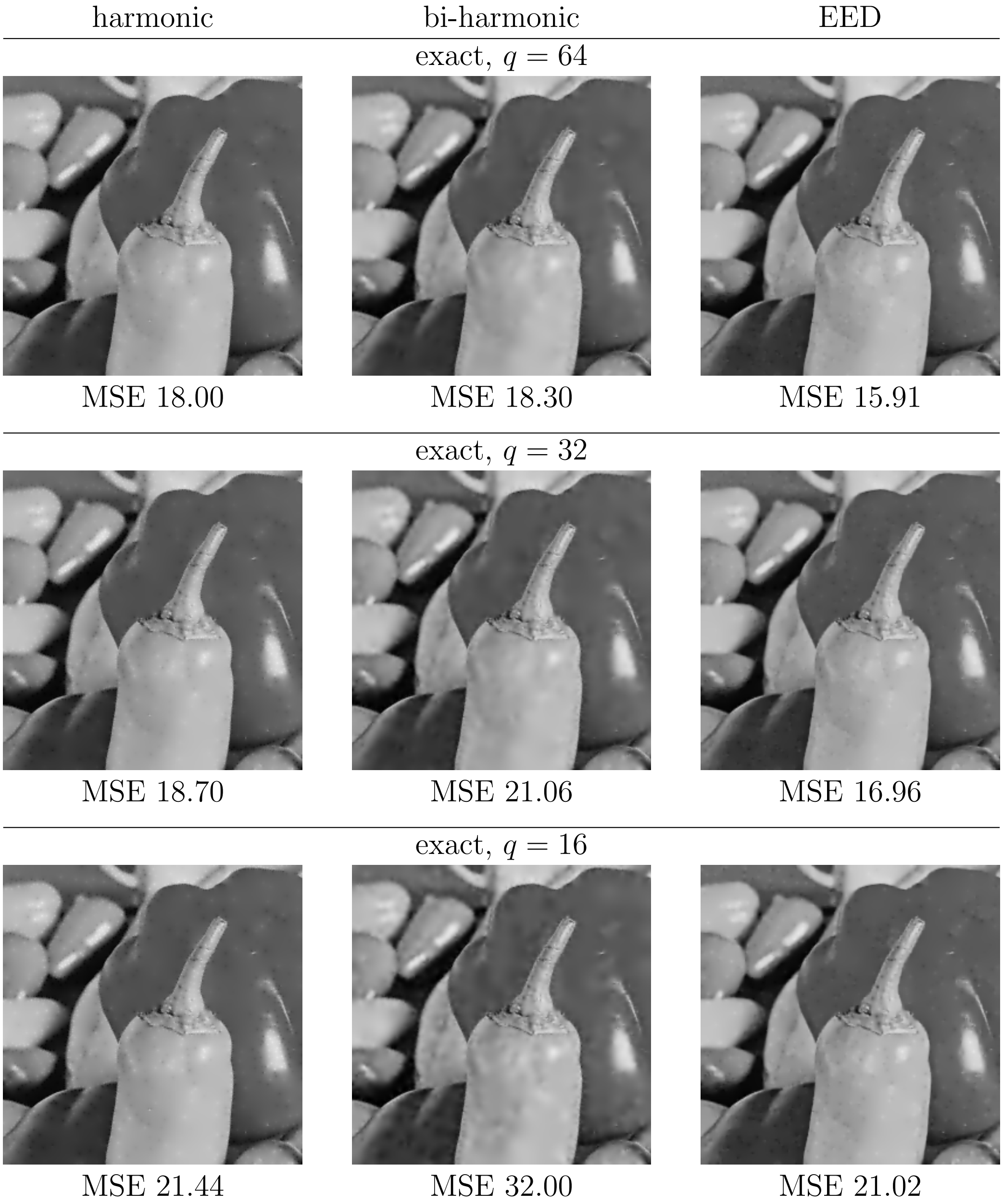

Figure 8: **Influence of Quantisation on Inpainting.** Reconstruction from the exact 5% masks from Fig. 5 with different inpainting operators. For all results, we have performed quantisation-aware grey value optimisation with 64, 32, and 16 different grey values. Reducing the number of grey values from 256 to 64 does not change the quality dramatically for all diffusion methods. However, the ranking of harmonic and biharmonic inpainting has already changed compared to Fig. 6. Reducing the parameter q even further reveals that harmonic and EED inpainting are much less sensitive to quantisation than biharmonic interpolation.

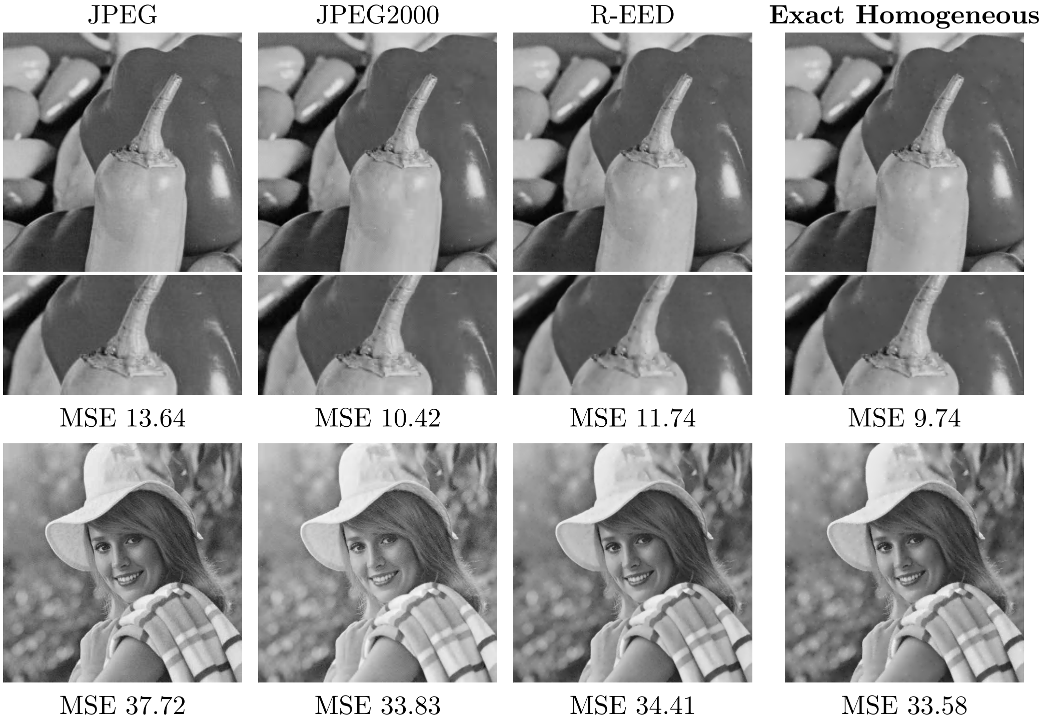

Figure 9: Compression results for *peppers* and *elaine*. **(a) Top:** Results for *peppers* (256×256 pixels) with compression ratio $\approx$ 8:1. **(b) Bottom:** Results for *elaine* (512×512 pixels) with compression ratio $\approx 18 : 1$.

the selection of known data, not afterwards. At low compression ratios, this does not have a high impact, but it makes a difference for very small files.
In comparison to other PDE-based methods, linear diffusion performs best in the area of low to medium compression rates (up to 15:1). Fig. 7 and Fig. 9 show that it can beat both R-EED and JPEG2000. On smooth images like *peppers*, harmonic diffusion with exact masks even outperforms JPEG2000. This demonstrates how powerful simple PDEs can be.
In Table 1, we allow both edge-enhancing anisotropic diffusion with probabilistic tree densification and exact harmonic masks obtained with optimal control. On low compression rates, all of our results outperform or match R-EED and are highly competitive to JPEG2000. Regarding high compression rates, the performance is comparable to R-EED. However, the more efficient integration of the entropy coder into the choice of mask points gives R-EED still a slight edge.

Table 1: MSE comparison on several test images. For the compression rate of 15:1 we use exact masks with homogeneous inpainting for elaine and lena. The rest of the error values for our method are obtained with tree-based EED.

Ratio	$\approx 15:1$				$\approx 60:1$			
Image	elaine	lena	trui	walter	elaine	lena	trui	walter
JPEG	34.69	20.06	16.24	6.73	77.39	97.37	116.98	69.40
JPEG2000	**31.23**	**14.73**	12.27	5.70	54.18	60.48	88.96	47.09
R-EED	35.48	16.56	11.27	5.48	**49.87**	56.96	**44.38**	24.05
Our Method	31.38	17.00	**10.48**	**4.53**	54.83	**53.56**	47.42	**21.74**

6 Conclusion

The investigation of the interplay between diffusion-based inpainting and different compression steps in this paper has both practical implications and provides general insights into codec design.

On the practical side, we have shown that codecs with parameter-free linear inpainting PDEs can beat both the quasi standard JPEG2000 of transform-based compression and the state of the art in PDE-based compression. This is an indication that simple and fast PDEs have a lot of potential that has not been fully used to this point. In particular, they might be integral for the design of *synchronous* compression codecs that are fast in both compression and decompression. However, the selection of known data still hinders this goal: Optimal control and stochastic sparsification have not yet reached real-time performance, but linear diffusion needs thorough optimisation to produce competitive quality. Therefore, future research should focus on faster mask selection algorithms.

A valuable general insight gained from our evaluation concerns the comparison of inpainting operators: *The performance of PDEs for compression can only be evaluated in the context of actual codecs.* Comparisons that do not consider all compression steps can lead to false rankings of inpainting operators that do not reflect their real compression capabilities. In particular, the sensitivity of the biharmonic operator to coarsely quantised known data makes the simpler harmonic diffusion the preferable choice for compression. If mask positions are suboptimal, but cheap to store, EED performs best. Note, however, that we have only covered optimisation-driven compression approaches with our paper. In the future we would like to investigate the potential of semantic approaches that combine segmentation or edge detection with diffusion. So far, these approaches have only been used in specialised applications such as depth map compression.

References

[1] Z. Belhachmi, D. Bucur, B. Burgeth, J. Weickert, How to choose interpolation data in images, SIAM Journal on Applied Mathematics 70 (1) (2009) 333–352.

[2] M. Mainberger, S. Hoffmann, J. Weickert, C. H. Tang, D. Johannsen, F. Neumann, B. Doerr, Optimising spatial and tonal data for homogeneous diffusion inpainting., in: A. Bruckstein, B. ter Haar Romeny, A. Bronstein, M. Bronstein (Eds.), Scale Space and Variational Methods in Computer Vision, vol. 6667 of *Lecture Notes in Computer Science*, Springer, Berlin, 26–37, 2011.

[3] L. Hoeltgen, S. Setzer, J. Weickert, An optimal control approach to find sparse data for Laplace interpolation, in: A. Heyden, F. Kahl, C. Olsson, M. Oskarsson, X.-C. Tai (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition, vol. 8081 of *Lecture Notes in Computer Science*, Springer, Berlin, 151–164, 2013.

[4] Y. Chen, R. Ranftl, T. Pock, A bi-level view of inpainting-based image compression, in: Proc. 19th Computer Vision Winter Workshop, Křtiny, Czech Republic, 19–26, 2014.

[5] C. Schmaltz, P. Peter, M. Mainberger, F. Ebel, J. Weickert, A. Bruhn, Understanding, optimising, and extending data compression with anisotropic diffusion, International Journal of Computer Vision 108 (3) (2014) 222–240.

[6] J. Weickert, Theoretical foundations of anisotropic diffusion in image processing, in: W. Kropatsch, R. Klette, F. Solina, R. Albrecht (Eds.), Theoretical Foundations of Computer Vision, vol. 11 of *Computing Supplement*, Springer, Vienna, Austria, 221–236, 1996.

[7] M. Mainberger, J. Weickert, Edge-based image compression with homogeneous diffusion, in: X. Jiang, N. Petkov (Eds.), Computer Analysis of Images and Patterns, vol. 5702 of *Lecture Notes in Computer Science*, Springer, Berlin, 476–483, 2009.

[8] J. Gautier, O. Le Meur, C. Guillemot, Efficient depth map compression based on lossless edge coding and diffusion, in: Proc. 29th Picture Coding Symposium, Kraków, Poland, 81–84, 2012.

[9] Y. Li, M. Sjostrom, U. Jennehag, R. Olsson, A scalable coding approach for high quality depth image compression., in: Proc. 3DTV-Conference:

The True Vision - Capture, Transmission and Display of 3D Video, Zurich, Switzerland, 1–4, 2012.

[10] S. Hoffmann, M. Mainberger, J. Weickert, M. Puhl, Compression of depth maps with segment-based homogeneous diffusion, in: A. Kuijper, K. Bredies, T. Pock, H. Bischof (Eds.), Scale-Space and Variational Methods in Computer Vision, vol. 7893 of *Lecture Notes in Computer Science*, Springer, Berlin, 319–330, 2013.

[11] I. Galić, J. Weickert, M. Welk, A. Bruhn, A. Belyaev, H.-P. Seidel, Towards PDE-based image compression, in: N. Paragios, O. Faugeras, T. Chan, C. Schnörr (Eds.), Variational, Geometric and Level-Set Methods in Computer Vision, vol. 3752 of *Lecture Notes in Computer Science*, Springer, Berlin, 37–48, 2005.

[12] P. Peter, S. Hoffmann, F. Nedwed, L. Hoeltgen, J. Weickert, From optimised inpainting with linear PDEs towards competitive image compression codecs, in: T. Bräunl, B. McCane, M. Rivers, X. Yu (Eds.), Advances in Image and Video Technology, Lecture Notes in Computer Science, Springer, Berlin, in press, 2016.

[13] H. Werner, Studies on contour, The American Journal of Psychology 47 (1) (1935) 40–64.

[14] J. H. Elder, Are edges incomplete?, International Journal of Computer Vision 34 (2/3) (1999) 97–122.

[15] B. F. Logan, Jr., Information in the zero crossings of bandpass signals, Bell System Technical Journal 56 (1977) 487–510.

[16] A. L. Yuille, T. A. Poggio, Scaling theorems for zero crossings, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (1) (1986) 15–25.

[17] D. Rotem, Y. Zeevi, Image reconstruction from zero-crossings, IEEE Transactions on Acoustics, Speech, and Signal Processing 34 (1986) 1269–1277.

[18] S. Chen, Image reconstruction from zero-crossings, in: Proc. 10th International Joint Conference on Artificial Intelligence, vol. 2, Milan, Italy, 742–744, 1987.

[19] S. R. Curtis, A. V. Oppenheim, J. S. Lim, Reconstruction of two-dimensional signals from threshold crossings, in: Proc. IEEE International Conference on Acoustics, Speech and Signal Processing, vol. 10, Tampa, FL, 1057–1060, 1985.

[20] R. Hummel, R. Moniot, Reconstructions from zero-crossings in scale space, IEEE Transactions on Acoustics, Speech, and Signal Processing 37 (12) (1989) 2111–2130.

[21] P. Grattoni, A. Guiducci, Contour coding for image description, Pattern Recognition Letters 11 (2) (1990) 95–105.

[22] S. Mallat, S. Zhong, Characterisation of signals from multiscale edges, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (7) (1992) 720–732.

[23] L. Dron, The multiscale veto model: A two-stage analog network for edge detection and image reconstruction, International Journal of Computer Vision 11 (1) (1993) 45–61.

[24] C. Saloma, P. Haeberli, Two-dimensional image reconstruction from Fourier coefficients computed directly from zero crossings, Applied Optics 32 (17) (1993) 3092–3093.

[25] J. Canny, A computational approach to edge detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 8 (6) (1986) 679–698.

[26] M. Mainberger, A. Bruhn, J. Weickert, S. Forchhammer, Edge-based compression of cartoon-like images with homogeneous diffusion, Pattern Recognition 44 (9) (2011) 1859–1873.

[27] S. Carlsson, Sketch based coding of grey level images, Signal Processing 15 (1) (1988) 57–83.

[28] U. Y. Desai, M. M. Mizuki, I. Masaki, B. K. P. Horn, Edge and Mean Based Image Compression, Tech. Rep. 1584 (A.I. Memo), Artificial Intelligence Lab, Massachusetts Institute of Technology, Cambridge, MA, 1996.

[29] Y. Wu, H. Zhang, Y. Sun, H. Guo, Two image compression schemes based on image inpainting, in: Proc. 2009 International Joint Conference on Computational Sciences and Optimization, Sanya, China, 816–820, 2009.

[30] V. Bastani, M. S. Helfroush, K. Kasiri, Image compression based on spatial redundancy removal and image inpainting, Journal of Zhejiang University – Science C (Computers & Electronics) 11 (2) (2010) 92–100.

[31] C. Zhao, M. Du, Image Compression based on PDEs, in: Proc. 2011 International Conference of Computer Science and Network Technology, vol. 3, Harbin, China, 1768–1771, 2011.

[32] P. Peter, J. Weickert, Colour image compression with anisotropic diffusion., in: Proc. 21st IEEE International Conference on Image Processing, Paris, France, 4822–4862, 2014.

[33] C. Schmaltz, N. Mach, M. Mainberger, J. Weickert, Progressive modes in PDE-based image compression., in: Proc. 30th Picture Coding Symposium, San Jose, CA, 233–236, 2013.

[34] L. Hoeltgen, M. Mainberger, S. Hoffmann, J. Weickert, C. H. Tang, S. Setzer, D. Johannsen, F. Neumann, B. Doerr, Optimising Spatial and Tonal Data for PDE-based Inpainting, in: Variational Methods in Image Analysis, De Gruyter, Berlin, in press, 2016.

[35] T. Iijima, Basic theory on normalization of pattern (in case of typical one-dimensional pattern), Bulletin of the Electrotechnical Laboratory 26 (1962) 368–388, in Japanese.

[36] P. Charbonnier, L. Blanc-Féraud, G. Aubert, M. Barlaud, Two deterministic half-quadratic regularization algorithms for computed imaging, in: Proc. First IEEE International Conference on Image Processing, vol. 2, Austin, TX, 168–172, 1994.

[37] L. Hoeltgen, J. Weickert, Why does non-binary mask optimisation work for diffusion-based image compression?, in: X.-C. Tai, E. Bae, T. Chan, M. Lysaker (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition, vol. 8932 of *Lecture Notes in Computer Science*, Springer, Berlin, 85–98, 2015.

[38] S. Grewenig, J. Weickert, A. Bruhn, From box filtering to fast explicit diffusion, in: M. Goesele, S. Roth, A. Kuijper, B. Schiele, K. Schindler (Eds.), Pattern Recognition, vol. 6376 of *Lecture Notes in Computer Science*, Springer, Berlin, 533–542, 2010.

[39] D. A. Huffman, A method for the construction of minimum redundancy codes, Proceedings of the IRE 40 (9) (1952) 1098–1101.

[40] J. J. Rissanen, Generalized Kraft inequality and arithmetic coding, IBM Journal of Research and Development 20 (3) (1976) 198–203.

[41] M. Mahoney, Adaptive weighing of context models for lossless data compression, Tech. Rep. CS-2005-16, Florida Institute of Technology, Melbourne, FL, 2005.

[42] A. Scheer, Entropy Coding for PDE-Based Image Compression, Bachelor thesis, Department of Computer Science, Saarland University, Saarbrücken, Germany, 2014.

[43] Joint Bi-level Image Experts Group, Information Technology Progressive Lossy/Lossless Coding of Bi-level Images, Standard, International Organization for Standardization, 1993.

[44] P. G. Howard, F. Kossentini, B. Martins, S. Forchhammer, W. J. Rucklidge, The emerging JBIG2 standard, IEEE Transactions on Circuits, Systems and Video Technology 8 (7) (1998) 838–848.

[45] L. Bottou, P. Haffner, P. G. Howard, P. Simard, Y. Bengio, Y. LeCun, High quality document image compression with DjVu, Journal of Electronic Imaging 7 (3) (1998) 410–425.

[46] G. Zeng, N. Ahmed, A block coding technique for encoding sparse binary patterns, IEEE Transactions on Acoustics, Speech, and Signal Processing 37 (5) (1989) 778–780.

[47] P. Fränti, O. Nevalainen, Compression of binary images by composite methods based on block coding, Journal of Visual Communication and Image Representation 6 (4) (1995) 366–377.

[48] S. A. Mohamed, M. Fahmy, Binary image compression using efficient partitioning into rectangular regions, IEEE Transactions on Communications 43 (5) (1995) 1888–1893.

[49] S. Zahir, M. Naqvi, A near minimum sparse pattern coding based scheme for binary image compression, in: Proc. 2005 IEEE International Conference on Image Processing, vol. 2, Genova, Italy, II–289, 2005.

[50] J. Max, Quantizing for minimum distortion, IRE Transactions on Information Theory 6 (1) (1960) 7–12.

[51] S. P. Lloyd, Least squares quantization in PCM, IEEE Transactions on Information Theory 28 (2) (1982) 129–137.

[52] A. Gersho, R. M. Gray, Vector Quantization and Signal Compression, vol. 159 of *The Springer International Series in Engineering and Computer Science*, Springer, New York, 1992.

[53] S. Hoffmann, G. Plonka, J. Weickert, Discrete Green's functions for harmonic and biharmonic inpainting with sparse atoms, in: X.-C. Tai, E. Bae, T. F. Chan, M. Lysaker (Eds.), Energy Minimization Methods in Computer Vision and Pattern Recognition, vol. 8932 of *Lecture Notes in Computer Science*, Springer, Berlin, 169–182, 2015.