PDE-based Lossless Medical Image Compression

DISSERTATION

zur Erlangung des Doktorgrades (Dr. rer. nat.)
der
Mathematisch-Naturwissenschaftlichen Fakultät
der
Rheinischen Friedrich-Wilhelms-Universität Bonn

vorgelegt von

Ikram JUMAKULYYEV

aus Lebap, Turkmenistan

Bonn, Januar 2024

Angefertigt mit Genehmigung der Mathematisch-Naturwissenschaftlichen Fakultät
der Rheinischen Friedrich-Wilhelms-Universität Bonn

1. Gutachter: Prof. Dr. Thomas Schultz
2. Gutachter: Prof. Dr. Frank Kurth

Tag der mündlichen Prüfung: 04.06.2024
Erscheinungsjahr: 2024

Abstract

Ikram JUMAKULYYEV

PDE-based Lossless Medical Image Compression

In this dissertation, our scientific goal is to investigate the potential of Partial Differential Equations (PDEs) for designing a lossless medical image compression codec. PDEs has previously been used as a basis for lossy image compression. The famous PDE used to design a lossy codec is called Edge-enhancing diffusion (EED) which is a family member of inpainting-based PDEs. It can reconstruct a close approximation of an original image from a small subset of its pixels. This makes it an attractive foundation for PDE based image compression. For medical images on the other hand, lossless compression is often considered to be safer, given that even subtle details could be diagnostically relevant.

The dissertation consists mainly of three of our contributions which build upon each other. In our first work, we designed PDEs that give better reconstruction than EED, especially in a very sparse subset of pixels. In this work, we generalize second-order EED to a fourth-order counterpart. It involves a fourth-order diffusion tensor that is constructed from the regularized image gradient in a similar way as in traditional second-order EED, permitting diffusion along edges, while applying a nonlinear diffusivity function across them. We show that our fourth-order diffusion tensor formalism provides a unifying framework for all previous anisotropic fourth-order diffusion based methods, and that it provides additional flexibility. We achieve an efficient implementation using a fast semi-iterative scheme. Experimental results on natural and medical images suggest that our novel fourth-order method produces more accurate reconstructions compared to the existing second-order EED.

In our second work, we introduce a PDE-based codec that achieves competitive compression rates for lossless image compression. It is based on coding the differences between the original image and its PDE-based reconstruction. These differences often have lower entropy than the original image, and can therefore be coded more efficiently. We optimize this idea via an iterative reconstruction scheme, and a separate coding of empty space, which takes up a considerable fraction of the field of view in many 3D medical images. We demonstrate that our PDE-based codec compares favorably to previously established lossless codecs. We also investigate the individual benefit from each ingredient of our codec on multiple examples, explore the effect of using homogeneous, edge enhancing, and fourth-order anisotropic diffusion, and discuss the choice of contrast parameters.

For our last work, we focused on diffusion Magnetic Resonance Image (dMRI) lossless compression as, to our best knowledge, there is no codec designed to exploit the properties of dMRI data. Diffusion MRI is a modern neuroimaging modality with a unique ability to acquire microstructural information by measuring water

self-diffusion at the voxel level. However, it generates huge amounts of data, resulting from a large number of repeated 3D scans. Each volume samples a location in q-space, indicating the direction and strength of a diffusion sensitizing gradient during the measurement. This captures detailed information about the self-diffusion, and the tissue microstructure that restricts it. Lossless compression with GZIP is widely used to reduce the memory requirements. We introduce a novel lossless codec for diffusion MRI data. It reduces file sizes by more than 30% compared to GZIP, and also beats lossless codecs from the JPEG family. Our codec builds on our previous work on lossless PDE-based compression of 3D medical images, but additionally exploits smoothness in q-space. We demonstrate that, compared to using only image space PDEs, q-space PDEs further improve compression rates. Moreover, implementing them with Finite Element Methods and a custom acceleration significantly reduces computational expense. Finally, we show that our codec clearly benefits from integrating subject motion correction, and slightly from optimizing the order in which the 3D volumes are coded.

Acknowledgements

First of all, I would like to thank my supervisor Prof. Dr. Thomas Schultz who supported and guided me to apply and win a German Academic Exchange Service (DAAD) Ph.D. scholarship for pursuing doctoral research in an interesting topic and later for his advice and constructive feedback throughout working on this dissertation. I will not forget the gracious supports he provided throughout my time on his team. This dissertation would not have been possible without his invaluable insights and expertise, and I could not have imagined having a better supervisor and mentor for my research.

I would like to greatly thank Prof. Dr. Frank Kurth for reviewing my dissertation and giving insightful comments.

I would like to thank my all colleagues for an wonderful working envirement. Thank you for your friendship and all interesting moments we had.

Finally and most importantly, my acknowledgments would not be complete without giving thanks to my friends and family for their encouraging support, especially my parents and my wife. Their loving support through all stages of PhD life kept me motivated to complete this dissertation.

Contents

List of Figures

List of Tables

List of Abbreviations

AAE	**A**verage **A**bsolute **E**rror
BH	**B**iharmonic **H**omogenous
C-EED	**C**uboidal **E**dge-**E**nhancing **D**iffusion based codec
CT	**C**omputed **T**omography
DICOM	**D**igital **I**maging and **C**ommunications in **M**edicine
dMRI	**d**iffusion **M**agnetic **R**esonance **I**maging
EED	**E**dge-**E**nhancing **D**iffusion
FOEED	**F**ourth-**O**rder **E**dge-**E**nhancing **D**iffusion
FDM	**F**inite **D**ifference **M**ethods
FEM	**F**inite **E**lement **M**ethods
FSI	**F**ast **S**emi-**I**terative **S**cheme
JPEG	**J**oint **P**hotographic **E**xperts **G**roup
PDE	**P**artial **D**ifferential **E**quation
R-EED	**R**ectangular subdivision **E**dge-**E**nhancing **D**iffusion based codec
LH	**L**inear **H**omogenous
MPEG	**M**oving **P**icture **E**xperts **G**roup
MSE	**M**ean **S**quared **E**rror
NIfTI	**N**euroimaging **I**n**f**ormatics **T**echnology **I**nitiative

Chapter 1

Introduction

1.1 Motivation

In hospitals and clinics, ever-developing medical imaging devices generate large amounts of data per patient for diagnosis, therapy and surgical planning purposes. Continuous medical data acquisition creates a need for efficient storage to overcome memory and limited bandwidth network transmission restrictions. Another point is that aggregating medical data as large datasets is necessary in research activities for developing automated or semi-automated human assistance techniques in medical image assesment or diagnosis as there is continuous effort in transforming healthcare systems currently centered on human expert assessment [54]. It has been reported (see Figure 1.1) that each year the acquired overall neuroimaging data size grows exponentially [26]. It is due to the proliferation of medical imaging devices, their increased resolution, and the increasing use of multiple contrasts or channels. The need for long-term storage and efficient transmission makes the development of compression schemes for the storage of medical images an important and active research goal.

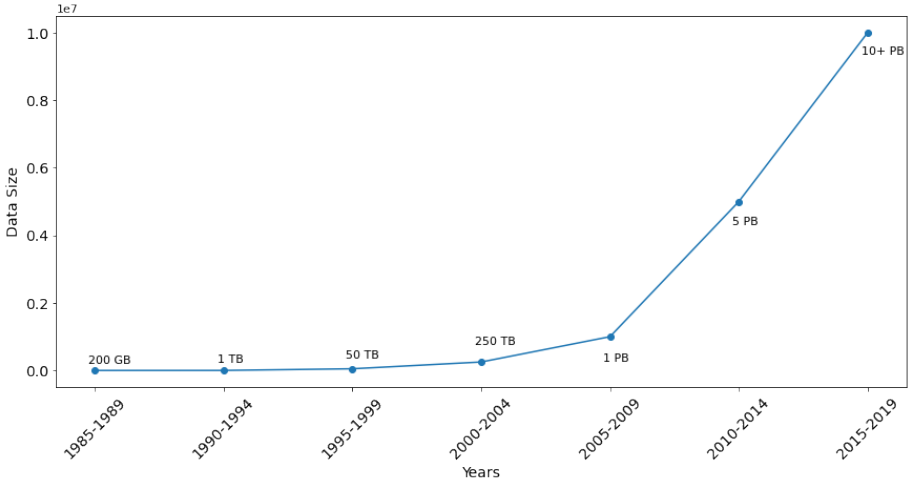


FIGURE 1.1: Increase of neuroimaging data size over years. Data were taken from I. D. Dinov [26]. GB = gigabyte (10^9 bytes), TB = terabyte (10^{12} bytes), PB = petabyte (10^{15} bytes).

In general, image compression techniques are mainly divided into two groups: Lossy and lossless compression. Lossy compression achieves much higher compression rates by replacing the original image with an approximation that can be stored more efficiently. However, it might lead to visually noticeable changes that are less

suitable for medical images, since potentially subtle, but diagnostically relevant details might be perturbed. Lossless compression techniques permit restoration of the full, unmodified input data, which however limits the achievable compression rates. In medical image compression, lossless compression is often preferred and is sometimes even legally required [52, 53, 75], since it guarantees not to interfere with the interpretation or quantification of the image contents. Therefore, this whole dissertation is devoted to designing a lossless compression codec for medical image data.

The use of diffusion-based inpainting has been explored for the lossy compression of images [30, 96, 97], videos [5, 56, 89], and audio [85]. This paradigm is based on storing information only for a sparse subset of the original samples, and interpolating it to approximate the remaining parts of the original signal. Interpolation is often done via Partial Differential Equations (PDEs) that are inspired by the well-known heat transfer equation, in analogy to how radiators that are sparsely distributed in a room would heat up the space in between them.

Medical images are usually volumetric 3D or series of volumes, 4D data. And almost all previous works on PDE-based compression have focused on 2D natural images or videos. Only a single example has considered a 3D extension [84]. Even more importantly, all above-mentioned codecs are for lossy compression, and their benefit relative to established transform-based codecs like JPEG [81] and JPEG2000 [101] tends to be most pronounced at high compression rates [97]. Usually, a high compression rate means more loss which is not desirable in medical images. To fill this gap, we worked on PDE-based lossless medical image compression.

1.2 Contributions

As a continuation of a research line that has explored the use of Partial Differential Equations (PDEs) for image compression [29, 30, 70, 97] we first designed a diffusion-based PDEs for image inpainting. The use of inpainting PDEs for image compression is based on storing only a small subset of all pixels, and interpolating between them in order to restore the remaining ones. There is a strong similarity between that interpolation process and image inpainting, whose goal it is to reconstruct missing or corrupted parts of an image. PDE-based methods for image inpainting and compression are inspired by the physical phenomenon of heat transport. We introduced a novel fourth-order PDE for edge enhancing diffusion (FOEED), steered by a fourth-order diffusion tensor. We implemented it using a fast semi-iterative scheme (FSI) and demonstrated that it achieved improved accuracy in several inpainting tasks, including reconstructing images from a small fraction of pixels or removing scratches. Our main motivation for using fourth-order diffusion in this context is the increased smoothness of results compared to second-order PDEs [114], which we expected to result in visually more pleasant and numerically more accurate reconstructions. Better reconstruction, i.e. more accurate approximation of the original image leads to a more compressible residual which is highly important to achieve lossless compression.

Next, we proposed our codec which is the first to explore the potential of PDE-based methods for the lossless compression of 3D medical images. We presented a lossless PDE-based codec that stores the residuals between the PDE-based reconstruction and the original values. Its success rests on three key ideas: First, we use a simple regular grid as the initial inpainting mask, so that the locations of the mask voxels do not have to be stored explicitly. Second, we encode and decode the image iteratively, alternating between PDE-based reconstructions and dilation of the

inpainting mask. Compared to a single reconstruction, this further reduces the entropy of the residuals that have to be stored. Third, we optionally code regions of empty space separately, since they take up a substantial fraction of the field of view in many medical images. We demonstrate that our codec achieves a higher compression rate than several established codecs on three Magnetic Resonance Images with different characteristics, as well as a Computed Tomography image. Moreover, we studied the effect of several variations of our codec, using different PDEs including FOEED [47], iteration modes, and contrast parameters.

Furthermore, we propose a PDE-based codec specifically designed for lossless compression of 4D diffusion Magnetic Resonance Imaging (dMRI) data. Our focus on dMRI data has two reasons: 1) To our best knowledge, neither PDE-based codec nor any other kind of compression codecs exists in the literature for dMRI specific (see Sec. 6.3); 2) With simple linear second-order and fourth-order inpainting PDEs [87] we can exploit redundancies along the fourth dimension that arises in diffusion MRI, i.e., the orientation of the diffusion gradient [7]. Especially being linear for inpainting PDEs, allows us to apply our developed fast implementation trick (see Sec. 6.4.2); The numerical PDEs with Finite Element Methods (FEM) numerical scheme is implemented by using the package called FEniCS [58, 66].

1.3 Outline and publications

A short introduction to image compression is given in Chapter 2. Chapter 3 provides mathematical background and an overview of diffusion PDEs used in image processing and compression.

The main research contribution of this dissertation is based on our three publications. These publications are presented in:

- Chapter 4: Jumakulyyev et al. "Fourth-order anisotropic diffusion for inpainting and image compression" In: *Anisotropy Across Fields and Scales.* Springer, Cham, 2021, pp. 99-124. [47].
- Chapter 5: Jumakulyyev et al. "Lossless PDE-based Compression of 3D Medical Images" In: *International Conference on Scale Space and Variational Methods in Computer Vision.* Springer. 2021, pp. 450-462. [50].
- Chapter 6: Jumakulyyev et al. "PDE-based Lossless Compression of 4D diffusion MRI" In: *Journal of Mathematical Imaging and Vision.* Springer Nature. 65 (2023), pp. 644-656. [46].

Chapter 7 concludes all the proposed methods and discusses future possibilities to extend these works.

Chapter 2

Image Compression

Image compression algorithms represent the same or similar quantity of image information in a reduced amount of data. It is a broad domain with a lot of applications in real life. We use image compression explicitly or implicitly. For example, all our pictures captured by our mobile phone cameras are compressed automatically and saved in our phone memories in a compact compressed form. Image compression is crucial for various applications, especially when dealing with digital images in scenarios where storage space, transmission bandwidth, or processing speed are limited resources. Image compression techniques can be categorized into two main groups: lossless compression and lossy compression. Lossless compression methods reduce the size of an image without any loss of information. This means that when the compressed image is decompressed, it is identical to the original. Lossy compression methods achieve higher compression ratios by discarding some of the less critical information in an image. While these methods result in a smaller file size, there is some loss of image quality. The choice between lossless and lossy compression depends on the specific use case. Lossless compression is preferred when preserving every bit of data is critical, such as in medical imaging or archival purposes. Lossy compression is suitable for scenarios where a trade-off between image quality and file size is acceptable, such as in web graphics and multimedia content.

This chapter is devoted to getting a reader familiar with image compression concepts that are necessary to understand upcoming chapters. For in-depth discussions, please refer to dedicated image compression literature, e.g., [92, 93].

2.1 Digital Images

A digital image is a captured picture of a continuous scene or object by a digital device such as a digital camera or medical image acquisition device. Digital images are stored in a digital medium in discrete form; They can be considered as discrete image functions defined on two-dimensional (2D) or three-dimensional (3D) rectilinear grids depending on image dimension. The famous example of 2D images are natural images that are, for example, pictures of nature, everyday moments etc. For 3D images, volumetric medical images acquired by specially designed medical devices are an example. Discrete image function can be real-valued or vector-valued. Greyscale images are real-valued images, while color images are vector-valued images. They have three red, green, and blue color components.

There are terms used to describe digital images, such as pixel or voxel, resolution, color model, bit depth, file format and metadata. A pixel is the fundamental building block of a digital image. It is a tiny square or rectangular element that holds color or intensity information. Pixels are arranged in rows and columns to form a grid, commonly referred to as the image matrix. A voxel is a 3D correspondent of

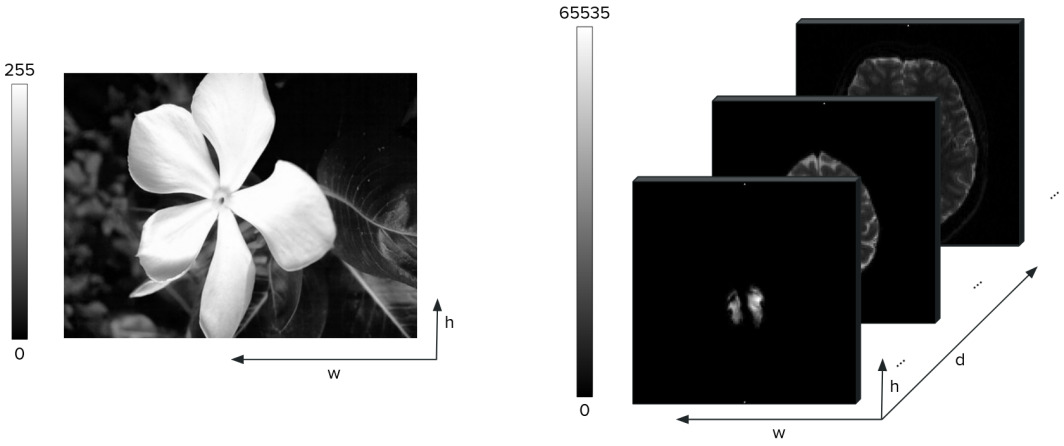


FIGURE 2.1: Left: Flower *natural image* with 256 grey-scale levels and $w \times h = 400 \times 300$ resolution. Right: $b = 0$ diffusion weighted MRI *medical image* with 4096 grey-scale levels and $w \times h \times d = 136 \times 136 \times 84$ voxels.

TABLE 2.1: Raw image file sizes. Visualization of the images are given in Figure 2.1.

Image	Resolution	Bit Depth	Number of Bits
flower	400×300	8 bits	960.000
B0 MRI	136×136×84	16 bits	248.586.24

a pixel. The resolution tells the number of pixels in an image. Digital images use color models to represent colors. The most common color models are RGB (Red, Green, Blue) and Grayscale. In RGB, each pixel is represented as a combination of red, green, and blue intensities. Mixing these three primary colors in different proportions produces a wide range of colors. Grayscale which is a single-channel color model that represents the intensity of light, usually from black to white. Each pixel has only one value representing its brightness. Bit depth determines a set of possible values that pixels can take or refers to the number of bits used to represent the color of each pixel/voxel, e.g., 8 bit grey-scale image pixels and 16 bit grey-scale 3D medical image voxels can have values from intervals $[0, 2^8 - 1]$ and $[0, 2^{16} - 1]$ respectively (see Figure 2.1). For a color RGB image 24 (3x8)-bits are used. For a raw and uncompressed image we can estimate the file size (number of bits) needed for memory storage if we know its resolution and bit depth. It is calculated by $resolution \times bit\ depth$. For an image with multiple color channels we should additionally multiply by the number of channels, i.e., $resolution \times bit\ depth \times channel\ number$. The last column of Table 2.1 presents our example images' (Figure 2.1) raw file sizes in bits. It's worth to note that those file sizes are required for image data storage. Usually image data is contained in a certain file format which has a header file containing image specifications such as image dimensions (width, height and depth if there is any), bit depth, etc. All this information is known as a metadata. A header file containing metadata takes relatively little memory.

2.2 Entropy Encoding

Entropy (also known as Shannon entropy) of an image is an average measure of information content in an image [99]. Images with high entropy have a more diverse

range of pixel values, suggesting that the image contains a lot of detail or information, while images with low entropy have less diversity in pixel values, indicating more uniform regions or less information. The entropy $H(I)$ of an image I is formulated as follows:

$$H(I) = -\sum_{i=0}^{2^{bd}} p(x_i) \log_2(p(x_i)) \,, \tag{2.1}$$

where x_i is a unique pixel value of an image I in the range of $[0, 2^{bd}]$, bd is the bit depth of I and $p(x_i)$ is the probability of x_i. The unit of image entropy $H(I)$ is bits. It is a consequence of using logarithms in base two. The entropy is helpful in image compression as it gives a lower limit for the average coding length in bits per pixel which can be achieved (or realized) by an entropy encoding scheme without any loss of information [104]. It is also used for various image analysis tasks, such as image segmentation, texture analysis, and feature extraction, to characterize the complexity and information content of an image.

Entropy coding is a method of assigning fixed- or variable-length codewords (strings of 0s and 1s) to the corresponding symbols according to the probability of the symbols (in our case unique pixel/voxel values). More formally, it is a data (in our case, image) compression technique used to represent data in a more compact form by taking advantage of the inherent statistical properties of the data. The main idea behind entropy encoding is to assign shorter codes to more frequent symbols and longer codes to less frequent symbols, resulting in an overall reduction in the size of the encoded data. In this way an average code length of the data is approximated to the entropy of the data, which is the most efficient way to represent the data. Two of the well-known entropy encoding techniques are Huffman and Arithmetic coding.

Huffman entropy encoding [40] is a technique of designing variable-length codes for coding given the probabilities of the symbols. The designed output code by Huffmann coding is uniquely decodable. The obtained codewords are prefix-free, i.e., longer codewords are not starting with shorter codewords which confuses decoder [41]. The ideal probability distribution for Huffmann coding i.e., it works best when the probabilities of symbols are powers of two or close to that. Huffman encoding has been used widely in many applications. It is the main component of JPEG image and MPEG video compression standards.

Arithmetic coding is another entropy encoding technique, and it is often used as an alternative to Huffman coding. Like Huffman coding, arithmetic coding works by assigning variable-length codes to symbols based on their probabilities of occurrence. Unlike Huffman coding, arithmetic coding does not use a discrete number of bits for symbol coding it. Rather encodes entire data sequences using fractional values in the interval [0, 1). Arithmetic coding can achieve higher compression efficiency compared to Huffman coding, especially for sources with uneven symbol probabilities. However, it requires more complex implementation due to the continuous nature of interval arithmetic. In general, Arithmetic coding is more efficient than Huffman coding because it encodes entire sequences of symbols in one step, rather than encoding symbols one by one as mentioned shortly before. This leads to a potentially more accurate representation of the original data's probabilities and can result in better compression ratios [112]. The choice between them often depends on the specific characteristics of the data being compressed and the trade-off between simplicity and efficiency.

2.3 Common Compression Algorithms

2.3.1 JPEG Family

JPEG [80], which stands for Joint Photographic Experts Group, is a widely used image compression standard that was developed by the Joint Photographic Experts Group committee in 1992. It has been designed to efficiently compress and store digital images while maintaining a balance between image quality and file size. JPEG is particularly well-suited for photographs and images with complex color gradients, making it one of the most popular formats for storing. It achieves compression by exploiting the limitations of human visual perception. It discards certain image information that the human eye is less sensitive to, while retaining the most critical visual elements. JPEG has lossy and lossless compression modes. For lossy modes JPEG and JPEG2000 are famous examples. For lossless modes there are lossless versions of JPEG and JPEG2000 as well as JPEG-LS. JPEG compression steps can be summarized as follows [80, 101]:

1) Image color space conversion: The original image, typically in RGB (Red, Green, Blue) color space, is converted to the YCbCr color space. This color space separates luminance (brightness) information (Y) from chrominance (color) information (Cb and Cr).

2) Chroma component subsampling: The human eye is more sensitive to changes in brightness than color. JPEG takes advantage of this by reducing the resolution of the color channels (Cb and Cr) while keeping the luminance channel (Y) at a higher resolution. Common subsampling ratios are 4:4:4 (no subsampling), 4:2:2, and 4:2:0.

3) Discrete Cosine Transform (DCT): Each color channel is divided into subblocks such as 8x8 block of pixel values. A mathematical transformation called the DCT (in JPEG) or Wavelet Transform (in JPEG2000) is applied to each block, converting it from the spatial domain to the frequency domain. This helps separate the image into different frequency components.

4) Quantization: The transformed coefficients are quantized, meaning they are rounded to a lower precision. This step introduces irreversible loss of information. The quantization process introduces the most significant loss of image quality in JPEG compression. Different quantization tables can be used to control the level of compression and resulting image quality.

5) Entropy Encoding: The quantized coefficients are further compressed using a variable-length coding scheme e.g., Huffman coding (in JPEG). Coefficients that represent common patterns are encoded with shorter codes, while less common coefficients are encoded with longer codes.

2.3.2 GZIP

GZIP [24] is a another widely used data compression algorithm. It was developed by Jean-Loup Gailly and Mark Adler in the early 1990s and is named after the GNU Project, under which it was initially developed. GZIP works by replacing repeated sequences of data with shorter codes, thereby reducing the overall size of the data. It employs the Deflate compression algorithm, which combines two main techniques: Huffman coding and LZ77 (Lempel-Ziv 77) dictionary-based compression. LZ77 compression is a dictionary-based algorithm that replaces repeated sequences of characters with references to a previously encountered occurrence. This reduces redundancy in the data. When a sequence is repeated, instead of storing the entire sequence again, GZIP just stores a reference to the previous occurrence and the

length of the repeated segment. GZIP uses Huffman coding to represent the LZ77 references efficiently. GZIP compression steps can be summarized as follows [24]:

1) LZ77 Compression: The input data is divided into blocks. Then within each block, LZ77 compression is applied to identify repeated sequences.

2) Entropy Encoding: Huffman coding is then used to encode the LZ77 references.

The above given two-step compression is actually called Deflate compression algorithm [23]. GZIP adds a header and trailer to the compressed data to provide information about the compression process and ensure accurate decompression. The header includes details like the compression method used, timestamp, and original file name.

Chapter 3

Diffusion PDEs for Image Processing

Diffusion PDEs are derived by the heat diffusion equation

$$\partial_t u = \operatorname{div}(D \cdot \nabla u) \,, \tag{3.1}$$

which relates temporal changes in a heat concentration $\partial_t u$ to the divergence of its spatial gradient ∇u. When diffusion takes place in an isotropic medium, the diffusivity D is a scalar that determines the rate of heat transfer. In an anisotropic medium, heat spreads out more rapidly in some directions than in others. In those cases, D is a diffusion tensor, i.e., a symmetric matrix that encodes this directional dependence.

When applied to image processing, the gray value at a certain location is interpreted as the heat concentration u (Figure 3.1). In diffusion-based image inpainting, Equation (3.1) is used to propagate information from the known pixels, whose intensity is fixed, to the unknown pixels which will ultimately reach a steady state in which their intensity is determined by their surrounding known pixels. In this sense, Equation (3.1) has a filling-in effect that can be exploited for image compression.

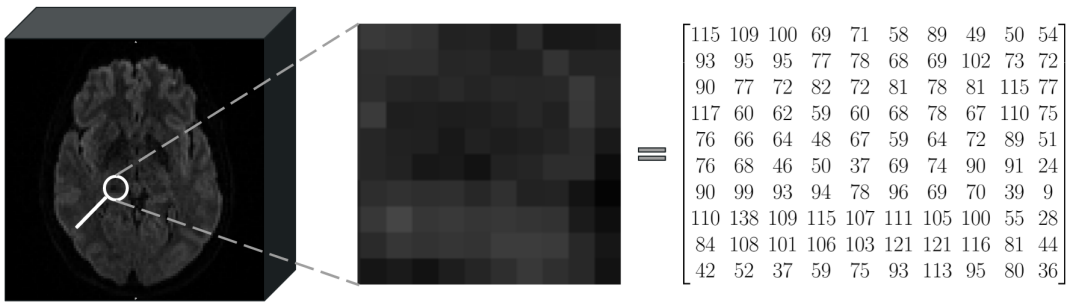


FIGURE 3.1: A discrete image is defined on a finite grid. Patch voxel values visualization by zooming in a DWI volume slice. The voxel values can be represented compactly as a matrix.

Different choices of the diffusivity function D lead to different kinds of diffusion. Linear diffusion [42] and nonlinear diffusion [82] were widely used for image smoothing and image enhancement. Edge structures in images can be enhanced by employing a diffusion tensor which allows diffusion in the direction perpendicular to the local gradient, while applying a nonlinear diffusivity function along the gradient direction. This idea has led to the development of anisotropic nonlinear edge-enhancing diffusion (EED) [110]. Among the six variants that were evaluated

for image compression by Galić et al. [29], EED led to the most accurate reconstructions. Subsequently, this idea was applied to three-dimensional data compression [84], and combined with motion compensation in order to obtain a framework for video compression [5]. When combined with a suitable scheme for selecting and storing the preserved pixels, a few additional optimizations, and at sufficiently high compression rates, anisotropic diffusion has been shown to beat the quality even of JPEG2000 [96].

3.1 Diffusion-based Smoothing

In order to apply Equation (3.1) to image smoothing, we have to restrict it to the image domain Ω, and specify the behavior along its boundary $\partial\Omega$. It is common to assume that no heat is transferred through that boundary (homogeneous Neumann boundary condition). Moreover, the positive real line $(0, \infty)$ is typically taken as the time domain. The resulting PDE can be written as

$$\begin{aligned} \partial_t u &= \operatorname{div}(D \cdot \nabla u), \quad \Omega \times (0, \infty) \,, \\ \partial_n u &= 0, \quad \partial\Omega \times (0, \infty) \,, \end{aligned} \tag{3.2}$$

where n is the normal vector to the boundary $\partial\Omega$. The original image $f : \Omega \to \mathbb{R}$ is used to specify an initial condition $u = f$ at $t = 0$. For increasing diffusion time t, u will correspond to an increasingly smoothed version of the image (see Figure 3.2).

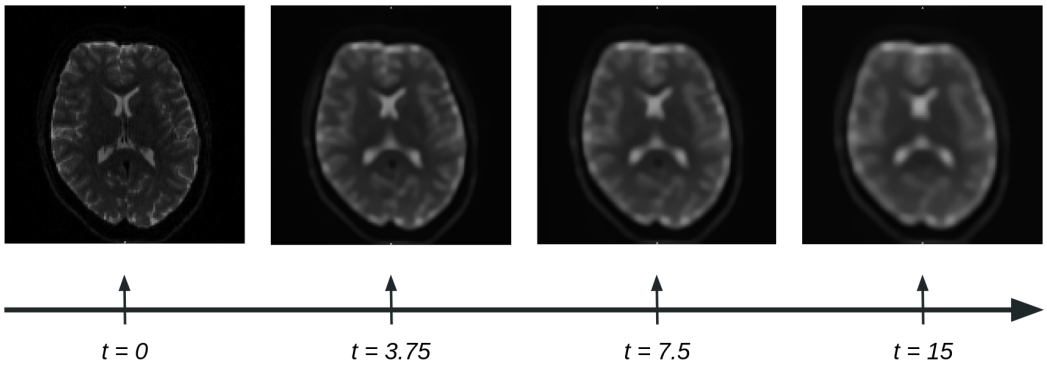


FIGURE 3.2: Axial view of $b = 0$ brain MRI. EED smooothing [110]. Diffusion time t is computed by multiplying time-step size (0.15) to iteration numbers $\{0, 25, 50, 100\}$ respectively.

3.2 Diffusion-based Inpainting

In image inpainting, we know the pixel values on a subset $K \subset \Omega$ of the image, and aim to reconstruct plausible values in the unknown regions. A diffusion-based model for inpainting can be derived from the one for smoothing, by modeling the set of locations at which the pixel values are known with Dirichlet boundary conditions. In this case, $f : K \to \mathbb{R}$ will be used to model the known values. In inpainting-based image compression, K will consist of a small fraction of the pixels in the original

image. With this, we obtain the following model for inpainting:

$$\begin{aligned} \partial_t u &= \operatorname{div}(D \cdot \nabla u), \quad \Omega \backslash K \times (0, \infty) \,, \\ \partial_n u &= 0, \quad \partial\Omega \times (0, \infty) \,, \\ u &= f, \quad K \times [0, \infty) \end{aligned} \tag{3.3}$$

In this case, the diffusion process spreads out the information from the known pixels to their spatial neighborhood. For time $t \to \infty$, image smoothing and inpainting both converge to a steady state, i.e., $\lim_{t\to\infty} \partial_t u = 0$. However, the steady-state of smoothing is trivial (u approaches a constant image with average gray value), while the Dirichlet boundary conditions in the inpainting case ensure a non-trivial steady-state, which is taken as the final inpainting result: $u_{\text{inpainted}} = \lim_{t\to\infty} u$.

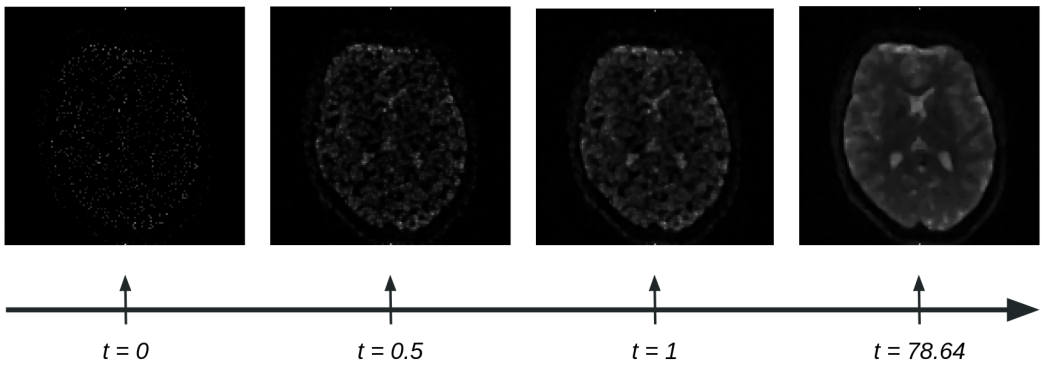


FIGURE 3.3: Axial view of $b = 0$ brain MRI. Fourth-Order EED inpainting [48]. Diffusion time t is computed by multipliying time-step size (0.01) to iteration numbers $\{0, 50, 100, 7864\}$ respectively.

Figure 3.3 shows the timeline for the inpainting process implemented by the explicit numerical scheme (more on numerical schemes will be given in Sec. 3.5). At diffusion time $t = 0$ only a small fraction of pixels are known (randomly chosen 10% of all pixels), the mask set K for Dirichlet boundary conditions. As time goes forward the known information from the set K spreads towards unknown pixels. Finally, at time around $t = 78.64$, the inpainting process reaches its steady state.

3.3 From Linear to Anisotropic Nonlinear Diffusion

So far, we haven't specified the diffusion coefficient D. For example, it can be a scalar and constant, independent from the location within the image. This results in an inpainting model based on linear homogeneous diffusion [42]. With $D = 1$, it can be written as

$$\partial_t u = \Delta u, \quad \Omega \backslash K \times (0, \infty) \,. \tag{3.4}$$

In this and all remaining equations in this section, the same boundary conditions are assumed as specified in Equation (3.3). Despite its simplicity, it has been demonstrated that using this inpainting model for image compression can already beat the JPEG standard when applied to cartoon-like images, and selecting the retained pixels to be close to image edges [70].

When the diffusion coefficient is a scalar but depends on u, i.e., $D = g(u)$, then we call the model inpainting based on nonlinear isotropic diffusion [82]. A common

variant is to make D depend on the local gradient magnitude, i.e.,

$$\partial_t u = \operatorname{div}(g(||\nabla u_\sigma||^2)\nabla u), \quad \Omega \backslash K \times (0,\infty)\,, \tag{3.5}$$

where g is a decreasing nonnegative diffusivity function, e.g., the Charbonnier diffusivity

$$g(s^2) = \frac{1}{\sqrt{1+\frac{s^2}{\lambda^2}}}\,, \tag{3.6}$$

and λ is a contrast parameter separating low from high diffusion areas [17]. In order to localize edges better and to make the problem well-posed, the image is presmoothed with a Gaussian before taking its gradient, i.e., $g(||\nabla u_\sigma||^2)$ is used instead of $g(||\nabla u||^2)$ [15].

In the above-discussed models, the diffusion occurs only in the gradient direction. This can be changed by replacing the scalar diffusivity with a second-order diffusion tensor, i.e., a symmetric positive definite matrix. This is the basis of anisotropic nonlinear diffusion [110],

$$\partial_t u = \operatorname{div}(\mathbf{D} \cdot \nabla u), \quad \Omega \backslash K \times (0,\infty)\,. \tag{3.7}$$

In edge-enhancing diffusion (EED), the diffusion tensor $\mathbf{D}$ is defined as

$$\mathbf{D} = g(||\nabla u_\sigma||^2) \cdot \mathbf{v}_1 \mathbf{v}_1^{\mathrm{T}} + 1 \cdot \mathbf{v}_2 \mathbf{v}_2^{\mathrm{T}}, \tag{3.8}$$

where $\mathbf{v}_1 = \frac{\nabla u_\sigma}{||\nabla u_\sigma||_2}$ and $\mathbf{v}_2 = \frac{\nabla u_\sigma^\perp}{||\nabla u_\sigma||_2}$. This means that diffusion across the edge ($\mathbf{v}_1$) is decreased depending on the gradient magnitude, while diffusion along the edge ($\mathbf{v}_2$) is allowed. Examples of EED based inpainting are included in our original work experimental results section 4.5. In general, EED based inpainting results in better interpolated images than linear homogeneous or nonlinear isotropic PDEs. This makes it a perfect second-order PDE model choice for PDE-based image compression.

3.4 From Second to Fourth Order Diffusion

All models discussed above, as well as several others that have been proposed for inpainting [117], share a common property: They rely on second order PDEs. In image denoising, higher-order PDEs have a long history, going back to work by Scherzer [95]. You and Kaveh [114] propose fourth-order PDEs as a solution to the so-called staircasing problem that arises in edge-enhancing second-order PDEs, such as the filter proposed by Perona and Malik [82]: While the second-order Perona-Malik equation creates visually unpleasant step edges from continuous variations of intensity, corresponding fourth-order methods move these discontinuities into the gradients, where they are less noticeable to the human eye [34]. Subsequently, other fourth-order PDE-based models have been introduced, and have mostly been applied for denoising [36, 37, 67].

For a specific family of higher-order diffusion filters, Didas et al. [25] have shown that, in addition to preserving average gray value, they also preserve higher moments of the initial image. Moreover, depending on the diffusivity function, they can lead to adaptive forward and backward diffusion, and therefore to the enhancement of image features such as curvature. Gorgi Zadeh et al. [115] made use of this property in order to enhance ridges and valleys, by steering fourth-order diffusion

with a fourth-order diffusion tensor. In Chapter 4, we present our work that adapts their method in order to achieve accurate inpainting and reconstruction from a small subset of pixels.

3.5 Numerical Methods for PDEs

It is often the case that the exact solutions of PDEs are either too complicated to determine in closed form or even impossible. Hence the need for reliable and efficient numerical methods and algorithms arises for approximating solutions of PDEs with a computer. Numerical methods provide approximate solutions to these equations by discretizing the domain and time, allowing us to work with discrete data and compute approximate solutions. There exist different methods for solving PDEs numerically. In subsequent subsections, we will briefly discuss two of them that we used to solve PDEs in this dissertation.

3.5.1 Finite Difference Methods

The Finite Difference Methods (FDM) are relatively old but still commonly used in numerical computations due to their simplicity. In FDM, the key idea is to approximate the continuous function u, the analytical solution of the PDEs defined on domain $\Omega \subset R^n$ with a grid function. A grid function is a function that is defined at all points of the discretized version of Ω, the finite-difference grid Ω_h. Partial derivatives of the analytical solution of the PDEs are discretized by difference quotients obtained with help of Taylor expansion. There are three common types of difference quotients (for function $u \in R^3$):

1) Forward Difference: Approximates the derivative at a point by looking forward in the grid. For a function u, the first-order spatial derivative along x is approximated as:

$$\begin{aligned} \partial_x u &= \frac{u_{i+1,j,k} - u_{i,j,k}}{h_x} + \mathcal{O}(h_x) \,, \\ \partial_x u &\approx D^+ u = \frac{u_{i+1,j,k} - u_{i,j,k}}{h_x} \,. \end{aligned} \tag{3.9}$$

2) Backward Difference: Approximates the derivative at a point by looking backward in the grid. For a function u, the first-order spatial derivative along x is approximated as:

$$\begin{aligned} \partial_x u &= \frac{u_{i,j,k} - u_{i-1,j,k}}{h_x} + \mathcal{O}(h_x) \,, \\ \partial_x u &\approx D^- u = \frac{u_{i,j,k} - u_{i-1,j,k}}{h_x} \,. \end{aligned} \tag{3.10}$$

3) Central Difference: Approximates the derivative at a point by considering points on both sides. For a function u, the first-order spatial derivative along x is approximated as:

$$\begin{aligned} \partial_x u &= \frac{u_{i+1,j,k} - u_{i-1,j,k}}{2h_x} + \mathcal{O}(h_x^2) \,, \\ \partial_x u &\approx D^{+,-} u = \frac{u_{i+1,j,k} - u_{i-1,j,k}}{2h_x} \,. \end{aligned} \tag{3.11}$$

where $u_{i,j,k}$ is a 3D discrete image grey value at the voxel location (i, j, k). h_x is the so-called discretization parameter. Equations 3.9 - 3.11 list finite difference approximations for partial derivative along the x-direction for a 3D image. Discretization quotients for derivatives in y- and z-directions can be written similarly. To achieve the discretization formula for mixed derivatives we apply the difference quotients in respective directions one after one.

Above mentioned differences are only for spatial discretization. When we have time-dependent PDEs we need to apply temporal discretization as well. There we have common three different choices as listed in Equation 3.12 in matrix-vector notations.

$$\begin{aligned}
\frac{\mathbf{u}^{k+1}-\mathbf{u}^k}{\tau} &= \mathbf{P}(\mathbf{u}^k)\mathbf{u}^k \ : \textit{Explicit Scheme (forward Euler scheme)} \\
\frac{\mathbf{u}^{k+1}-\mathbf{u}^k}{\tau} &= \mathbf{P}(\mathbf{u}^{k+1})\mathbf{u}^{k+1} \ : \textit{Implicit Scheme (backward Euler scheme)} \\
\frac{\mathbf{u}^{k+1}-\mathbf{u}^k}{\tau} &= \mathbf{P}(\mathbf{u}^k)\mathbf{u}^{k+1} \ : \textit{Semi-Implicit Scheme}
\end{aligned} \tag{3.12}$$

where $\mathbf{u}^k$ is an mn dimensional image vector at iteration (time level) k and m, n are image width and height respectively. τ is a temporal discretization parameter, also called step size. The matrix $\mathbf{P}(\mathbf{u})$ is a spatial discretization of PDEs dependent on image $\mathbf{u}$ for nonlinear PDEs and independent (constant matrix $\mathbf{P}$) for linear PDEs. In the explicit Euler scheme, $\mathbf{u}^{k+1}$ can be calculated explicitly from $\mathbf{u}^k$ without needing to solve a system of equations which is not the case in implicit and semi-implicit schemes. This simple nature makes the explicit scheme computationally very cheap in each time step iteration, in computing $\mathbf{u}^{k+1}$. But in the whole numerical process, e.g. in reaching to the steady state in inpainting PDEs (Equation 3.3) explicit scheme is very slow as it can have a small time step size due to numerical stability reasons [111, 115]. The explicit scheme is conditionally stable. More specifically, the numerical scheme $\mathbf{u}^{k+1} = (\mathbf{I} + \tau\mathbf{P}(\mathbf{u}^k))\mathbf{u}^k$ has to satisfy the condition $||\mathbf{u}^{k+1}||_2 \leq ||\mathbf{u}^k||_2$, i.e. the L_2 stability. We can guarantee that by requiring the spectral norm of the system matrix $\mathbf{I} + \tau\mathbf{P}(\mathbf{u}^k)$ to be bounded by 1, i.e., $\rho(\mathbf{I} + \tau\mathbf{P}(\mathbf{u}^k)) \leq 1$ [111]. For a negative semi-definite $\mathbf{P}(\mathbf{u}^k)$, the condition boils down to $\tau \leq \frac{2}{\rho(\mathbf{P}(\mathbf{u}^k))}$; $\rho(\mathbf{P}(\mathbf{u}^k))$ can be estimated via Gershgorin's Theorem.

The computational inefficiency of the explicit scheme is solved by Hafner et al. [35]. They propose a so-called Fast Semi-Iterative Scheme (FSI) which extrapolates the basic solver iteration with the previous iterate and serves as an accelerated explicit scheme. The acceleration of the explicit scheme is given as

$$\mathbf{u}^{m,k+1} = \alpha_k \cdot (\mathbf{I} + \tau\mathbf{P}(\mathbf{u}^{m,k}))\mathbf{u}^{m,k} + (1-\alpha_k) \cdot \mathbf{u}^{m,k-1}\,, \tag{3.13}$$

where $\mathbf{u}^{m,-1} := \mathbf{u}^{m,0}$ and $\alpha_k = \frac{4k+2}{2k+3}$ for $k = 0, \cdots, n-1$. Here m stands for outer cycle, i.e. m-th cycle with inner cycle of length n. And for passing to the next outer cycle, we set $\mathbf{u}^{m+1,0} := \mathbf{u}^{m,n}$.

By replacing the derivatives in the original PDE with their finite difference approximations, the problem results in a system of algebraic equations that can be solved numerically. Boundary conditions and initial conditions should be incorporated into the system of equations. Boundary conditions may be Dirichlet, Neumann, or mixed conditions, depending on the problem.

FDM is a versatile and widely used technique for solving a variety of PDEs, including heat conduction, wave propagation, and diffusion problems. However, it is important to note that the accuracy and stability of FDM depend on the choice of grid spacing and time step size, and more complex geometries may require specialized techniques or adaptations like staggered grids or higher-order schemes. We used FDM for solving PDEs defined on 2D and 3D spatial image domains that are rectangle and cuboid respectively.

3.5.2 Finite Element Methods

FDM are usually easy to implement on a regular domain dicretized to equidistant grid. For complex domains, Finite Element Methods (FEM) are more suitable. In the FEM, the central idea is to look for the approximate solution of the PDEs, u_h in a chosen finite-dimensional vector space V_h:

$$u_h = \sum_{i=0}^{N-1} U_i \phi_i \,, \tag{3.14}$$

where $\{\phi_i\}_{i=0}^{N-1}$ is a basis of finite vector space V_h. The solution u_h is determined by U_i coefficients and ϕ_i basis functions. It also depends on $\{\hat{\phi}_i\}_{i=0}^{N-1}$ which is a basis of another vector space $\hat{V}_h$. If we assume that we know V_h, $\hat{V}_h$ and their basis functions, then solving PDEs with FEM is as simple as solving a linear system:

$$AU = b \,, \tag{3.15}$$

where $A_{ij} = a(\hat{\phi}_i, \phi_j)$, $U \in R^N$ is the vector of coefficients $\{U_j\}_{j=0}^{N-1}$ and $b_i = L(\hat{\phi}_i)$; $a(\cdot,\cdot)$ and $L(\cdot)$ are bilinear and linear forms respectively. Before the problem boils down to a linear system of equations, there are two main steps in FEM. In subsection, we give a big picture of those FEM steps and refer the reader to [12] for a detailed theoretical discussion and to [58, 66] for practical implementation with the package called FEniCS. Two main steps in FEM:

- *Switch from strong to weak formulation:* The PDEs formulation with boundary conditions is called a strong formulation as it states the conditions (e.g. differentiability) at every point over a domain that a solution must satisfy. On the other hand, a weak formulation (also known as variational formulation) states the conditions that must be met only in an average sense, i.e., in an integral sense. A general variational formulation can be written as following:

  $$\text{Find } u \in V \text{ such that } a(u,v) = L(v),\ \forall v \in \hat{V} \,, \tag{3.16}$$

  where $a(\cdot,\cdot)$ and $L(\cdot)$ are bilinear and linear forms respectively. The (infinite dimensional) function spaces, trial space V and test space $\hat{V}$ are chosen such that the problem makes sense for $u \in V, v \in \hat{V}$. The exact definitions of V and $\hat{V}$ depend on the PDEs being solved. The well-posedness of the weak problem (3.16) is verified by the Lax-Milgram theorem that lists sufficient conditions on bilinear a and linear L forms for a variational problem to be well-posed, i.e., existence and uniqueness of a solution and its stability.

- *Discretize weak formulation:* To solve numerically, we need to discretize the continuous variational problem (3.16) into a discrete variational problem by restricting it to a pair of discrete spaces:

$$\text{Find } u_h \in V_h \subset V \text{ such that } a(u_h, v_h) = L(v_h),\ \forall v_h \in \hat{V}_h \subset V_h\ , \tag{3.17}$$

  where V_h and $\hat{V}_h$ are closed finite-dimensional subspaces of V and $\hat{V}$ respectively. This discretization is also known as a Galerkin approximation of the problem (3.16) over subspaces V_h and $\hat{V}_h$. The same theorem, the Lax-Milgram theorem studies the well-posedness of the discrete Galerkin approximation (3.17). By using the ansatz (3.5.2) for the approximate solution u_h in terms of the trial space basis functions we can rewrite the Galerkin approximation (3.17) as the linear system (3.15) given above. From the linear system (3.15), it is easy to see that the discretization (3.17) preserves the structural properties of the continuous problem (3.16). In particular, if the bilinear form, a is symmetric, it is reflected in the discrete linear problem (3.17) as A being a symmetric matrix. The same is true for coercivity of a that suffices A to be a positive-definite. The Galerkin approximation error bound is studied by Céa's lemma. It formulates the error in terms of some problem-specific constants.

FEM is a versatile and widely used numerical technique, especially in structural analysis, heat transfer, fluid dynamics, electromagnetics, and many other fields. Its flexibility in handling complex geometries and boundary conditions, as well as its ability to adaptively refine the mesh for improved accuracy, makes it a powerful tool for solving PDEs in practical engineering and scientific applications. We employed FEM for solving linear inpainting PDEs in q-space, i.e., on the surface of a sphere (see Section 6.4.2).

Chapter 4

Fourth-Order Anisotropic Diffusion for Inpainting and Image Compression

One of the main tasks in digital image processing is image inpainting which is widely used in image restoration, image manipulation, and the restoration of damaged or historical images. Image inpainting is a process of filling in missing or corrupted parts of an image with plausible content, making the inpainted regions visually coherent and consistent with the surrounding areas. The inpainting methods can be categorized into two main groups such as exemplar-based methods and propagation-based methods. The exemplar-based methods use information from the surrounding areas of the missing region to synthesize a replacement. This involves finding similar patches or structures in the available parts of the image and using them to reconstruct the missing portion. The latter one, propagation-based methods propagate information from the known parts of the image to the unknown or damaged regions. This can involve diffusion-based techniques or solving partial differential equations to extend known information into the missing areas. Our focus in this dissertation is propagation-based methods.

Diffusion-based inpainting involves using partial differential equations (PDEs) like the heat equation to propagate information from available parts of the image to the unavailable regions. The diffusion process helps to estimate the missing pixel values based on the neighboring pixels and their intensities. It is done by interpreting the physical diffusion process as treating the known image values as the initial condition and allowing the diffusion process to fill in the unknown regions. The change in time is formulated by spatial variations which basically explains how gradually filling in the missing regions happens.

Great ability of diffusion PDEs in image inpainting led them to be used in image compression, particularly in lossy compression schemes. Inpainting PDEs which has great ability to fill-in missing information allowed to throw away a substantial fraction of pixels in an image. This made it possible to reduce the amount of data needed to represent an image while attempting to maintain essential visual information.

In this chapter we discuss our proposed PDE-based inpainting method. It is a fourth-order generalization of second-order edge-enhancing diffusion PDEs exploited for image inpainting and image compression. Our proposed fourth-order diffusion PDEs method employs a fourth-order diffusion tensor. The diffusion tensor is designed from the regularized local image gradient information. The fourth-order diffusion tensor takes the anisotropy of edge-enhancing diffusion PDEs method to a higher level. We provide a theoretical unifying framework to define all anisotropic fourth-order diffusion based methods which have been proposed so far.

In the implementation of our proposed method we use a fast semi-iterative scheme.

To establish the usefulness of our proposed new model, we performed three different set of experiments: 1) Inpainting of images from a sparse subset of pixels. This experiment shows the usefulness of our method in image compression. 2) Inpainting of images when most pixels are known. This experiment shows performance of our method in classical inpainting tasks such as scratch removal. 3) In our third experiment set, we demonstrate how results depend on the chosen diffusivity function and contrast parameter. The diffusivity function and contrast parameters are the main ingredients of the diffusion tensor. They define in which direction and with which magnitude anisotropic diffusion occurs. Our experimental results demonstrated that our proposed new method achieved improved accuracy in all above-mentioned experiment sets.

The content of this chapter is published as a peer reviewed book chapter:

Jumakulyyev Ikram, Schultz Thomas "Fourth-order anisotropic diffusion for inpainting and image compression" In: *Anisotropy Across Fields and Scales.* Springer, Cham, 2021, pp. 99-124. [47].

Contribution of the dissertation author: Conceptualization, methodology, literature review, algorithm implementation and validation, results interpretation and visualization, manuscript composition, validation experiment design.

4.1 Abstract

Edge-enhancing diffusion (EED) can reconstruct a close approximation of an original image from a small subset of its pixels. This makes it an attractive foundation for PDE based image compression. In this work, we generalize second-order EED to a fourth-order counterpart. It involves a fourth-order diffusion tensor that is constructed from the regularized image gradient in a similar way as in traditional second-order EED, permitting diffusion along edges, while applying a non-linear diffusivity function across them. We show that our fourth-order diffusion tensor formalism provides a unifying framework for all previous anisotropic fourth-order diffusion based methods, and that it provides additional flexibility. We achieve an efficient implementation using a fast semi-iterative scheme. Experimental results on natural and medical images suggest that our novel fourth-order method produces more accurate reconstructions compared to the existing second-order EED.

4.2 Introduction

The increased availability and resolution of imaging technology, including digital cameras and medical imaging devices, along with advances in storage capacity and transfer bandwidths, have led to a proliferation of large image data. This makes image compression an important area of research. Image compression techniques can be divided into two main groups: Lossy and lossless compression. Lossless compression techniques permit restoration of the full, unmodified image data, which however limits the achievable compression rates. Our work is concerned with lossy compression, which achieves much higher compression rates by replacing the original image with an approximation that can be stored more efficiently.

We continue a line of research that has explored the use of Partial Differential Equations (PDEs) for lossy image compression [29, 30, 70, 97]. This approach is based on storing only a small subset of all pixels, and interpolating between them

in order to restore the remaining ones. There is a strong similarity between that interpolation process and image inpainting, whose goal it is to reconstruct missing or corrupted parts of an image. PDE-based methods for image inpainting and compression are inspired by the physical phenomenon of heat transport. It is described by the heat diffusion equation

$$\partial_t u = \operatorname{div}(D \cdot \nabla u) \,, \tag{4.1}$$

which relates temporal changes in a heat concentration $\partial_t u$ to the divergence of its spatial gradient ∇u. When diffusion takes place in an isotropic medium, the diffusivity D is a scalar that determines the rate of heat transfer. In an anisotropic medium, heat spreads out more rapidly in some directions than in others. In those cases, D is a diffusion tensor, i.e., a symmetric matrix that encodes this directional dependence.

When applied to image processing, the gray value at a certain location is interpreted as the heat concentration u. In diffusion-based image inpainting, Equation (6.1) is used to propagate information from the known pixels, whose intensity is fixed, to the unknown pixels which will ultimately reach a steady state in which their intensity is determined by their surrounding known pixels. In this sense, Equation (6.1) has a filling-in effect that can be exploited for image compression.

Different choices of the diffusivity function D lead to different kinds of diffusion. Linear diffusion [42] and nonlinear diffusion [82] were widely used for image smoothing and image enhancement. Edge structures in images can be enhanced by employing a diffusion tensor which allows diffusion in the direction perpendicular to the local gradient, while applying a nonlinear diffusivity function along the gradient direction. This idea has led to the development of anisotropic nonlinear edge-enhancing diffusion (EED) [110]. Among the six variants that were evaluated for image compression by Galić et al. [29], EED led to the most accurate reconstructions. Subsequently, this idea was applied to three-dimensional data compression [84], and combined with motion compensation in order to obtain a framework for video compression [5]. When combined with a suitable scheme for selecting and storing the preserved pixels, a few additional optimizations, and at sufficiently high compression rates, anisotropic diffusion has been shown to beat the quality even of JPEG2000 [96].

In this paper, we introduce a novel fourth-order PDE that generalizes second-order EED, and achieves even more accurate reconstructions. We build on prior works that proposed fourth-order analogs of the diffusion equation, and used them for image processing [25, 37, 62, 67, 95, 114]. In particular, we extend a work by Gorgi Zadeh et al. [115], who introduced the idea of steering anisotropic fourth-order diffusion with a fourth-order diffusion tensor. However, their method focuses on the curvature enhancement property of nonlinear fourth-order diffusion [25] in order to better localize ridge and valley structures. Deriving a suitable PDE for image inpainting requires a different definition of the diffusion tensor, more similar to the one in edge-enhancing diffusion [110]. Two anisotropic fourth-order PDEs for inpainting were previously introduced by Li et al. [62]. However, they only apply them to image restoration tasks in which small parts of an image are missing (such as in Figure 4.8), not to the reconstruction from a small subset of pixels. Moreover, we demonstrate that the fourth-order diffusion tensor based framework is more general in the sense that it can be used to express anisotropic fourth-order diffusion as it was described by Hajiaboli [37] or by Li et al. [62], while providing additional flexibility.

4.3 Background and Related Work

We will now formalize the above-mentioned idea of diffusion-based inpainting (Section 4.3.1), and review two concepts that play a central role in our method: Anisotropic nonlinear diffusion (Section 4.3.2) and fourth-order diffusion (Section 4.3.3). Further details can be found in works by Galić et al. [29] and Weickert [110], respectively. Finally, we provide additional context with a brief discussion of alternative approaches to image compression (Section 4.3.4).

4.3.1 Diffusion-based Inpainting

In order to apply Equation (6.1) to image smoothing, we have to restrict it to the image domain Ω, and specify the behavior along its boundary $\partial\Omega$. It is common to assume that no heat is transferred through that boundary (homogeneous Neumann boundary condition). Moreover, the positive real line $(0, \infty)$ is typically taken as the time domain. The resulting PDE can be written as

$$\begin{aligned} \partial_t u &= \operatorname{div}(D \cdot \nabla u), \quad \Omega \times (0, \infty) \,, \\ \partial_n u &= 0, \quad \partial\Omega \times (0, \infty) \,, \end{aligned} \tag{4.2}$$

where n is the normal vector to the boundary $\partial\Omega$. The original image $f : \Omega \to \mathbb{R}$ is used to specify an initial condition $u = f$ at $t = 0$. For increasing diffusion time t, u will correspond to an increasingly smoothed version of the image.

In image inpainting, we know the pixel values on a subset $K \subset \Omega$ of the image, and aim to reconstruct plausible values in the unknown regions. A diffusion-based model for inpainting can be derived from the one for smoothing, by modeling the set of locations at which the pixel values are known with Dirichlet boundary conditions. In this case, $f : K \to \mathbb{R}$ will be used to model the known values. In inpainting-based image compression, K will consist of a small fraction of the pixels in the original image. With this, we obtain the following model for inpainting:

$$\begin{aligned} \partial_t u &= \operatorname{div}(D \cdot \nabla u), \quad \Omega \backslash K \times (0, \infty) \,, \\ \partial_n u &= 0, \quad \partial\Omega \times (0, \infty) \,, \\ u &= f, \quad K \times [0, \infty) \end{aligned} \tag{4.3}$$

In this case, the diffusion process spreads out the information from the known pixels to their spatial neighborhood. For time $t \to \infty$, image smoothing and inpainting both converge to a steady state, i.e., $\lim_{t\to\infty} \partial_t u = 0$. However, the steady-state of smoothing is trivial (u approaches a constant image with average gray value), while the Dirichlet boundary conditions in the inpainting case ensure a non-trivial steady-state, which is taken as the final inpainting result: $u_{\text{inpainted}} = \lim_{t\to\infty} u$

4.3.2 From Linear to Anisotropic Nonlinear Diffusion

So far, we assumed that the diffusion coefficient D is a scalar and constant, independent from the location within the image. This results in an inpainting model based on linear homogeneous diffusion [42]. With $D = 1$, it can be written as

$$\partial_t u = \Delta u, \quad \Omega \backslash K \times (0, \infty) \,. \tag{4.4}$$

In this and all remaining equations in this section, the same boundary conditions are assumed as specified in Equation (6.3). Despite its simplicity, it has been demonstrated that using this inpainting model for image compression can already beat the JPEG standard when applied to cartoon-like images, and selecting the retained pixels to be close to image edges [70].

When the diffusion coefficient is a scalar but depends on u, i.e., $D = g(u)$, then we call the model inpainting based on nonlinear isotropic diffusion [82]. A common variant is to make D depend on the local gradient magnitude, i.e.,

$$\partial_t u = \operatorname{div}(g(||\nabla u_\sigma||^2)\nabla u), \quad \Omega\backslash K \times (0,\infty)\,, \tag{4.5}$$

where g is a decreasing nonnegative diffusivity function, e.g., the Charbonnier diffusivity

$$g(s^2) = \frac{1}{\sqrt{1+\frac{s^2}{\lambda^2}}}\,, \tag{4.6}$$

and λ is a contrast parameter separating low from high diffusion areas [17]. In order to localize edges better and to make the problem well-posed, the image is presmoothed with a Gaussian before taking its gradient, i.e., $g(||\nabla u_\sigma||^2)$ is used instead of $g(||\nabla u||^2)$ [15].

In the above-discussed models, the diffusion occurs only in the gradient direction. This can be changed by replacing the scalar diffusivity with a second-order diffusion tensor, i.e., a symmetric positive definite matrix. This is the basis of anisotropic nonlinear diffusion [110],

$$\partial_t u = \operatorname{div}(\mathbf{D}\cdot\nabla u), \quad \Omega\backslash K \times (0,\infty)\,. \tag{4.7}$$

In edge-enhancing diffusion (EED), the diffusion tensor $\mathbf{D}$ is defined as

$$\mathbf{D} = g(||\nabla u_\sigma||^2)\cdot \mathbf{v}_1\mathbf{v}_1^{\mathrm{T}} + 1\cdot \mathbf{v}_2\mathbf{v}_2^{\mathrm{T}}, \tag{4.8}$$

where $\mathbf{v}_1 = \frac{\nabla u_\sigma}{||\nabla u_\sigma||_2}$ and $\mathbf{v}_2 = \frac{\nabla u_\sigma^\perp}{||\nabla u_\sigma||_2}$. This means that diffusion across the edge ($\mathbf{v}_1$) is decreased depending on the gradient magnitude, while diffusion along the edge ($\mathbf{v}_2$) is allowed. Examples of EED based inpainting are included in our experimental results. In general, EED based inpainting results in better interpolated images than linear homogeneous or nonlinear isotropic PDEs. This makes it a current state-of-the-art choice for PDE-based image compression.

4.3.3 From Second to Fourth Order Diffusion

All models discussed above, as well as several others that have been proposed for inpainting [117], share a common property: They rely on second order PDEs. In image denoising, higher-order PDEs have a long history, going back to work by Scherzer [95]. You and Kaveh [114] propose fourth-order PDEs as a solution to the so-called staircasing problem that arises in edge-enhancing second-order PDEs, such as the filter proposed by Perona and Malik [82]: While the second-order Perona-Malik equation creates visually unpleasant step edges from continuous variations of intensity, corresponding fourth-order methods move these discontinuities into the gradients, where they are less noticeable to the human eye [34]. Subsequently, other fourth-order PDE-based models have been introduced, and have mostly been applied for denoising [36, 37, 67].

For a specific family of higher-order diffusion filters, Didas et al. [25] have shown that, in addition to preserving average gray value, they also preserve higher moments of the initial image. Moreover, depending on the diffusivity function, they can lead to adaptive forward and backward diffusion, and therefore to the enhancement of image features such as curvature. Gorgi Zadeh et al. [115] made use of this property in order to enhance ridges and valleys, by steering fourth-order diffusion with a fourth-order diffusion tensor. Our work adapts their method in order to achieve accurate inpainting and reconstruction from a small subset of pixels.

4.3.4 Alternative Approaches to Image Compression

The dominant lossy image compression techniques today are JPEG and JPEG2000. They are based on the discrete cosine transform (DCT) and wavelet transform, respectively. However, they are not sensitive to the geometry of an image, i.e. those standards are not tailored to their geometrical behavior [13]. Especially, the JPEG standard involves dividing the image into small square blocks. This can cause a degradation called "blocking effect" [102], and can result in unsatisfactory reconstructions especially at high compression rates.

It is an ongoing research trend to apply machine learning methods to image compression, such as convolutional and recurrent neural networks [6, 103, 106]. Learning based approaches tend to perform very well on the specific class of images on which they were trained, but require a huge amount of data. For example, Toderici et al. [107] used for training a dataset of 6 million 1280×720 images taken from the web.

4.4 Method

We will now introduce our novel PDE (Section 4.4.1), investigate its relationship to previously proposed anisotropic fourth-order diffusion (Section 4.4.2), and comment on our chosen discretization, as well as numerical stability (Section 4.4.3).

4.4.1 Anisotropic Edge-Enhancing Fourth Order PDE

Our fourth-order PDE builds on a model that was proposed by Gorgi Zadeh et al. [115] for ridge and valley enhancement. It can be stated concisely using Einstein notation, where summation is implied for indices appearing twice in the same expression:

$$\partial_t u = -\partial_{ji} \left[\mathcal{D}(\mathbf{H}_\rho(u_\sigma)) : \mathbf{H}(u) \right]_{ij} \tag{4.9}$$

In this equation, $\mathbf{H}(u)$ denotes the Hessian matrix of image u. The "double dot product" $\mathbf{T} = \mathcal{D} : \mathbf{H}$ indicates that matrix $\mathbf{T}$ is obtained by applying a linear map $\mathcal{D}$ to $\mathbf{H}$, and the square bracket notation $[\mathbf{T}]_{ij}$ indicates taking the (i,j)th component:

$$[\mathbf{T}]_{ij} = \left[\mathcal{D}(\mathbf{H}_\rho(u_\sigma)) : \mathbf{H}(u) \right]_{ij} = \left[\mathcal{D}(\mathbf{H}_\rho(u_\sigma)) \right]_{ijkl} [\mathbf{H}(u)]_{kl} \tag{4.10}$$

Since $\mathcal{D}$ maps matrices to matrices, it is a fourth-order tensor. Since its role is analogous to the second-order diffusion tensor in Equation (4.7), it is referred to as a fourth-order diffusion tensor.

The diffusion tensor $\mathcal{D}$ in Equation (4.9) is a function of the local normalized Hessian $\mathbf{H}_\rho(u_\sigma)$ which contains the information that is relevant to achieve curvature enhancement. For image inpainting, we propose to instead steer the fourth-order

diffusion in analogy to edge-enhancing diffusion, i.e., as a function of the structure tensor $\mathbf{J}(u_\sigma)$, which is obtained from image u after Gaussian pre-smoothing with bandwidth σ. We construct our fourth order diffusion tensor $\mathcal{D}$ from its eigenvalues μ_i and eigentensors $\mathbf{E}_i$ via the spectral decomposition:

$$\mathcal{D}\left(\mathbf{J}(u_\sigma)\right) = \mu_1 \mathbf{E}_1 \otimes \mathbf{E}_1 + \mu_2 \mathbf{E}_2 \otimes \mathbf{E}_2 + \mu_3 \mathbf{E}_3 \otimes \mathbf{E}_3 + \mu_4 \mathbf{E}_4 \otimes \mathbf{E}_4 \tag{4.11}$$

The eigenvalues and eigentensors are defined as

$$\begin{aligned}
&\mu_1 = g(\lambda_1),\ \mathbf{E}_1 = \mathbf{v}_1 \otimes \mathbf{v}_1\ , \\
&\mu_2 = 1,\ \mathbf{E}_2 = \mathbf{v}_2 \otimes \mathbf{v}_2\ , \\
&\mu_3 = \sqrt{g(\lambda_1)},\ \mathbf{E}_3 = \frac{1}{\sqrt{2}}(\mathbf{v}_1 \otimes \mathbf{v}_2 + \mathbf{v}_2 \otimes \mathbf{v}_1)\ , \\
&\mu_4 = 0,\ \mathbf{E}_4 = \frac{1}{\sqrt{2}}(\mathbf{v}_1 \otimes \mathbf{v}_2 - \mathbf{v}_2 \otimes \mathbf{v}_1)\ ,
\end{aligned} \tag{4.12}$$

where g is a nonnegative decreasing diffusivity function, λ_i and $\mathbf{v}_i$ are eigenvalues and eigenvectors of the structure tensor $\mathbf{J}(u_\sigma) = \nabla u_\sigma \nabla u_\sigma^{\mathrm{T}}$, i.e., $\lambda_1 = ||\nabla u_\sigma||_2^2$, $\mathbf{v}_1 = \frac{\nabla u_\sigma}{||\nabla u_\sigma||_2}$ and $\lambda_2 = 0$, $\mathbf{v}_2 = \frac{\nabla u_\sigma^\perp}{||\nabla u_\sigma||_2}$. The above-defined eigentensors are orthonormal with respect to the dot product $\mathbf{A} : \mathbf{B} = \mathrm{trace}(\mathbf{B}^{\mathrm{T}}\mathbf{A})$ [115].

Combining this new definition of the fourth-order diffusion tensor with Dirichlet boundary conditions as in Equation (6.3) results in our proposed model:

$$\begin{aligned}
\partial_t u &= -\partial_{ji}[\mathcal{D}(\mathbf{J}(u_\sigma)) : \mathbf{H}(u)]_{ij}, \quad \Omega \backslash K \times (0, \infty)\ , \\
u &= f, \quad K \times [0, \infty)
\end{aligned} \tag{4.13}$$

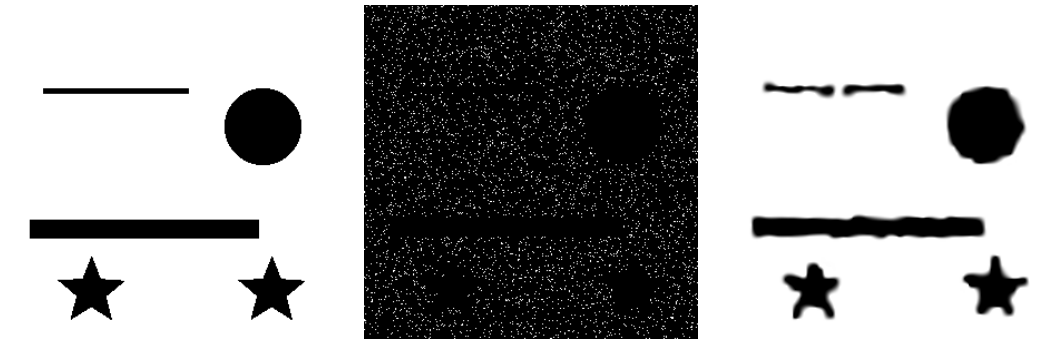
As it is customary in PDE-based inpainting, we allow Equation (4.13) to evolve until a steady state has been reached, i.e., the time derivative becomes negligible. In our numerical implementation, we use a Fast Semi-Iterative Scheme (FSI) [35] to greatly accelerate convergence to a large stopping time.

In the definition of our fourth-order diffusion tensor $\mathcal{D}$, the choice of μ_1 and μ_2 is analogous to anisotropic edge enhancing diffusion [110]. However, two additional terms occur in the fourth-order case, μ_3 and μ_4. As noted in [115], μ_4 is irrelevant, since the corresponding eigentensor $\mathbf{E}_4$ is anti-symmetric, and the dot product $\mathbf{E}_4 : \mathbf{H}$ with the Hessian of any sufficiently smooth image will be zero due to its symmetry. To better understand the role of μ_3, we observe that

$$\begin{aligned}
\mathbf{E}_3 : \mathbf{H} &= \frac{1}{\sqrt{2}}\left(\mathbf{v}_1^{\mathrm{T}}\mathbf{H}\mathbf{v}_2 + \mathbf{v}_2^{\mathrm{T}}\mathbf{H}\mathbf{v}_1\right) \\
&= \frac{1}{\sqrt{2}}\left(u_{\left(\frac{\mathbf{v}_1+\mathbf{v}_2}{\sqrt{2}}\right)\left(\frac{\mathbf{v}_1+\mathbf{v}_2}{\sqrt{2}}\right)} - u_{\left(\frac{\mathbf{v}_1-\mathbf{v}_2}{\sqrt{2}}\right)\left(\frac{\mathbf{v}_1-\mathbf{v}_2}{\sqrt{2}}\right)}\right)\ ,
\end{aligned} \tag{4.14}$$

which amounts to a mixed second derivative of u, in directions along and orthogonal to the regularized image gradient $\frac{\nabla u_\sigma}{||\nabla u_\sigma||_2}$ or, equivalently, to the difference of second derivatives in the directions that are exactly in between the two. This term vanishes if the Hessian is isotropic, or if the gradient is parallel to one of the Hessian eigenvectors. Therefore, the role of μ_3 can be seen as steering the amount of diffusion in cases of a Hessian anisotropy that goes along with a misalignment between gradient and Hessian eigenvectors.

Gorgi Zadeh et al. [115] simply set μ_3 to the arithmetic mean of μ_1 and μ_2. In our work, we empirically evaluated several alternative options for μ_3 by reconstructing



(a) Original test image of size 300×300

(b) Randomly chosen 5% of pixel values

(c) Second-order EED inpainting based on (b)

(d) Fourth-order EED inpainting with $\mu_3 = 1$

(e) Fourth-order EED inpainting with $\mu_3 = \frac{\mu_1+\mu_2}{2}$

(f) Fourth-order EED inpainting with $\mu_3 = \sqrt{\mu_1\mu_2}$

FIGURE 4.1: Reconstruction of a synthetic test image (a) from 5% of its pixels (b) based on second-order diffusion (c) and fourth-order diffusion with different coefficients for the mixed term μ_3 (d–f). Visually, the reconstruction in (f) is most similar to the original image.

TABLE 4.1: Numerical reconstruction errors on the test image (Figure 4.1)

Errors	EED	FOEED ($\mu_3 = 1$)	FOEED ($\mu_3 = \frac{\mu_1+\mu_2}{2}$)	FOEED ($\mu_3 = \sqrt{\mu_1\mu_2}$)
MSE	647.183	660.588	634.321	533.987
AAE	5.043	4.581	4.505	4.140

the test image shown in Figure 4.1 (a), which contains one rectangle, one circle, and two stars, from a randomly selected subset of 5% of its pixels. In this experiment, we compare EED based inpainting with our novel fourth-order edge enhancing diffusion (FOEED) with different settings of μ_3: Specifically, $\mu_3 = 1$ corresponds to the maximum of μ_1 and μ_2 (Figure 4.1 (d)), $\mu_3 = (1 + g(\lambda_1))/2$ corresponds to their arithmetic mean (Figure 4.1 (e)), and $\mu_3 = \sqrt{g(\lambda_1)}$ corresponds to their geometric mean (Figure 4.1 (f)). In all cases, we used the Charbonnier diffusivity (Equation (5.2)), which is popular for image compression [29], the same contrast parameter ($\lambda = 0.1$) and smoothing parameter ($\sigma = 1$). The only difference is time step size, where second-order EED permitted a stable step size of 0.25, while a smaller step size of 0.05 was chosen for FOEED. A more detailed theoretical and empirical analysis of stability will be given in Section 4.4.3.

A numerical comparison of the results is given in Table 4.1. For evaluation, we used the well-known mean squared error (MSE) and average absolute error (AAE) between original and reconstructed images. For two-dimensional gray-valued images u and v with the same dimensions $m \times n$, the MSE and AAE are defined as

$$\begin{aligned} \mathrm{MSE}(u,v) &= \frac{1}{mn}\sum_{i,j}(u_{i,j} - v_{i,j})^2 \,, \\ \mathrm{AAE}(u,v) &= \frac{1}{mn}\sum_{i,j}|u_{i,j} - v_{i,j}| \,. \end{aligned} \tag{4.15}$$

According to Table 4.1, the most accurate results are achieved by setting μ_3 to the geometric mean of μ_1 and μ_2. Fourth-order EED with this setting produces higher accuracy than second-order EED. Visually, Figure 4.1 supports this conclusion. Specifically, fourth-order EED with $\mu_3 = \sqrt{\mu_1\mu_2}$ is the only variant that correctly connects the thin bar at the top of the test image, and it leads to a straighter shape of the thicker bar below, which is more similar to its original rectangular shape. In all subsequent experiments, we set $\mu_3 = \sqrt{\mu_1\mu_2}$.

4.4.2 A Unifying Framework for Fourth-Order Diffusion

Several fourth-order diffusion PDEs have been used for image processing previously. We can better understand how they relate to our newly proposed PDE by observing that the fourth-order diffusion tensor $\mathcal{D}$ introduces a unifying framework for fourth-order diffusion filters. In particular, given its coefficients d_{ijkl}, we can expand Equation (4.9) by using Einstein notation as

$$\partial_t u = -\partial_{ji}[d_{ijkl}u_{kl}] \tag{4.16}$$

Effectively, the fourth-order diffusion tensor allows us to separately set the diffusivities for all $2^4 = 16$ fourth-order derivatives of the two-dimensional image u. We will now demonstrate how several well-known fourth-order PDEs can be expressed

in this framework, starting with the You-Kaveh PDE [114]

$$\partial_t u = -\Delta(g(|\Delta u|)\Delta u)\,, \tag{4.17}$$

which can be rewritten as

$$\begin{aligned}\partial_t u = &-\partial_{xx}[g(|\Delta u|)u_{xx} + 0\cdot u_{xy} + 0\cdot u_{yx} + g(|\Delta u|)u_{yy}]\\ &-\partial_{yx}[0\cdot u_{xx} + 0\cdot u_{xy} + 0\cdot u_{yx} + 0\cdot u_{yy}]\\ &-\partial_{xy}[0\cdot u_{xx} + 0\cdot u_{xy} + 0\cdot u_{yx} + 0\cdot u_{yy}]\\ &-\partial_{yy}[g(|\Delta u|)u_{xx} + 0\cdot u_{xy} + 0\cdot u_{yx} + g(|\Delta u|)u_{yy}]\,.\end{aligned} \tag{4.18}$$

Here and in all subsequent examples, many terms have zero coefficients. For brevity, we will omit them from now on.

Hajiaboli's anisotropic fourth-order PDE [37] is

$$\partial_t u = -\Delta\left(g(||\nabla u||)^2 u_{NN} + g(||\nabla u||)u_{TT}\right)\,, \tag{4.19}$$

where N and T are unit vectors parallel and orthogonal to the gradient, respectively. It can be rewritten as

$$\begin{aligned}\partial_t u = &-\partial_{xx}\left[\left(\frac{g(||\nabla u||)^2 u_x^2 + g(||\nabla u||)u_y^2}{u_x^2+u_y^2}\right)u_{xx} + \left(\frac{g(||\nabla u||)^2 u_x u_y - g(||\nabla u||)u_x u_y}{u_x^2+u_y^2}\right)u_{xy}\right.\\ &\left.+\left(\frac{g(||\nabla u||)^2 u_x u_y - g(||\nabla u||)u_x u_y}{u_x^2+u_y^2}\right)u_{yx} + \left(\frac{g(||\nabla u||)^2 u_y^2 + g(||\nabla u||)u_x^2}{u_x^2+u_y^2}\right)u_{yy}\right]\\ &-\partial_{yy}\left[\left(\frac{g(||\nabla u||)^2 u_x^2 + g(||\nabla u||)u_y^2}{u_x^2+u_y^2}\right)u_{xx} + \left(\frac{g(||\nabla u||)^2 u_x u_y - g(||\nabla u||)u_x u_y}{u_x^2+u_y^2}\right)u_{xy}\right.\\ &\left.+\left(\frac{g(||\nabla u||)^2 u_x u_y - g(||\nabla u||)u_x u_y}{u_x^2+u_y^2}\right)u_{yx} + \left(\frac{g(||\nabla u||)^2 u_y^2 + g(||\nabla u||)u_x^2}{u_x^2+u_y^2}\right)u_{yy}\right]\end{aligned} \tag{4.20}$$

From this method, Li et al. [62] derived two anisotropic fourth-order PDEs that, to our knowledge, are the only anisotropic fourth-order models that have been applied to inpainting previously. We will refer to them as Li 1

$$\partial_t u = -\Delta(g(||\nabla u||)u_{NN} + u_{TT}) \tag{4.21}$$

and Li 2

$$\partial_t u = -\Delta(u_{TT})\,. \tag{4.22}$$

Li 1 can be re-written as

$$\begin{aligned}\partial_t u = -\partial_{xx}\Bigg[&\left(\frac{g(||\nabla u||)u_x^2+u_y^2}{u_x^2+u_y^2}\right)u_{xx}+\left(\frac{g(||\nabla u||)u_xu_y-u_xu_y}{u_x^2+u_y^2}\right)u_{xy}\\&+\left(\frac{g(||\nabla u||)u_xu_y-u_xu_y}{u_x^2+u_y^2}\right)u_{yx}+\left(\frac{g(||\nabla u||)u_y^2+u_x^2}{u_x^2+u_y^2}\right)u_{yy}\Bigg]\\-\partial_{yy}\Bigg[&\left(\frac{g(||\nabla u||)u_x^2+u_y^2}{u_x^2+u_y^2}\right)u_{xx}+\left(\frac{g(||\nabla u||)u_xu_y-u_xu_y}{u_x^2+u_y^2}\right)u_{xy}\\&+\left(\frac{g(||\nabla u||)u_xu_y-u_xu_y}{u_x^2+u_y^2}\right)u_{yx}+\left(\frac{g(||\nabla u||)u_y^2+u_x^2}{u_x^2+u_y^2}\right)u_{yy}\Bigg],\end{aligned} \tag{4.23}$$

while Li 2 becomes

$$\begin{aligned}\partial_t u = -\partial_{xx}\Bigg[&\left(\frac{u_y^2}{u_x^2+u_y^2}\right)u_{xx}+\left(\frac{-u_xu_y}{u_x^2+u_y^2}\right)u_{xy}+\left(\frac{-u_xu_y}{u_x^2+u_y^2}\right)u_{yx}+\left(\frac{u_x^2}{u_x^2+u_y^2}\right)u_{yy}\Bigg]\\-\partial_{yy}\Bigg[&\left(\frac{u_y^2}{u_x^2+u_y^2}\right)u_{xx}+\left(\frac{-u_xu_y}{u_x^2+u_y^2}\right)u_{xy}+\left(\frac{-u_xu_y}{u_x^2+u_y^2}\right)u_{yx}+\left(\frac{u_x^2}{u_x^2+u_y^2}\right)u_{yy}\Bigg].\end{aligned} \tag{4.24}$$

We observe that Li 1 is based on a similar idea as our proposed PDE: It permits fourth-order diffusion along the edge, while applying a nonlinear diffusivity function across the edge. However, expressing Li et al.'s models in terms of fourth-order diffusion tensors $\mathcal{D}_1$ and $\mathcal{D}_2$ reveals that our approach is more general. In particular, we can observe that

$$\begin{aligned}\mathcal{D}_1 : \mathbf{H} &= g(||\nabla u||)u_{NN}\mathbf{I}+u_{TT}\mathbf{I}\,,\\ \mathcal{D}_2 : \mathbf{H} &= u_{TT}\mathbf{I}\,,\end{aligned} \tag{4.25}$$

where $\mathbf{I}$ is the 2×2 identity matrix. In our model, $\mathcal{D} : \mathbf{H}$ can yield arbitrary anisotropic tensors. In this sense, our model more fully accounts for anisotropy compared to the ones by Hajiaboli and Li et al.

The fourth-order Equation (4.16) involves inner second derivatives of the image, which then get scaled by diffusivities, before outer second derivatives are taken. We observe that, in both cases, our model accounts for mixed derivatives that are ignored by previous approaches to anisotropic fourth-order diffusion: In the outer derivatives, this can be seen from the fact that Eq. (4.9) involves mixed derivatives, while Hajiaboli and Li et al. only consider the Laplacian.

Similarly, our definition of a fourth-order diffusion tensor $\mathcal{D}$ accounts for mixed derivatives also in the inner derivatives. Following Equation (4.14), we obtain

$$\begin{aligned}\mathcal{D} : \mathbf{H} &= \mu_1(\mathbf{E}_1\otimes\mathbf{E}_1) : \mathbf{H}+\mu_2(\mathbf{E}_2\otimes\mathbf{E}_2) : \mathbf{H}+\mu_3(\mathbf{E}_3\otimes\mathbf{E}_3) : \mathbf{H}\\ &= \mu_1 u_{\mathbf{v}_1\mathbf{v}_1}\mathbf{E}_1+\mu_2 u_{\mathbf{v}_2\mathbf{v}_2}\mathbf{E}_2+\frac{\mu_3}{\sqrt{2}}\left(u_{\left(\frac{\mathbf{v}_1+\mathbf{v}_2}{\sqrt{2}}\right)\left(\frac{\mathbf{v}_1+\mathbf{v}_2}{\sqrt{2}}\right)}-u_{\left(\frac{\mathbf{v}_1-\mathbf{v}_2}{\sqrt{2}}\right)\left(\frac{\mathbf{v}_1-\mathbf{v}_2}{\sqrt{2}}\right)}\right)\mathbf{E}_3\,.\end{aligned} \tag{4.26}$$

Comparing Equations (4.25) and (4.26) reveals differences in the considered directions: First, N and T are derived from the unregularized gradient, while the corresponding directions $\mathbf{v}_1$ and $\mathbf{v}_2$ in our model include a Gaussian pre-smoothing. A

second difference is that our model involves an additional term, which is steered by μ_3, and accounts for the directions in between the regularized gradient and its orthogonal vectors, i.e., $\left(\frac{\mathbf{v}_1+\mathbf{v}_2}{\sqrt{2}}\right)$ and $\left(\frac{\mathbf{v}_1-\mathbf{v}_2}{\sqrt{2}}\right)$. As it was demonstrated in the previous section, this term can have a noticeable effect on the outcome. Overall, we conclude that our newly proposed model is more general than the previously published ones.

4.4.3 Discretization and Stability

When discretizing Equation (4.13) with standard finite differences

$$\begin{aligned} u_{xx} &\approx \frac{(u_{i-1,j} - 2u_{i,j} + u_{i+1,j})}{(\Delta x)^2} \,, \\ u_{yy} &\approx \frac{(u_{i,j-1} - 2u_{i,j} + u_{i,j+1})}{(\Delta y)^2} \,, \\ u_{xy} &\approx \frac{(u_{i-1,j-1} + u_{i+1,j+1} + u_{i-1,j+1} + u_{i+1,j-1})}{4(\Delta x)(\Delta y)} \,, \\ u_{yx} &= u_{xy} \,, \end{aligned} \tag{4.27}$$

we can write it down in matrix-vector form as in [115],

$$\mathbf{u}^{k+1} = \mathbf{u}^k(\mathbf{I} - \tau\, \mathbf{P}_k) \,, \tag{4.28}$$

where u^k is an mn dimensional image vector at iteration k. m, n are image width and height respectively; Δx and Δy are the corresponding pixel edge lengths. $\mathbf{P}_k$ is a positive semi-definite matrix that, with step size τ, leads to the system matrix $(\mathbf{I} - \tau\, \mathbf{P}_k)$. The notation $\mathbf{P}_k$ indicates that it is iteration dependent, i.e., $\mathbf{P}_k = \mathbf{P}(\mathbf{u}_k)$.

Stability of fourth-order PDEs for image processing is typically formalized in an L_2 sense, i.e., a time step is chosen such that

$$||\mathbf{u}^{k+1}||_2 \leq ||\mathbf{u}^k||_2 \,. \tag{4.29}$$

In an inpainting scenario, it depends on our initialization of the unknown pixels whether we can expect Equation (4.29) to hold. Therefore, we rely on a stability analysis of the smoothing variant of our proposed PDE. This variant is obtained by removing the Dirichlet boundary conditions and instead solving a standard initial value problem. In this case, the stability analysis presented by Gorgi Zadeh et al. [115] carries over. It ensures that time step sizes

$$\tau \leq \frac{2}{16(\Delta x)^2 + 16(\Delta y)^2 + 2(\Delta x \Delta y)} \tag{4.30}$$

are stable in the L_2 sense. For a spatial discretization $\Delta x = \Delta y = 1$, this yields $\tau \leq 1/17 \approx 0.0588$. In inpainting, we empirically obtained a useful steady state with a time step size $\tau \leq 0.066$, independent of the initialization. The fact that this slightly exceeds the theoretical step size reflects the fact that Equation (4.30) results from deriving a sufficient, not a necessary condition for stability.

Stability of fourth-order schemes generally requires a quite small time step τ. This makes it computationally expensive to reach the steady state by evaluating Equation (4.28). Hafner et al. [35] propose a remedy to this problem, the so-called Fast Semi-Iterative Scheme (FSI). It extrapolates the basic solver iteration with the previous iterate and serves as an accelerated explicit scheme. The acceleration of the

explicit scheme (4.28) is given as

$$\mathbf{u}^{m,k+1} = \alpha_k \cdot (\mathbf{I} - \tau \mathbf{P}(\mathbf{u}^{m,k}))\mathbf{u}^{m,k} + (1 - \alpha_k) \cdot \mathbf{u}^{m,k-1}\,, \tag{4.31}$$

where $\mathbf{u}^{m,-1} := \mathbf{u}^{m,0}$ and $\alpha_k = \frac{4k+2}{2k+3}$ for $k = 0, \cdots, n-1$. Here m stands for outer cycle, i.e. m-th cycle with inner cycle of length n. And for passing to the next outer cycle, we set $\mathbf{u}^{m+1,0} := \mathbf{u}^{m,n}$. The stability analysis requires the matrix $\mathbf{P}$ to be symmetric. This is satisfied since the diffusion tensor $\mathcal{D}$ is symmetric, and symmetric central discretizations are used. In our implementation, we used $n = 40$, and stopped iterating after the first outer cycle for which $\|\mathbf{u}^m - \mathbf{u}^{m-1}\|_2 < 10^{-4}$.

4.5 Experimental Results

To establish the usefulness of our proposed new model, we applied it to the reconstruction of images from a sparse subset of pixels (Section 4.5.1). Moreover, we evaluate performance for a more classic inpainting task, scratch removal (Section 4.5.2). We also demonstrate how results depend on the chosen diffusivity function and contrast parameter (Section 4.5.3).

4.5.1 Reconstruction From a Sparse Set of Pixels

Improving image reconstruction from a sparse set of known pixels was the main motivation behind our work. Therefore, we applied it to two well-known natural images, *toucan* and *peppers*, as well as to a medical image, a slice of a T_1 weighted brain MR scan *(t1slice)*. For *toucan*, we kept a random subset of only 2% of the pixels. Due to the lower resolution of the *peppers* and *t1slice* images, we kept 5% and 20%, respectively.

In all three cases, results from our approach (FOEED) were compared to results from second-order EED, as well as from the two anisotropic fourth-order PDEs proposed by Li et al. [62]. In all experiments, we used the Charbonnier diffusivity function, we set the contrast parameter to $\lambda = 0.1$, and the pre-smoothing parameter to $\sigma = 1$.

Results for *toucan* are shown in Figure 4.2, for *peppers* in Figure 4.3, and for *t1slice* in Figure 4.4. A quantitative evaluation in terms of MSE and AAE is presented in Table 4.2. In terms of the numerical results, our proposed method produced a more accurate reconstruction than any of the competing approaches. Visually, there is a clear difference between second-order (EED) and fourth-order approaches (Li1, Li2, FOEED). Especially, we found that the shapes of edges were reconstructed more accurately. For example, we noticed this around the body and face in the *toucan* image (Figure 4.2). Similarly, the white and grey matter boundaries were better separated in the *t1slice* (Figure 4.4).

As we expected based on the theoretical analysis in Section 4.4.2, visual differences between the fourth-order methods are more subtle. However, in the *peppers* image (Figure 4.3), the tall and thin and the small and thick peppers in the foreground are much more clearly separated in the FOEED result than in any of the others.

In addition to experimenting with grayscale versions of the *toucan* and *peppers* images, we also applied EED and our FOEED filter channel-wise to the original RGB color versions. Results for *toucan* can be found in Figure 4.5, for *peppers* in Figure 4.6. Table 4.3 again provides a quantitative comparison. Similar observations can be made as in the grayscale images: Again, FOEED leads to lower reconstruction errors

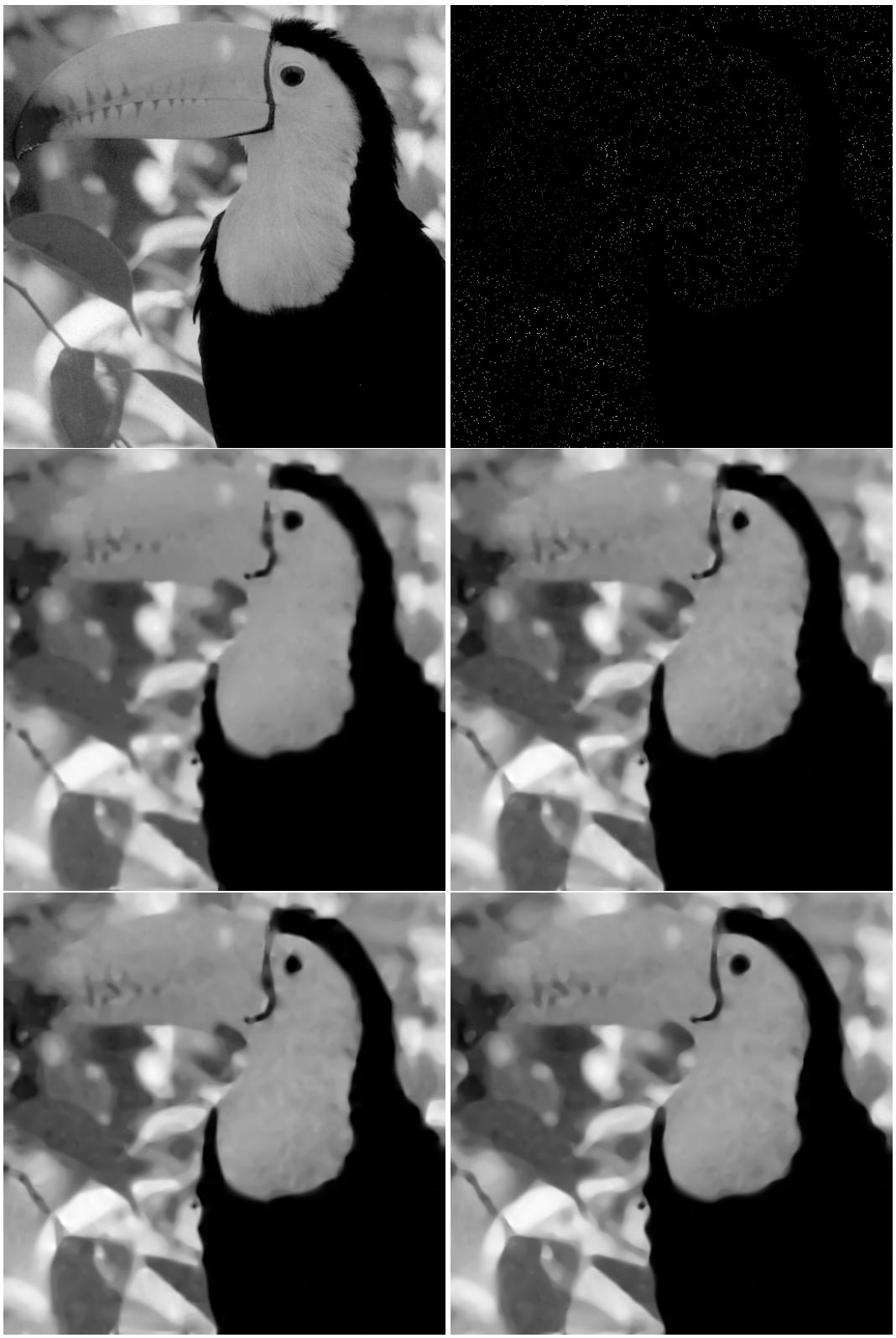


FIGURE 4.2: 1st row left: original *toucan* image of size 512×512; right: randomly chosen 2% of pixel values; 2nd row left: EED based inpainted image; right: Li1 based inpainted image; 3rd row left: Li2 based inpainted image; right: FOEED based inpainted image;

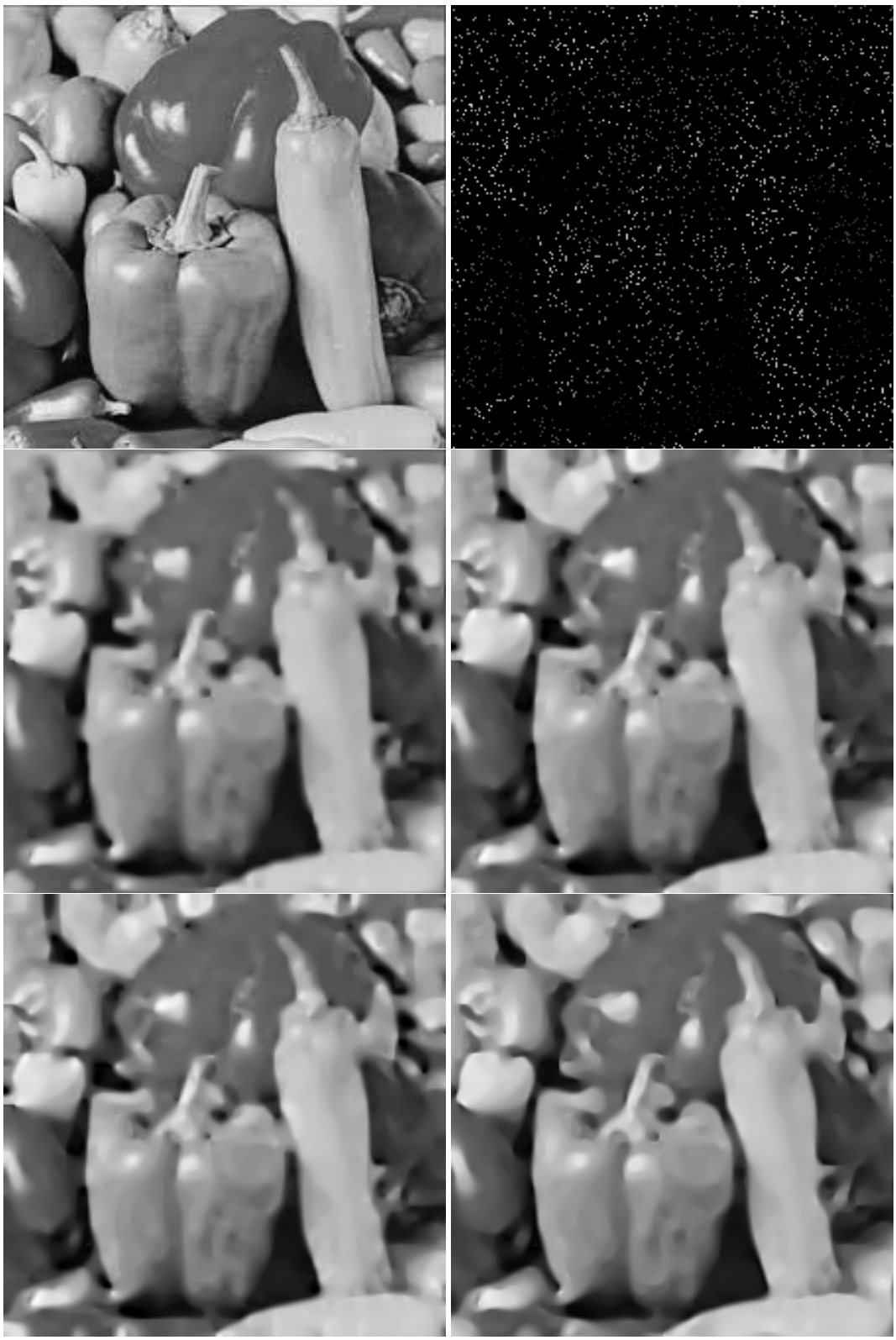


FIGURE 4.3: 1st row left: original *peppers* image of size 225×225; Right: randomly chosen 5% of pixel values; 2nd row left: EED based inpainted image; Right: Li1 based inpainted image; 3rd row left: Li2 based inpainted image; Right: FOEED based inpainted image.

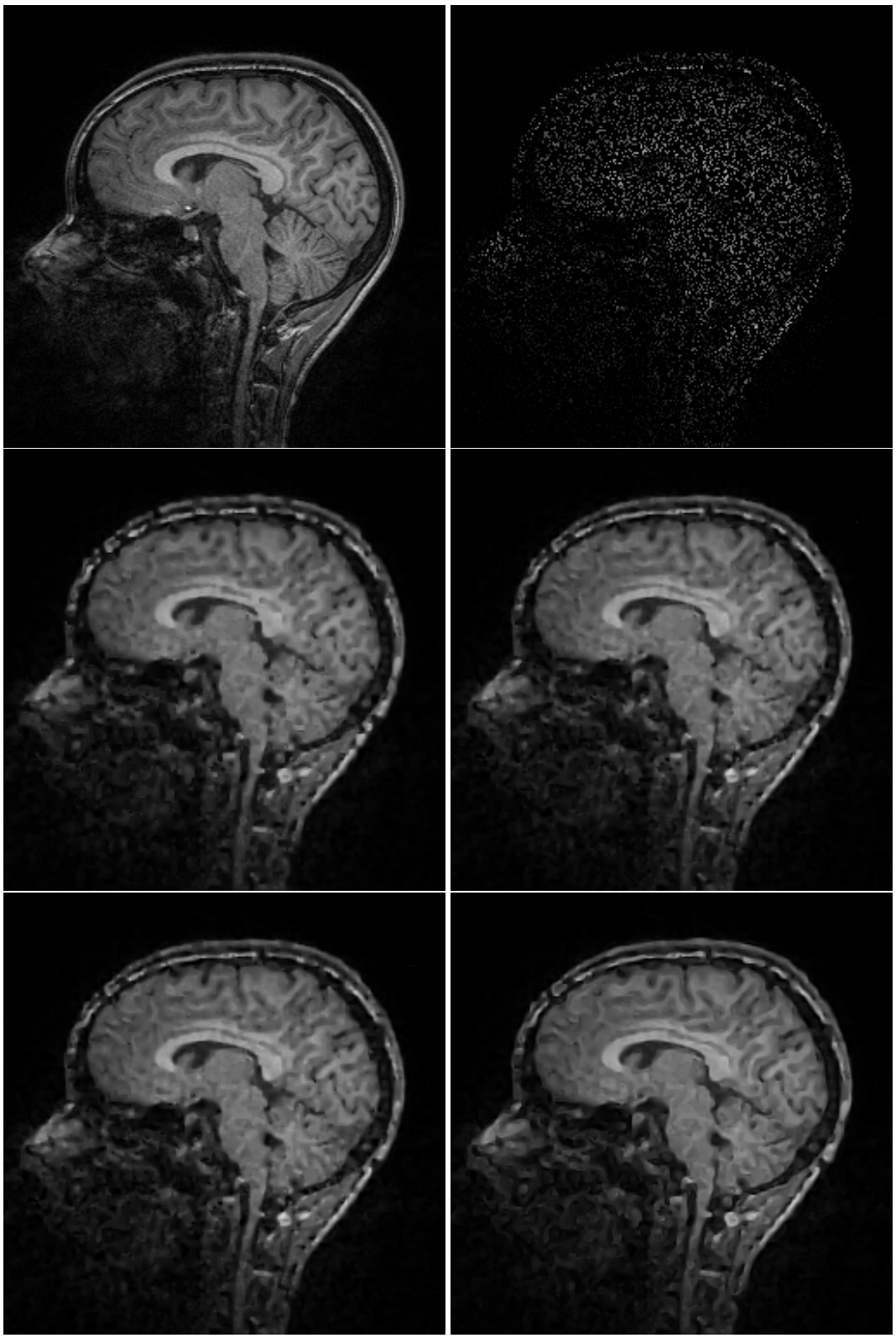


FIGURE 4.4: 1st row left: original *t1slice* image of size 256×256; Right: randomly chosen 20% of pixel values; 2nd row left: EED based inpainted image; Right: Li1 based inpainted image; 3rd row left: Li2 based inpainted image; Right: FOEED based inpainted image.

TABLE 4.2: Numerical Comparison of Inpainting Models for Gray-Valued Images

Image	Errors	EED	FOEED	Li1	Li2
toucan	MSE	105.37	**96.228**	100.994	102.665
	AAE	4.488	**4.164**	4.397	4.465
peppers	MSE	467.261	**443.129**	455.633	459.606
	AAE	10.94	**10.523**	11.042	11.107
t1-slice	MSE	166.356	**150.002**	152.698	155.955
	AAE	5.895	**5.698**	5.789	5.853

FIGURE 4.5: RGB *toucan* image, reconstructed from randomly chosen 2% of pixel values using EED (left) or FOEED (right).

than EED, it visually reconstructs edges more accurately, and separates the peppers more clearly.

Finally, we reconstructed images from a larger number of pixels, to obtain visually cleaner results. Qualitative and numerical results are presented in Figure 4.7 and Table 4.4, respectively. FOEED still yields lower numerical errors than EED. Unsurprisingly, the differences become smaller and less visually prominent as the mask density increases. The table also reveals that FOEED requires more CPU time compared to standard EED. However, due to the use of FSI in both cases, the difference in running times until convergence is much lower than the difference in time step sizes.

4.5.2 Scratch Removal

Li et al. [62] proposed their anisotropic fourth-order PDE for more classical image inpainting tasks, such as scratch removal. We evaluated whether our more general filter can also provide a benefit in such a scenario by reconstructing a scratched version of the *peppers* image. Similar to Li et al., we first made the scratches rather thin, covering only 6% of all pixels. Results are shown in Figure 4.8 and in Table 4.5. In this case, all methods work well: Numerical errors are small and similar between methods, and even though FOEED achieves the best numerical result, differences are difficult to discern visually.

FIGURE 4.6: RGB *peppers* image, reconstructed from randomly chosen 5% of pixel values using EED (left) or FOEED (right).

TABLE 4.3: Numerical Comparison of Inpainting Models for RGB Images

Image	Errors	EED	FOEED
toucan	MSE	119.062	**108.061**
	AAE	4.819	**4.59416**
peppers	MSE	478.799	**441.203**
	AAE	11.049	**10.543**

Therefore, we created a more challenging version with thicker scratches, covering 18% of all pixels (Figure 4.9). The corresponding numerical comparison is shown in Table 4.6. Here, FOEED achieves the most accurate reconstruction. Visually, we again observe that edges are reconstructed more accurately, and objects are more clearly separated, with fourth-order compared to second-order diffusion, and that steering it with a fourth-order diffusion tensor again provides small additional benefits over the previous methods.

4.5.3 Effect of Diffusivity Function and Contrast Parameter

For image inpainting with second-order PDEs, the Charbonnier diffusivity was previously found to work better than other established diffusivity functions. To assess whether this is still true in the fourth-order case, we repeated the reconstruction of the *peppers* image as shown in Figure 4.3 with different diffusivities. Table 4.7 summarizes the results. We conclude that the Charbonnier diffusivity still appears to be optimal.

Finally, in Figure 4.10, we illustrate how the reconstructed image depends on the contrast parameter λ. As expected, increasing λ leads to an increased blurring of edges. In the limit, the diffusivity function takes on values close to 1 over a substantial part of the image, and our model starts to approximate homogeneous fourth-order diffusion.

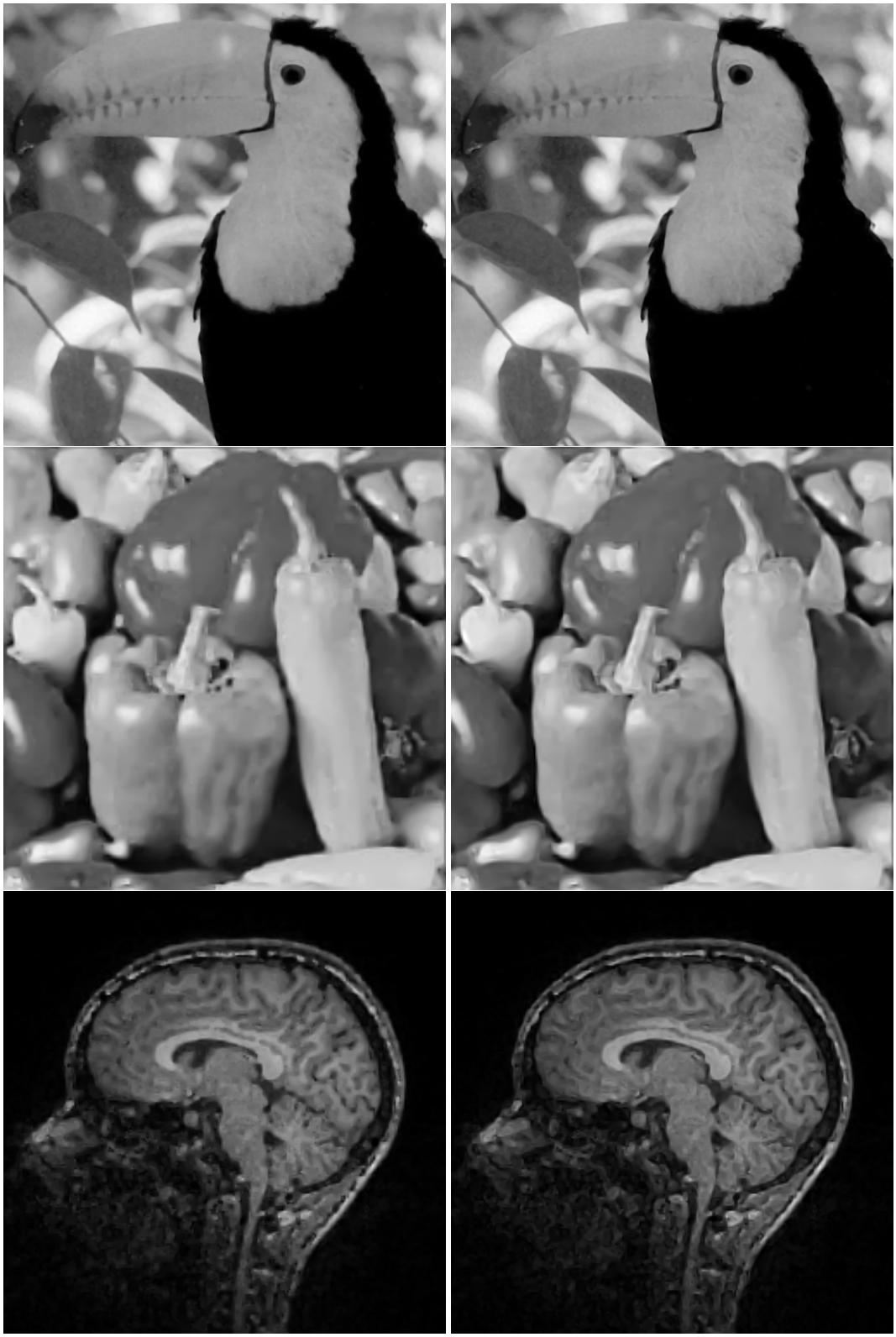


FIGURE 4.7: Higher quality reconstructions from a larger subset of pixels. 1st row: *toucan* image, reconstructed with EED (left) or FOEED (right) from randomly chosen 14% of pixels; 2nd row left: same for 20% of pixels from *peppers*; 3rd row left: same for 30% of pixels from *t1slice*. As expected, increasing the fraction of known pixels reduces the differences in the results of the two schemes.

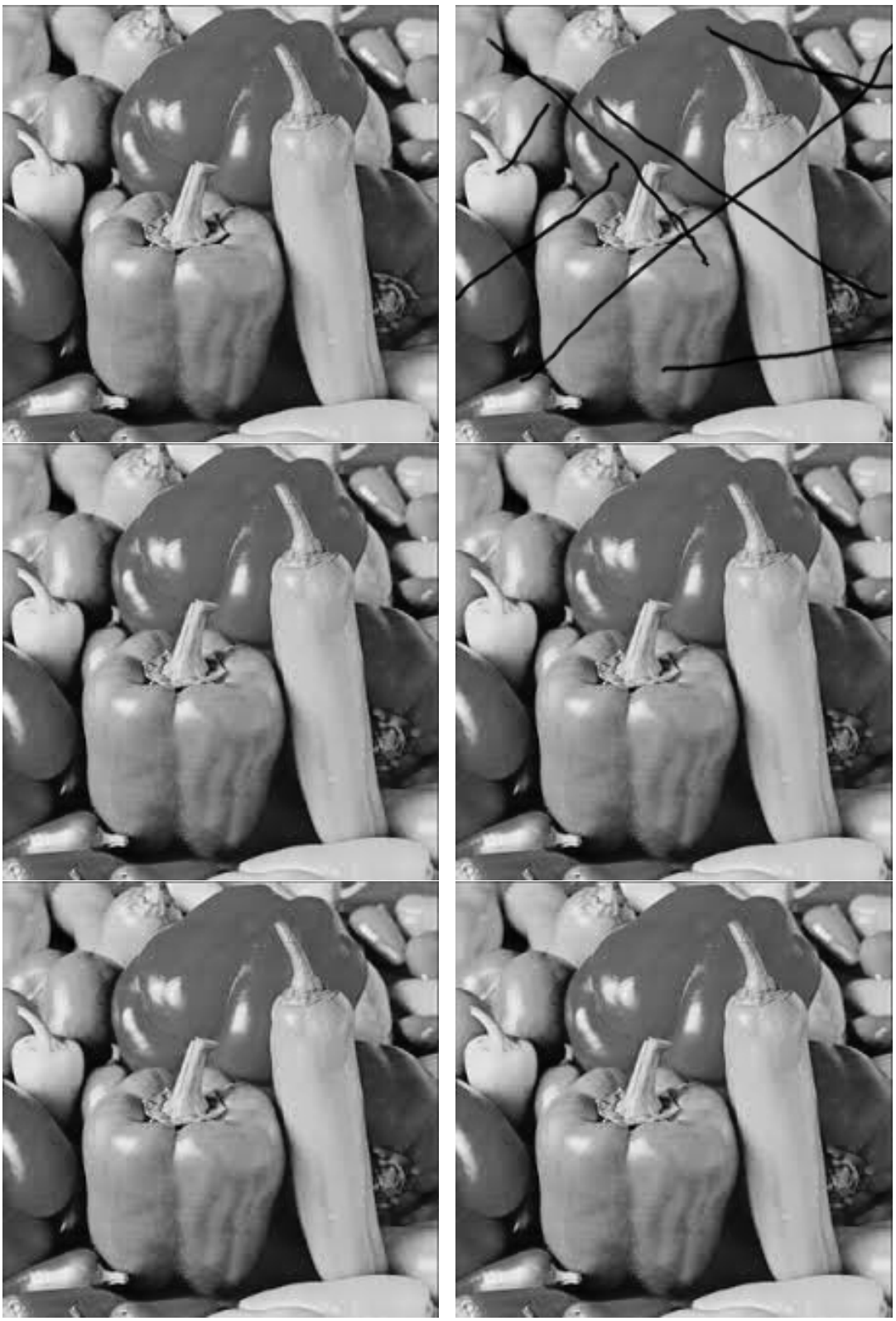


FIGURE 4.8: 1st row left: original *peppers* image of size 225×225; Right: corrupted image. 2nd row left: EED based inpainting; Right: Li1 based inpainting. 3rd row left: Li2 based inpainting; Right: FOEED based inpainting.

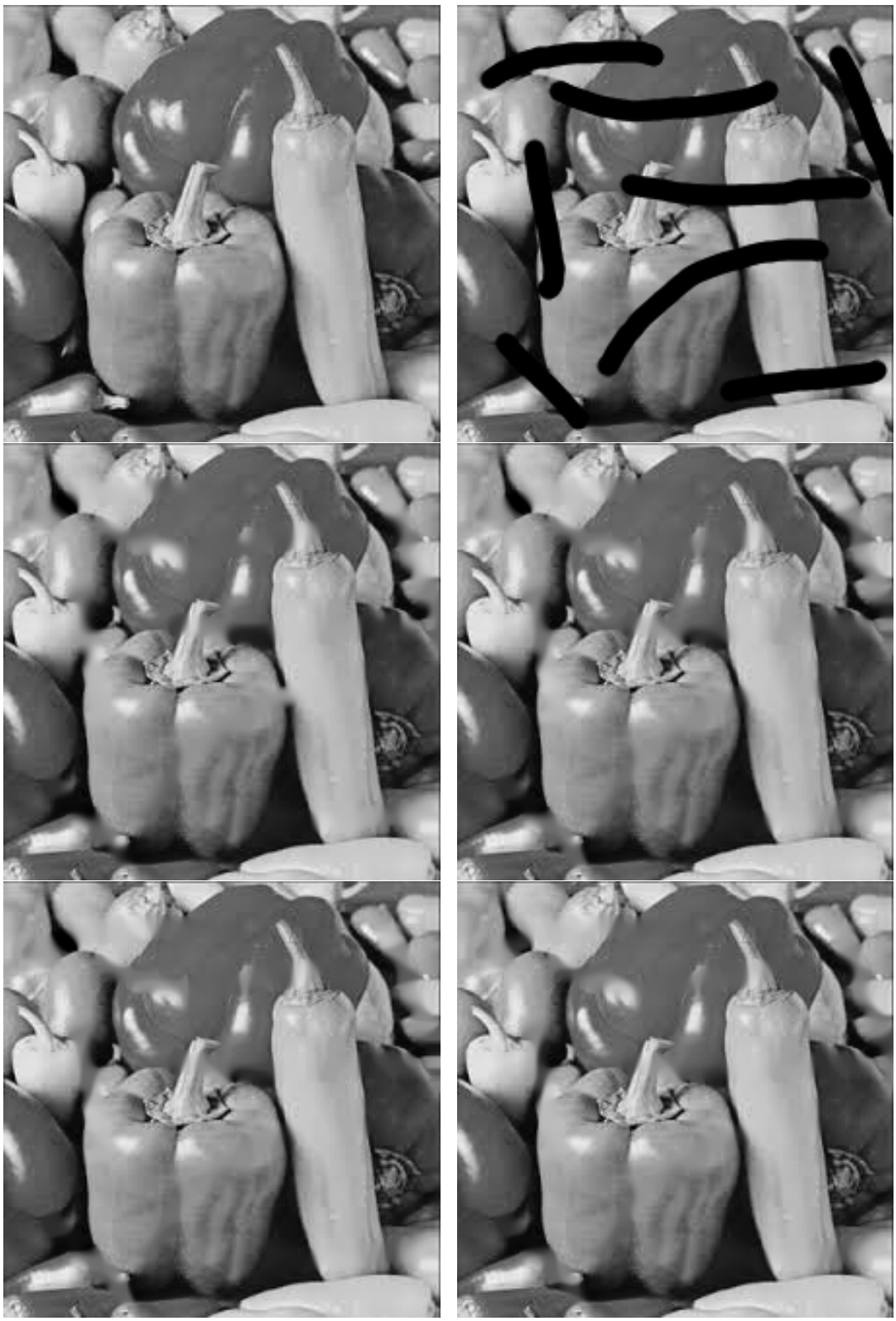


FIGURE 4.9: 1st row left: original *peppers* image of size 225×225; Right: corrupted image. 2nd row left: EED based inpainting; Right: Li1 based inpainting. 3rd row left: Li2 based inpainting; Right: FOEED based inpainting.

TABLE 4.4: Numerical comparison and computation times corresponding to Figure 4.7

Image	Errors	EED	FOEED	CPU time
toucan	MSE	18.029	**17.295**	53.060 (FOEED)
	AAE	1.696	**1.686**	21.259 (EED)
peppers	MSE	113.5	**110.885**	20.999 (FOEED)
	AAE	4.565	**4.441**	19.79 (EED)
t1-slice	MSE	114.845	**107.323**	24.74 (FOEED)
	AAE	4.610	**4.553**	10.64 (EED)

TABLE 4.5: Numerical comparison for peppers with thinner scratches (Figure 4.8)

Image	Errors	EED	FOEED	Li1	Li2
peppers	MSE	9.520	**7.813**	8.161	8.132
	AAE	0.363	**0.326**	0.346	0.346

4.6 Conclusions

We introduced a novel fourth-order PDE for edge enhancing diffusion (FOEED), steered by a fourth-order diffusion tensor. We implemented it using a fast semi-iterative scheme, and demonstrated that it achieved improved accuracy in several inpainting tasks, including reconstructing images from a small fraction of pixels, or removing scratches.

Our main motivation for using fourth-order diffusion in this context is the increased smoothness of results compared to second-order PDEs [114], which we expected to result in visually more pleasant reconstructions. The model in our current work is still based on a single edge direction at each pixel, extracted via a traditional second-order structure tensor. It is left as a separate research goal for future work to combine this with approaches for the estimation of complex structures such as crossings or bifurcations [1, 98], and with their improved reconstruction, e.g., by operating on the space of positions and orientations [11, 22, 28].

Finally, our current work only considered reconstructions from a random subset of pixels. A practical image compression codec that uses our novel PDE should investigate how it interacts with more sophisticated approaches for selecting and coding inpainting masks [96].

TABLE 4.6: Numerical comparison for peppers with thicker scratches (Figure 4.9)

Image	Errors	EED	FOEED	Li1	Li2
peppers	MSE	104.744	**78.761**	101.670	101.592
	AAE	2.455	**2.099**	2.465	2.450

TABLE 4.7: Numerical Comparison of FOEED with Different Diffusivity Functions

Image	Errors	Charbonnier [17] $\frac{1}{\sqrt{1+(\frac{s}{\lambda})^2}}$	Aubert [18] $\frac{(\frac{s}{\lambda})^2}{(s^2+\lambda^2)^2}$	Perona-Malik [82] $\frac{1}{1+(\frac{s}{\lambda})^2}$	Perona-Malik2 [82] $e^{-(\frac{s}{\lambda})^2}$	Geman-Reynolds [33] $\frac{2\lambda^2}{(s^2+\lambda^2)^2}$
peppers	MSE	**443.129**	458.961	478.411	491.153	491.186
	AAE	**10.523**	10.587	10.943	11.157	11.007

FIGURE 4.10: From left to right: FOEED based inpainted image with $\lambda = 0.1$, $\lambda = 0.5$, $\lambda = 15.5$.

Acknowledgement

This research was supported by the German Academic Exchange Service (DAAD).

Chapter 5

Lossless PDE-based Compression of 3D Medical Images

Inpainting PDEs can achieve a very close reconstruction of an original image from a very sparse subset of pixels. This led to research of inpainting PDEs in image compression. The well-known PDEs used in image compression is edge-enhancing diffusion PDEs (EED). Works done on image compression with EED are all lossy compression. This lossy codecs are not usually useful for medical image compression due to diagnostic and legal reasons. In this chapter, we present our designed new codec for lossless compression of three dimensional medical images. Our work is the first to explore the potential of PDE-based methods for the lossless compression of 3D medical images.

The codec consists of the following parts:

Constructing the Initial Mask: We use a simple regular grid as the initial inpainting mask, so that the locations of the mask voxels do not have to be stored explicitly. This releases us from substantial effort in selecting a suitable small subset of pixels as an inpainting mask from which the original image can be approximately reconstructed and the cost of coding the initial mask. More precisely, for a 3D medical image, our initial mask is the hexahedral grid consisting of voxels of every fourth in each of all three spatial dimensions. This amounts to storing the intensities of approximately 1.6% of all voxels. We tried also to use more sophisticated masks that exploit edge information, but found that, even though it yielded even more close reconstruction and hence more compressible residuals, the cost of coding the masks grew disproportionally.

Iterative Reconstruction and Residual Coding: To achieve a lossless codec the residuals in all non-mask voxels need to be coded. However, we found the initial reconstruction to be so coarse that this does not yet yield a competitive compression rate. This reflects the fact that our initial mask does not adequately sample all semantically relevant image structures. We compensate for this by an iterative reconstruction and coding of residuals. In each iteration, we first reconstruct the image from the current inpainting mask. We then store the residuals in the immediate vicinity of the current mask. Those residuals are typically the most compressible, since the uncertainty in the inpainting result tends to increase with distance away from the known part of the image. Voxels whose residuals are stored are added to the next iteration of the inpainting mask. The decoder mirrors this iterative reconstruction, again starting with the initial mask, then adding the stored residuals from the immediate neighbors to the reconstruction results. This yields the original intensities in a subdomain of the image that grows with each iteration, until all voxels have become part of the mask. The mask growing is performed by applying a morphological dilation to the mask. For reconstruction, we experimentally determined the suitability of three different PDEs for lossless compression: Linear homogeneous

diffusion as a simple baseline, edge-enhancing diffusion (EED), which is a popular choice for PDE-based lossy compression, and our introduced fourth-order generalization of EED.

Our experimental tests are divided into four groups depending on medical image properties: 1) A diffusion MRI scan with b = 0 with 73% of voxel values are zeros. This results in a test case that is analogous to hybrid compression, in which only a clinically relevant region of interest (ROI) is losslessly compressed. 2) A diffusion MRI scan with b = 700 with less than 10% of voxels are zeros. This test case is for medical images with a noisy background region. 3) A T1 weighted MR image with much less noise and more than 65% zero voxels. This T1 weighted MRI has higher spatial resolution which could be exploited by a PDE-based inpainting. 4) A foot CT image which has the noisy appearance within the foreground region. We demonstrate that our codec achieves a higher compression rate than several established codecs on three MRIs with different characteristics, as well as a CT image.

The content of this chapter is published as a peer reviewed conference paper:

Jumakulyyev Ikram, Schultz Thomas "Lossless PDE-based Compression of 3D Medical Images" In: *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer. 2021, pp. 450-462. [50].

Contribution of the dissertation author: Conceptualization, methodology, literature review, algorithm implementation and validation, results interpretation and visualization, manuscript composition, validation experiment design.

5.1 Abstract

Inpainting with Partial Differential Equations (PDEs) has previously been used as a basis for lossy image compression. For medical images, lossless compression is often considered to be safer, given that even subtle details could be diagnostically relevant. In this work, we introduce a PDE-based codec that achieves competitive compression rates for lossless image compression. It is based on coding the differences between the original image and its PDE-based reconstruction. These differences often have lower entropy than the original image, and can therefore be coded more efficiently. We optimize this idea via an iterative reconstruction scheme, and a separate coding of empty space, which takes up a considerable fraction of the field of view in many 3D medical images. We demonstrate that our PDE-based codec compares favorably to previously established lossless codecs. We also investigate the individual benefit from each ingredient of our codec on multiple examples, explore the effect of using homogeneous, edge enhancing, and fourth-order anisotropic diffusion, and discuss the choice of contrast parameters.

5.2 Introduction

The overall size of neuroimaging data that is acquired each year has been reported to grow exponentially [26], due to the proliferation of medical imaging devices, their increased resolution, and the increasing use of multiple contrasts or channels. This makes the development of compression schemes for the storage of 3D medical images an important and timely research goal.

The use of diffusion-based inpainting has been explored for the lossy compression of images [30, 96, 97], videos [5, 56, 89], and audio [85]. This paradigm is based on storing information only for a sparse subset of the original samples, and interpolating it to approximate the remaining parts of the original signal. Interpolation

is often done via Partial Differential Equations (PDEs) that are inspired by the well-known heat transfer equation, in analogy to how radiators that are sparsely distributed in a room would heat up the space in between them.

Almost all previous works on PDE-based compression have focused on 2D natural images or videos. Only a single example has considered a 3D extension [84]. Even more importantly, all above-mentioned codecs are for lossy compression, and their benefit relative to established transform-based codecs like JPEG [81] and JPEG2000 [101] tends to be most pronounced at high compression rates [97]. However, compression schemes that lead to visually noticeable changes are less suitable for medical images, since potentially subtle, but diagnostically relevant details might be perturbed. Therefore, lossless compression is often preferred and is sometimes even legally required [52, 53, 75], since it guarantees not to interfere with interpretation or quantification of the image contents.

Our work is the first to explore the potential of PDE-based methods for the lossless compression of 3D medical images. In Section 6.4, we present a lossless PDE-based codec that stores the residuals between the PDE-based reconstruction and the original values. Its success rests on three key ideas: First, we use a simple regular grid as the initial inpainting mask, so that the locations of the mask voxels do not have to be stored explicitly. Second, we encode and decode the image iteratively, alternating between PDE-based reconstructions and a dilation of the inpainting mask. Compared to a single reconstruction, this further reduces the entropy of the residuals that have to be stored. Third, we optionally code regions of empty space separately, since they take up a substantial fraction of the field of view in many medical images.

In Section 6.5, we demonstrate that our codec achieves a higher compression rate than several established codecs on three Magnetic Resonance Images with different characteristics, as well as a Computed Tomography image. Moreover, we study the effect of several variations of our codec, using different PDEs, iteration modes, and contrast parameters.

5.3 Related Work

Several lossless compression standards are widely used in medical imaging. The Digital Imaging and Communications in Medicine (DICOM) standard defines a unified image file format for different devices, manufacturers, and modalities [59]. DICOM accounts for lossless compression with JPEG-LS, as well as lossy and lossless JPEG and JPEG2000. Consequently, these are most often used as a reference to which new codecs are compared: Lossy JPEG and JPEG2000 in case of lossy and hybrid or near-lossless medical image compression schemes [84, 119], the lossless JPEG family for lossless compression [52, 53].

The Neuroimaging Informatics Technology Initiative (NIfTI) defines an alternative file format that has been widely adopted for brain imaging. It consists of a header, followed by a binary representation of voxel intensities. NIfTI files are commonly compressed by simply applying GZIP [24] to them. GZIP is based on the Deflate algorithm [23], which is in turn based on the LZ77 and Huffman compression schemes, which have occasionally been used as an additional reference for lossless image compression [52]. In Section 5.5.1, we will compare our own codec to JPEG-LS, lossless JPEG and JPEG2000, as well as GZIP.

Prior to our work, the only PDE-based image compression codec for 3D medical images was C-EED [84]. It is based on edge-enhancing diffusion (EED) [110] and a cuboid subdivision scheme that extends the rectangular subdivision in the

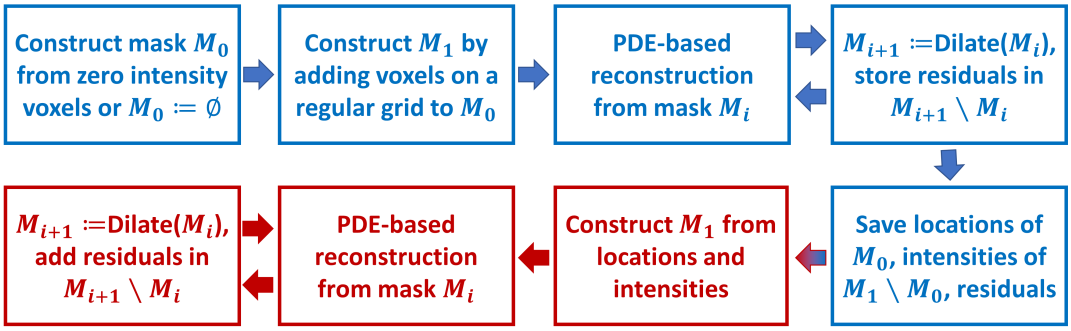


FIGURE 5.1: An overview of the individual steps taken to encode (blue) or decode (red) a 3D image. At the core of our codec is an iteration that alternates between PDE-based reconstruction from an inpainting mask, and a dilation of that mask.

previously proposed R-EED codec [97]. Since it aims for lossy compression, C-EED is based on very different design decisions than our codec. In particular, it applies brightness optimizations and quantization to the mask voxel values, which makes it more efficient to store them but, in our context, would require storing residuals even for the voxels in the inpainting mask.

Recent work has demonstrated the potential of deep learning for lossless compression of natural images [73]. Adapting such an approach to 3D medical images will have to account for data privacy, which makes it difficult to obtain large-scale training data and raises concerns about inference attacks [77]. We consider this to be a separate line of research which is outside of our scope.

5.4 Our Proposed Lossless Codec

Figure 5.1 shows a high-level overview of our PDE-based lossless codec. Section 5.4.1 will provide details on the first two steps, in which the encoder (blue) constructs an initial inpainting mask. The next two steps are the core of our approach. They alternate between a PDE-based reconstruction and a dilation of the mask, and will be discussed in Section 5.4.2. Finally, the initial mask and residuals are stored in compressed form (Section 6.4.4). The decoder (red) mirrors the encoder in that it again alternates between reconstruction and mask dilation.

5.4.1 Constructing the Initial Mask

In most lossy PDE-based image compression schemes, a substantial effort goes into selecting a suitable small subset of pixels as an inpainting mask from which the original image can be approximately reconstructed. To increase image quality, semantically important image features such as edges and corners are typically included in the mask [64, 70], and optimal inpainting masks have been approximated by sophisticated mask selection methods [29, 30, 96, 97].

Our lossless PDE-based codec restores the original image exactly by also coding the residual with respect to the PDE-based reconstruction. This strategy yields an advantage in terms of compression rate since the residuals are more compressible than the original intensities. However, our strategy only achieves a net benefit as long as the cost of coding the initial mask does not exceed the gain from increased compressibility of the residuals.

Therefore, we simply use a regular grid as our initial mask, which has the advantage of not having to store any voxel locations. In particular, for a 3D input image of size $n_x \times n_y \times n_z$, our initial mask is the hexahedral grid consisting of voxels $(4i, 4j, 4k)$, where $i \in \{0, 1, \ldots, \lfloor (n_x - 1)/4 \rfloor\}$, and j, k are defined accordingly. This amounts to storing the intensities of approximately 1.6% of all voxels. We attempted to use more sophisticated masks that exploit edge information, but found that, even though it yielded even more compressible residuals, the cost of coding the masks grew disproportionally.

Our codec exploits the fact that many medical images contain a substantial amount of empty space, which typically yields the lowest possible intensity, and can be coded efficiently as a run length encoded binary mask. In the following, we assume that the minimum intensity will be zero. In practice, our encoder deals with negative intensities, as they arise in computed tomography (CT), by subtracting the minimum from the original input and storing it, so that the decoder can add it again to its output. In some cases, the gain from including voxels with zero intensity in the preliminary inpainting mask M_0 is substantial. In others, its cost outweighs its benefit, because intensities within empty space are perturbed by strong measurement noise, or the image contains little or no empty space. In this case, our encoder simply sets $M_0 := \emptyset$. The initial inpainting mask M_1 arises as the union of M_0 and the voxels on the above-described grid. We only store the intensities of grid voxels outside of M_0.

5.4.2 Iterative Reconstruction and Residual Coding

A straightforward lossless PDE-based codec would reconstruct the image from the inpainting mask M_1, and it would code the residuals in all non-mask voxels. However, we found the initial reconstruction to be so coarse that this does not yet yield a competitive compression rate. This reflects the fact that our initial mask does not adequately sample all semantically relevant image structures. We compensate for this by an iterative reconstruction and coding of residuals.

In each iteration, we first reconstruct the image from the current inpainting mask M_i. We then store the residuals in the immediate vicinity of the current mask. Those residuals are typically the most compressible, since the uncertainty in the inpainting result tends to increase with distance away from the known part of the image. Voxels whose residuals are stored are added to the inpainting mask M_{i+1} that will be used in the next iteration. The decoder mirrors this iterative reconstruction, again starting with the initial mask M_1, then adding the stored residuals from the immediate neighbors to the reconstruction results. This yields the original intensities in a subdomain of the image that grows with each iteration, until all voxels have become part of the mask.

We grow the mask by applying a morphological dilation to it. We experimented with two different structuring elements. The first is a cube, which amounts to a $3 \times 3 \times 3$ neighborhood. We call this Mode 0. The second is a cross, which amounts to the six face-connected neighbors. We call this Mode 1. Compression in Mode 0 requires two or three iterations, while Mode 1 takes six or seven iterations, depending on boundary effects. Section 5.5.3 will investigate the effect of the two different modes on the final compression rates. The computational cost of later iterations decreases with the number of remaining unknown voxels, and because we initialize them with the inpainting result from the previous iteration.

Residuals could be positive or negative. We avoid having to store them as signed integers by performing subtractions (in the encoder) and additions (in the decoder)

in modular arithmetic, with the maximum value as the modulus. As mentioned above, the minimum intensity at this point will always be zero.

5.4.3 Choice of PDE and its Parameters

Our compression strategy bears a certain conceptual resemblance to some established lossless codecs, such as JPEG-LS, which predict the values that have not yet been coded from the ones that are already known, and only code the residuals. Whether we can beat their compression rate should partly depend on whether PDE-based predictions are more successful at decreasing residual entropy compared to the simpler predictor used in JPEG-LS.

We experimentally determined the suitability of three different PDEs for lossless compression: Linear homogeneous diffusion as a simple baseline, edge-enhancing diffusion (EED), which is a popular choice for PDE-based lossy compression [97], and a recently introduced fourth-order generalization of EED [48].

Second-order diffusion can be stated as

$$\partial_t u = \operatorname{div}(\mathbf{D} \cdot \nabla u), \tag{5.1}$$

where u denotes the image intensity as a function of location within the image domain, and of diffusion time t. Diffusion-based inpainting uses the intensities in the mask voxels as Dirichlet boundary conditions, and obtains the inpainted result as the steady state that is attained as $t \to \infty$ [29].

In linear homogeneous diffusion, the diffusion tensor $\mathbf{D}$ is the identity. For EED, it is a symmetric matrix field that encodes directional dependence, so that diffusion across images edges is decreased depending on the gradient magnitude, while diffusion along the edge is free. In fourth-order EED, the first-order divergence and gradient operators in Eq. (5.1) are replaced with second-order counterparts, and the second-order diffusion tensor $\mathbf{D}$ is replaced with a fourth-order tensor that acts on the Hessian. For the sake of brevity, we refer the reader to [48, 110] for the full mathematical details and the numerical implementation of these PDEs.

The definitions of diffusion tensors for second- and fourth-order EED involve a diffusivity function, which determines the diffusivity across the edge as a function of gradient magnitude. As it is customary in PDE-based inpainting, we select the Charbonnier diffusivity function,

$$g(s^2) = \frac{1}{\sqrt{1 + \frac{s^2}{\lambda^2}}}, \tag{5.2}$$

where $s = \|\nabla_\sigma u\|$ is the gradient magnitude, computed with a certain amount of pre-smoothing. In our experiments, we fixed it at $\sigma = 1$. We also tried other values of σ, but found that this had only a very minor effect on the compression rate. This agrees with experimental findings in lossy image compression [96].

A second parameter in Eq. (5.2) is the contrast parameter λ, which corresponds to the scale of $\|\nabla_\sigma u\|$ at which g switches from high to low diffusivity. This parameter affects the quality of the PDE-based reconstructions, and therefore, the compression rate. Which contrast parameter value is optimal depends on the image contents, inpainting mask, and PDE. Some lossy PDE-based codecs have optimized λ by trying out different candidate values [96, 97].

Empirically optimizing λ causes a noticeable computational expense and, as will be reported in more detail in Section 5.5.4, we found its benefit in the context of

lossless compression to be relatively minor. Therefore, we rely on a heuristic choice of λ. It is based on one suggested by Perona and Malik [82], who proposed to set λ to the 90th percentile of the gradient magnitudes in the input image. We adapt this in two ways: First, we only consider gradient magnitudes outside of the initial mask M_1, in order to exclude the potentially large flat regions of empty space. Second, we need to account for the fact that inpainting from a sparse mask results in an image that is much smoother than the original one. For this reason, we divide the value at the 90th percentile by the empirical divisor 25. All reported results are based on this simple heuristic.

5.4.4 Compressed File Format

Our compressed files consist of a header, the locations of zero intensity voxels (if separating them yielded a benefit), the values at the initial mask voxels, as well as the residuals at non-mask voxels. We assume that intensity values are 16 bit integers, as it is common in medical imaging. If zero intensity voxels are coded separately, this is done as a binary mask, which is compressed using run length encoding, followed by the Deflate algorithm. The mask intensities and residuals are compressed using the Deflate algorithm or pure Huffman coding, depending on which choice resulted in the smaller size.

To ensure that comparisons to compressed NIfTI files are fair, we add the full NIfTI header (348 bytes), which includes the image dimensions among other information. In addition, we have to store the original minimum and maximum values (4 bytes), sizes of the compressed data streams for zero voxel binary mask and mask intensities (8 bytes), the contrast parameter (4 bytes), as well as single byte that encodes the type of PDE (2 bits), the dilation mode (1 bit), and the types of encoding for mask intensities and residuals (2 bits).

5.5 Results

For our experiments, we chose four 3D medical images which are illustrated in Figure 5.2. Even though three of them are from brain imaging, they have been chosen to represent diverse contrasts and properties.

1. A scan from diffusion MRI [7] with diffusion weight $b = 0$ and $136 \times 136 \times 84$ voxels. A brain extraction algorithm has zeroed out 73% of the voxels. This results in a test case that is analogous to hybrid compression, in which only a clinically relevant region of interest (ROI) is losslessly compressed [113, 119].

2. A diffusion MRI scan with $b = 700$ and $104 \times 104 \times 72$ voxels. This time, no brain extraction has been performed, and there is substantial measurement noise in the background, yielding less than 10% voxels with exactly zero intensity. This should provide a challenging test case for our codec, since the noisy background region should be difficult to compress losslessly.

3. A T1 weighted MR image with $256 \times 256 \times 220$ voxels. No brain extraction has been performed, but there is much less noise, leading to more than 65% zero voxels. Due to the higher spatial resolution, we expected a larger degree of spatial dependencies which could be exploited by a PDE-based inpainting.

4. A foot CT image with $256 \times 256 \times 256$ voxels, which we expected to be challenging due to the noisy appearance within the foreground region.

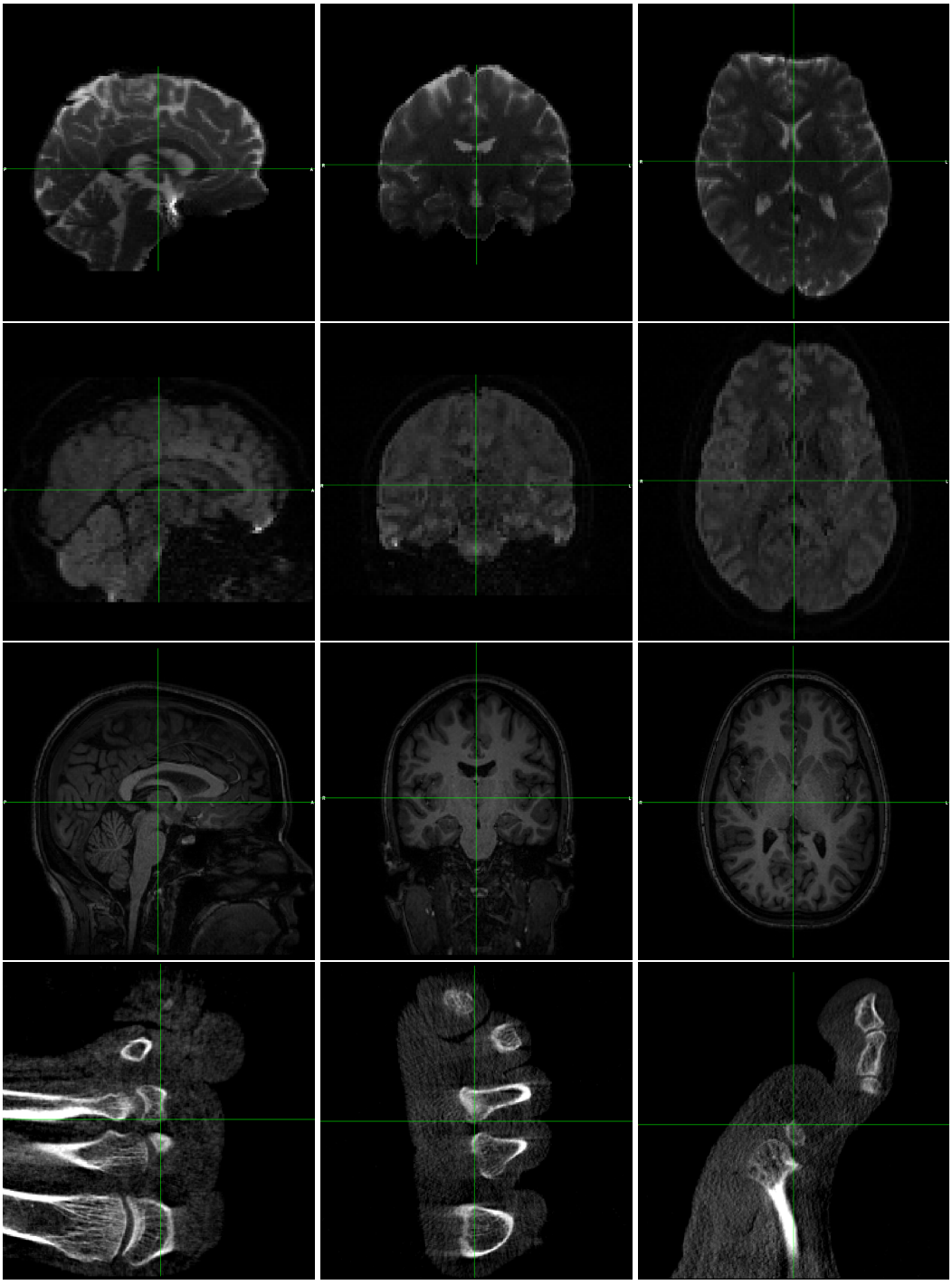


FIGURE 5.2: The four 3D medical images used in our experiments. Top to bottom: Three brain MR images with *B0*, *B700*, and *T1* weighting, and a foot *CT* image. Right to left: The middle slices on the sagittal, coronal, and axial planes.

TABLE 5.1: A comparison of different variants of our PDE-based codec to established lossless standards. Positive percentages indicate a relative benefit from our codec.

Image	PDE Codec	GZIP	JPEG	JPEG2000	JPEG-LS
B0	R-ILH-1	+26.489%	+29.747%	+17.238%	+2.980%
B0	R-IEED-1	+28.036%	+31.225%	+18.979%	+5.022%
B0	R-IFOEED-1	+28.784%	+31.940%	+19.821%	+6.009%
B700	R-ILH-1	+23.778%	+6.922%	−4.461%	+7.123%
B700	R-IEED-1	+27.167%	+11.061%	+0.184%	+11.253%
B700	R-IFOEED-1	+27.552%	+11.530%	+0.711%	+11.721%
T1	R-ILH-1	+32.294%	+31.912%	−5.142%	−1.650%
T1	R-IEED-1	+35.615%	+35.252%	+0.015%	+3.336%
T1	R-IFOEED-1	+37.925%	+37.575%	+3.602%	+6.804%
CT	R-ILH-1	+16.954%	+37.111%	+11.198%	+3.527%
CT	R-IEED-1	+19.886%	+39.332%	+14.334%	+6.934%
CT	R-IFOEED-1	+20.658%	+39.916%	+15.158%	+7.830%

5.5.1 Comparison to Other Codecs

Table 5.1 compares the compression rate of our proposed codec to four established alternatives, by specifying the relative differences in final file sizes. Positive values indicate a benefit of our codec. Details on the four lossless codecs included in our comparison are given in Section 5.3. Results consider different variants of our codec, using linear homogeneous (LH), second-order edge enhancing (EED), or fourth-order edge enhancing diffusion (FOEED). All cases use iterative reconstruction from a regular grid (R-I) in Mode 1 (see Section 5.4.2). The effect of using fewer or no iterations will be studied separately in Section 5.5.3.

In all four examples, we observe a clear improvement when moving from basic linear homogeneous diffusion to anisotropic diffusion. Highest compression rates were achieved with the recently introduced fourth-order EED. It allowed us to achieve higher lossless compression rates than any established codec. In many cases, the margin was considerable.

However, FOEED also had the highest computational cost. For the B700 image, iterative 3D reconstruction on a single 3.3 GHz CPU core took 27 s with LH diffusion, 478 s with EED, and 6185 s with FOEED. We expect that these times could be shortened significantly by a parallel implementation. This was not the focus of our current work.

5.5.2 Non-PDE Baseline for Further Comparisons

We require a suitable baseline to pinpoint the effect that the iterative reconstruction and tuning of the contrast parameter have on the final compression rate. Since we use Huffman coding or the Deflate algorithm for the final encoding, file sizes achieved with GZIP are a natural reference. In addition, we designed a baseline codec that is "in between" GZIP and our PDE-based codec.

First, we found that, when 3D images contain a substantial amount of empty space, coding it separately can increase compression rates. Therefore, our baseline codec performs the same run length encoding as our PDE-based codec if it decreases the overall file size. This was the case in all examples except B700.

TABLE 5.2: Delta coding of intensities already reduced entropy in all test images. We separately report this for the whole images, and within their non empty space regions.

Image	Image Entropy	Delta Coded Image Entropy	Nonzero Region Entropy	Delta Coded Nonzero Region Entropy
B0	3.63997	3.50099	10.33056	9.63958
B700	5.34271	5.01050	5.39311	5.11973
T1	3.14553	2.73150	6.46612	5.54785
CT	2.81012	2.66688	6.71209	6.15826

TABLE 5.3: Our non-PDE baseline that makes use of delta coding and optional empty space coding already results in a clear improvement over GZIP.

Image	GZIP (bytes)	Non-PDE Baseline	Zero Density	Zero Mask (bytes)
B0	692.372	+21.431%	72.92%	9.928
B700	610.968	+18.015%	9.07%	43.681
T1	5.207.535	+26.955%	65.70%	290.969
CT	4.515.257	+8.347%	71.10%	315.173

Second, we were wondering how much we can benefit already from a very simple, non PDE-based prediction of voxel intensities. To this end, we performed a delta coding, i.e., we fed differences between subsequent voxel intensities instead of the intensities themselves into the final compression. Table 5.2 shows that, in all cases, this decreased the entropy. It also slightly increased compression rates.

Finally, as in our PDE-based codec, our baseline codec uses either Deflate or pure Huffman coding, depending on what results in a smaller file. We used the same implementations from zlib and dippykit, respectively. Table 5.3 shows that this baseline already improves considerably over GZIP.

5.5.3 Effect of Iterative Construction of Residuals

Table 5.4 shows how the iterative alternation between reconstruction and residual coding that is described in Section 5.4.2 affects the overall file sizes achieved with our codec. Differences are relative to the non-PDE baseline from the previous section. In this experiment, second-order EED has been used for reconstruction. Positive values indicate a benefit of PDE-based predictions over delta coding.

Results indicate that a single reconstruction from a sparse regular grid is not sufficient to obtain a benefit from second-order anisotropic diffusion. On the other

TABLE 5.4: Compared to the non-PDE baseline, direct coding of residuals after a single reconstruction with second-order EED does not yet result in a clear benefit. However, an iterative reconstruction as described in Section 5.4.2 does.

Image	Direct Residual	Iteration in Mode 0	Iteration in Mode 1
B0	+0.095%	+4.367%	+8.406%
B700	+1.784%	+6.847%	+11.164%
T1	−0.698%	+6.715%	+11.845%
CT	−6.476%	+7.286%	+12.590%

TABLE 5.5: Results from varying the threshold that our heuristic uses to set the contrast parameter λ. Despite a noticeable effect on λ itself, the corresponding differences in compression rates are rather small. Improvements are relative to the non-PDE baseline.

Image	30% Threshold		60% Threshold		90% Threshold	
	λ	Improvement	λ	Improvement	λ	Improvement
B0	2.04741	+4.272%	4.42198	+4.362%	11.83980	+4.367%
B700	0.01379	+6.468%	0.10905	+6.794%	0.56127	+6.847%
T1	0.12347	+7.108%	0.30572	+7.017%	0.58445	+6.715%
CT	0.12464	+7.060%	0.22596	+7.225%	0.65364	+7.286%

hand, the proposed iterative reconstruction achieves a clear additional reduction in compressed file size. It is most pronounced in Mode 1, which dilates the inpainting mask with a cross-shaped structuring element and consequently requires more iterations than Mode 0, which dilates with a box.

5.5.4 Effect of Contrast Parameter

Even though it can be seen from Table 5.1 that moving from isotropic to anisotropic diffusion noticeably improved compression rates, we found that fine-tuning the contrast parameter in the diffusivity function is less important. Table 5.5 explores the effect of varying the ad-hoc threshold value of 90% that was used in Section 5.4.3 to two other values, 60% and 30%. For each image and threshold, the table reports the corresponding values of contrast parameter λ, as well as the resulting improvement over the non-PDE baseline.

All differences due to the contrast parameter are below 0.5%. This supports our decision to rely on a heuristic setting of the contrast parameter for lossless compression, rather than spending computational resources on trying to optimize it. Results in Table 5.5 used second-order EED with iteration mode 0, because fine-tuning the contrast parameter would be even more costly in mode 1.

5.6 Conclusion

PDE-based inpainting has previously been shown to have a strong potential for lossy image compression, especially at high compression rates. We demonstrated that this approach also holds promise for lossless compression of 3D medical images. In particular, we propose a codec that beats state-of-the-art alternatives by combining a simple yet efficient to code initial inpainting mask with iterative reconstruction and coding of residuals, as well as a separate coding of empty space. In the future, we are planning to extend our work to exploit redundancies along the fourth axis that arises in diffusion MRI, i.e., orientation of the diffusion gradient [7]. This will require operating on the space of positions and orientation.

Acknowledgement

This research was supported by the German Academic Exchange Service (DAAD). The brain MR images were kindly provided by Tobias Schmidt-Wilcke, University of Düsseldorf. The foot CT dataset is courtesy of Philips Research.

Chapter 6

Combining Image Space and q-Space PDEs for Lossless Compression of Diffusion MR Images

Diffusion Magnetic Resonance Imaging is a non-invasive imaging technique used to study the microstructural organization of biological tissues, particularly the brain's white matter. It provides information about the diffusion of water molecules in tissues, which can reveal the structure and organization of cell membranes, axons, and other cellular components. Diffusion MRIs are 4D data with a large number of repeated 3D volumetric scans. This results into huge amounts of data. An efficient diffusion MRI-oriented compression codec is essential for managing large datasets, facilitating storage, and enabling faster data transfer and analysis. Commonly, diffusion MRI datasets are compressed using GZIP which is a lossless compression scheme based on dictionary coding and Huffman coding.

In this chapter we introduce a novel PDE-based lossless codec for diffusion MRI data. This new codec is designed on top of our previous codec on lossless PDE-based compression of 3D medical images by additionally exploiting smoothness in q-space. Q-space is diffusion MRI volume sample space indicating the direction and strength of a diffusion sensitizing gradient during the measurement. Our previously proposed 3D codec can be applied for each 3D individually. However, doing so does not exploit q-space redundancy, i.e., the fact that measurements for nearby gradient directions are usually similar. Moreover, it is very time consuming.

The core of the 4D codec is image-space inpainting with a novel approach of PDE-based inpainting in q-space. The codec uses a spatial PDE as in our 3D codec for the first few volumes until the sufficient amount of samples (diffusion-weighted images/ 3D dMRI scans) are available. Our 4D codec adaptively determines a suitable point for switching from spatial to q-space inpaiting. Then we employ q-space PDE-based inpainting. The general idea of q-space inpainting is that when a certain number of diffusion-weighted images with different gradient directions are known, it is possible to use them to predict images that correspond to a new direction which means a new 3D dMRI volume can be reconstructed given its gradient direction. For complexity reasons we perform q-space inpaiting with linear homogeneous diffusion. We also introduce our method for accelerated computation, it makes q-space inpainting of a 3D volume at a very reasonable computational cost. We further demonstrated the importance of including motion correction, and propose an efficient implementation that is based on affine image transformations via a common reference.

We evaluate our codec on two dMRI datasets. Both datasets have been collected from the same subject in the same scanner. For the first scan, the subject received the usual instruction of staying as still as possible during the acquisition. For the second scan, the subject was asked to move his head, to deliberately introduce motion artifacts. We demonstrate with results that compared to GZIP the proposed codec can reduce required disk space by more than 30%. It also beats lossless codecs from the JPEG family.

We additionally demonstrate the benefit of motion correction. In the first scan, in which the subject tried to keep his head still, compensating for small involuntary movements yields a benefit although it is slight. On the other hand, when strong head motion is present, in the second scan, motion correction has a big impact on the q-space inpainting and hence on the final file size.

The content of this chapter is published as a peer reviewed journal paper:

Jumakulyyev Ikram, Schultz Thomas "PDE-based Lossless Compression of 4D diffusion MRI" In: *Journal of Mathematical Imaging and Vision.* Springer Nature. 65 (2023), pp. 644-656. [46].

Contribution of the dissertation author: Conceptualization, methodology, literature review, algorithm implementation and validation, results interpretation and visualization, manuscript composition, validation experiment design.

6.1 Abstract

Diffusion MRI is a modern neuroimaging modality with a unique ability to acquire microstructural information by measuring water self-diffusion at the voxel level. However, it generates huge amounts of data, resulting from a large number of repeated 3D scans. Each volume samples a location in q-space, indicating the direction and strength of a diffusion sensitizing gradient during the measurement. This captures detailed information about the self-diffusion, and the tissue microstructure that restricts it. Lossless compression with GZIP is widely used to reduce the memory requirements. We introduce a novel lossless codec for diffusion MRI data. It reduces file sizes by more than 30% compared to GZIP, and also beats lossless codecs from the JPEG family. Our codec builds on recent work on lossless PDE-based compression of 3D medical images, but additionally exploits smoothness in q-space. We demonstrate that, compared to using only image space PDEs, q-space PDEs further improve compression rates. Moreover, implementing them with Finite Element Methods and a custom acceleration significantly reduces computational expense. Finally, we show that our codec clearly benefits from integrating subject motion correction, and slightly from optimizing the order in which the 3D volumes are coded.

6.2 Introduction

With the development of new medical imaging techniques, and constant refinement of existing ones, the associated storage requirements have been reported to grow exponentially each year [26]. This explains why medical image compression is an active area of research.

Our work belongs to the family of compression algorithms that are based on Partial Differential Equations (PDEs). The general idea behind this approach is to store a sparse subset of the image information, and to reconstruct the remaining image via PDE-based inpainting [29].

PDE-based compression has a long tradition for the lossy compression of natural images [29, 96] and videos [5, 56, 89]. The benefit of PDE-based approaches relative to transform-based codecs like JPEG [80] and JPEG2000 [101] has often been most pronounced at high compression rates [96]. Even though this strategy for lossy compression has also been transferred to three-dimensional images [84], in medical imaging, lossless compression is often preferred to ensure that all diagnostically relevant details are preserved. In some cases, it is even legally forbidden to apply lossy compression for medical image archival [52, 75].

We recently introduced a PDE-based codec for 3D medical images that stores the residuals between the original image and an intermediate PDE-based reconstruction to ensure that the final reconstruction is lossless, and we demonstrated that this strategy led to competitive compression rates [50]. In our current work, we extend this idea for the specific use case of image datasets from diffusion MRI.

Diffusion MRI (dMRI) [7, 60] is a variant of Magnetic Resonance Imaging in which diffusion sensitizing gradients are introduced into the measurement sequence. If the hydrogen nuclei that generate the MR signal undergo a net displacement along the gradient direction during the measurement, the signal is attenuated. Assuming that these displacements result from (self-) diffusion, comparing diffusion-weighted to non-weighted measurements permits computation of an apparent diffusion coefficient.

Taking measurements with different gradient directions captures the directional dependence of the diffusivity. It results from interactions between water and tissue microstructure and therefore carries information about structures that are much smaller than the MR image resolution. Important applications of dMRI include the detection of microstructural changes that are related to aging or disease, and the reconstruction of major white matter tracts, which is referred to as fiber tracking or tractography [44].

The large number of repeated measurements in diffusion MRI leads to large amounts of data. In practice, resulting image datasets are often compressed using GZIP [24]. In our previous work [50], we demonstrated that, compared to this, PDE-based lossless compression can further reduce the memory requirement of individual dMRI volumes by more than 25%. However, applying our codec to each 3D volume independently does not exploit the fact that measurements for nearby gradient directions are usually similar. Moreover, it is relatively time consuming.

In our current work, we address both of these limitations by combining the previous idea of lossless compression via image-space inpainting with a novel approach of PDE-based inpainting in q-space, which is the space spanned by diffusion sensitizing gradient directions and magnitudes. We find that predictions from linear diffusion in q-space can be made with low computational effort, and are strong enough to further improve compression rates.

The remainder of our work is organized as follows: Section 5.3 provides the required background and discusses prior work on 4D image compression. Section 6.4 introduces the components of our proposed codec. Section 6.5 demonstrates that the resulting compression rates exceed those of several baselines and investigates the effects of specific design choices. Section 6.6 concludes with a brief discussion.

6.3 Background and Related Work

We will now introduce the main ideas behind diffusion PDE-based image inpainting and compression (Section 6.3.1), clarify the foundations of diffusion MRI and

q-space (Section 6.3.2), and briefly review the literature on 4D medical image compression (Section 6.3.3).

6.3.1 Diffusion PDE-based Inpainting and Compression

Inspired by their use for modeling physical phenomena, Partial Differential Equations (PDEs) have a long tradition for solving problems in image processing. In particular, the PDE describing heat diffusion has provided a framework for image smoothing and inpainting [10, 16, 42, 72, 79, 82, 110].

The heat equation captures the relationship between temporal changes in a temperature $\partial_t u$ and the divergence of its spatial gradient ∇u,

$$\partial_t u = \operatorname{div}(D \cdot \nabla u) \,, \tag{6.1}$$

where D is the thermal diffusivity of the medium. In a homogeneous and isotropic medium, the diffusivity D is a constant scalar. In a non-homogeneous isotropic medium, D would still be a scalar, but depend on the spatial location. In an anisotropic medium, heat dissipates more rapidly in some directions than in others. In that case, D is a symmetric positive definite matrix that is referred to as a diffusion tensor.

In image processing, the gray value at a certain location is interpreted as a temperature u, and Equation (6.1) is coupled with suitable boundary conditions. For image smoothing,

$$\begin{aligned} \partial_t u &= \operatorname{div}(D \cdot \nabla u), \quad \Omega \times (0, \infty) \,, \\ \partial_n u &= 0, \quad \partial\Omega \times (0, \infty) \,, \end{aligned} \tag{6.2}$$

where Ω is the image domain, and n is the normal vector to its boundary $\partial\Omega$. The original image $f : \Omega \to \mathbb{R}$ is used to specify an initial condition $u = f$ at $t = 0$. For increasing diffusion time t, u will correspond to an increasingly smoothed version of the image.

In image inpainting, values are known at a subset of pixel locations, and unknown values should be filled in. For this, a Dirichlet boundary condition is introduced, which fixes values at a subset K of pixel locations [29, 110]

$$\begin{aligned} \partial_t u &= \operatorname{div}(D \cdot \nabla u), \quad \Omega \backslash K \times (0, \infty) \,, \\ \partial_n u &= 0, \quad \partial\Omega \times (0, \infty) \,, \\ u &= f, \quad K \times [0, \infty) \,. \end{aligned} \tag{6.3}$$

and a steady-state is computed at which $\partial_t u \approx 0$. The ability of PDEs to reconstruct plausible images even from a very sparse subset of pixels made them useful for image compression [29, 89, 96].

Different choices of diffusivity D introduce considerable flexibility with respect to shaping the final result. Fixing $D = 1$ turns Equation (6.3) into second-order linear homogenous (LH) diffusion

$$\partial_t u = Lu, \quad \Omega \backslash K \times (0, \infty) \tag{6.4}$$

with $Lu = \Delta u$, where Δ denotes the Laplace operator, and the steady state satisfies the Laplace equation $\Delta u = 0$. Even though the resulting reconstructions suffer from singularities [29] and can often be improved by the more complex models discussed below, they have been used to design compression codecs for cartoon-like images

[68], flow fields [45], and depth maps [32, 38, 63]. Its simple linear nature and fast convergence to the steady-state also make LH diffusion an attractive choice for real-time video compression [5, 56].

Compared to LH diffusion, decreasing the diffusivity as a function of image gradient magnitude permits a better preservation of salient edges [17, 82]. This is referred to as nonlinear diffusion, since the results are no longer linear in the original image f. Rather than just decreasing the overall diffusivity close to edges, modeling D as an anisotropic diffusion tensor permits smoothing along edges, while maintaining or even increasing the contrast perpendicular to them. One widely used model is referred to as Edge-Enhancing Diffusion (EED) [109].

All PDEs that have been discussed up to this point are of second order. Fourth- and higher-order extensions have also been studied, both for smoothing [25, 37, 67, 114, 115] and for inpainting [49, 62]. In the simplest case, setting $Lu = -\Delta^2 u$ in Equation (6.4) leads to the biharmonic (BH) equation. In two and three dimensions, it does not suffer from the singularities that are present in the results of LH diffusion [19, 29], while preserving a simple linear nature. For this reason, BH has been considered for the design of compression codecs [3, 19, 86, 88]. However, it no longer satisfies a min-max principle [25] and it increases running time and sensitivity to quantization error.

Our own previous work [49] proposed an anisotropic fourth-order PDE in which a fourth-order diffusion tensor is constructed from the image gradient in a similar way as in second-order EED. We thus refer to it as Fourth-Order Edge-Enhancing Diffusion (FOEED). It was shown to result in more accurate inpainting results than second-order EED, and higher PDE-based compression rates, in several examples [50].

Our current work is concerned with compressing data from diffusion MRI, which is similar to hyperspectral images in that it contains a large number of values (channels) at each location [3]. However, the channels in hyperspectral images have a one-dimensional order, while our channels correspond to positions on a spherical shell in q-space, which we will now introduce.

6.3.2 Diffusion MRI

The signal in Magnetic Resonance Imaging is generated by the hydrogen atoms within water molecules. Their heat motion is referred to as self-diffusion, since it takes place despite a zero concentration gradient. The extent to which this motion is restricted in a cellular environment correlates with microstructural parameters such as cellular density or integrity. Moreover, in the white matter of the human brain, which contains the tracts that connect different brain regions, self-diffusion can occur more freely in the local direction of those tracts than perpendicular to it [8]. Therefore, measuring the apparent diffusion coefficient in different directions provides relevant information about small-scale structures that are below the image resolution of in vivo MRI.

This motivates the use of diffusion MRI. It goes back to the idea of measuring diffusion by introducing a pair of diffusion sensitizing magnetic field gradients into a nuclear magnetic resonance sequence [100]. Integrating it with spatially resolved Magnetic Resonance Imaging permits diffusion measurements at a voxel level [60]. Repeating the measurements with differently oriented gradients reveals a biologically relevant directional dependence in various tissue types, including muscle and the white matter of the brain [90].

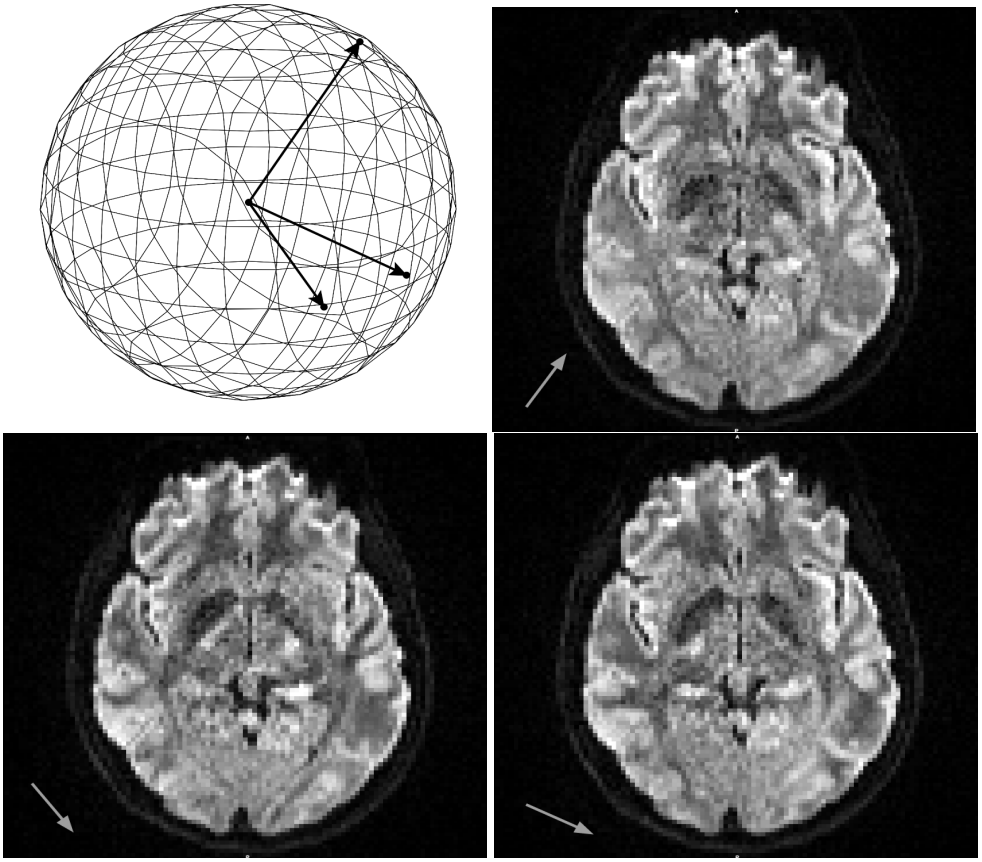


FIGURE 6.1: Illustration of three diffusion sensitizing gradient directions on a shell in q space, with equal $b = 700$ (top left). The three diffusion-weighted images have been measured with different gradient directions, as indicated at the bottom left of each image. Comparing them reveals the directional dependence of the dMRI signal.

Several key parameters of the diffusion sensitization can be summarized in the gradient wave vector

$$\mathbf{q} = \frac{1}{2\pi}\gamma\delta\mathbf{g}, \tag{6.5}$$

where γ is the gyromagnetic ratio of hydrogen nuclei in water, δ is the duration of the diffusion sensitizing gradients, and $\mathbf{g}$ corresponds to their direction and strength. The normalized MR echo magnitude $|E(\mathbf{q},\tau)|$ additionally depends on the time τ between the pair of gradient pulses. It is computed as the ratio between the corresponding diffusion-weighted measurement and an unweighted measurement with $\mathbf{q} = \mathbf{0}$. It is antipodally symmetric, $|E(-\mathbf{q},\tau)| = |E(\mathbf{q},\tau)|$.

The relevance of this q-space formalism derives from a Fourier relationship between $|E(\mathbf{q},\tau)|$ and the ensemble average diffusion propagator $\bar{P}(\mathbf{R},\tau)$, which specifies the probability of a molecular displacement $\mathbf{R}$ within a fixed diffusion time [14]. An alternative parameterization of the diffusion gradients is in terms of their direction and a factor $b = 4\pi^2\|\mathbf{q}\|^2(\tau - \delta/3)$, which also accounts for the fact that the diffusion weighting increases with the effective diffusion time $(\tau - \delta/3)$.

Due to practical constraints on the overall duration of dMRI measurements, the sampling of q-space is usually limited to one or several reference measurements with $\mathbf{q} = \mathbf{0}$, as well as one or a few shells with constant $\|\mathbf{q}\|$, and thus constant b.

This is illustrated in Figure 6.1. Such setups focus on the directional dependence of the signal, and typically strive for a uniform distribution of gradient directions on these shells [20]. Our codec assumes dMRI data with such a "shelled" structure, an assumption that is shared by well-established algorithms in the field [4].

6.3.3 4D Medical Image Compression

Many medical imaging modalities, including Magnetic Resonance Imaging, Computed Tomography, and ultrasound, can be used to image volumes repeatedly, in order to capture time-dependent phenomena such as organ motion, perfusion, or blood oxygenation. Considerable work has been done on lossless and lossy compression of the resulting 4D (3D plus time) image data. Much of it has borrowed from video coding, and has often involved motion compensation [51, 94], which is combined with wavelet transforms [9, 51, 57, 65, 116] or hierarchical vector quantization [78] for compression.

Almost all of these works have compared their compression rates to codecs from the JPEG family. We will also compare our codec to JPEG-LS, lossless JPEG, and JPEG2000. Additionally, we compare compression rates against GZIP [24] which, in conjunction with the Neuroimaging Informatics Technology Initiative (NIfTI) file format, is currently most widely used to compress diffusion MRI data in practice. To make this comparison fair, we also use Huffman coding or Deflate within our own codec, as opposed to computationally efficient alternatives that might further improve compression rates [27, 112].

Even though the volumetric images in diffusion MRI are also taken sequentially, their temporal order is less relevant than the q-space structure that was described above: Measuring with the same diffusion sensitizing gradients, but in a different order, should yield equivalent information, even though it permutes the temporal order. To the best of our knowledge, no codec has been proposed so far that exploits this very specific structure. There has been extensive work on compressed sensing for diffusion MRI (see [74, 105] and references therein), but with a focus on reducing measurement time, rather than efficient storage of the measured data.

Recent work has demonstrated the potential of deep learning for lossless compression of 3D medical images [76]. Extending this specifically for diffusion MRI is an interesting future direction. However, our PDE-based approach has the advantage of not requiring any training data. Since medical data is a particularly sensitive type of personal data, obtaining diverse large-scale datasets can be difficult, and the potential of model attacks that could cause data leakage is concerning [71, 77].

6.4 Proposed Lossless Codec

Traditional PDE-based image compression [29, 49, 50, 96] performs inpainting in image space, which relies on piecewise smoothness of the image. A key contribution of our current work is to additionally exploit the smoothness in q-space. As it can be seen in Figure 6.1, dMRI signals that are measured with similar gradient directions are correlated.

Our codec uses a spatial PDE for the first few volumes, which is described in more detail in Section 6.4.1. Once sufficiently many samples are available so that a q-space PDE, described in Section 6.4.2, produces stronger compression than the spatial PDE, we switch to it.

The q-space PDE assumes that all volumes are in correct spatial alignment, which might be violated in practice due to subject motion. For this reason, our codec includes a mechanism for motion compensation, described in Section 6.4.3. Our overall compressed file format is specified in Section 6.4.4.

6.4.1 Lossless 3D Spatial Codec

The initial few volumes are compressed with an image space PDE-based codec that follows our recent conference paper [50]. To make our current work self-contained, we briefly summarize the most relevant points, focusing on the forward, i.e., encoding direction. The decoding process just mirrors the respective steps. The codec is composed of three main parts: Data sparsification (initial mask selection), prediction (iterative reconstruction), and residual coding.

Initial Mask Selection: As an initial mask, our codec simply stores voxel intensities on a sparse regular grid. More specifically, for a given 3D input image of size $n_1 \times n_2 \times n_3$, the initial mask is chosen as a hexahedral grid consisting of voxels $(4i_1, 4i_2, 4i_3)$, where $i_j \in \{0, 1, \ldots, \lfloor \frac{(n_j-1)}{4} \rfloor\}$, $j \in 1, 2, 3$.

Most lossy PDE-based codecs select a mask adaptively [29, 96], which better preserves important image features such as edges and corners [68]. However, this introduces the need to store the locations of the selected pixels, which can be avoided by the use of fixed grids [38, 83]. In the context of lossless compression, we achieved higher compression rates by combining the latter strategy with iterative reconstruction.

Iterative Reconstruction: Making PDE-based compression lossless requires coding the differences between the original image and the PDE-based reconstruction, and is beneficial in terms of compression rates to the extent that those residuals are more compressible than the original image. In general, residuals become more compressible the more accurate the reconstruction is. Therefore, the overall compression rate can be increased by iteratively coding residuals of some pixels, and refining the remaining ones based on them.

Our previous work [50] explored different iterative schemes. The variant that is used here codes the residuals in all remaining face-connected neighbors of the current mask voxels, i.e., up to six voxels per mask voxel. Those neighbors become part of the mask for the next iteration, and the process continues until all voxels have been coded.

Among the PDEs that have been explored for inpainting, we currently consider the two that worked best in [50], i.e., traditional edge-enhancing diffusion (EED) [110] and our recent fourth-order generalization (FOEED) [49].

Residual Coding: Residuals are computed in modular arithmetic, so that they can be represented as unsigned integers. The final compression of the initial mask and the residuals is either done via a Huffman entropy encoder or the Deflate algorithm, depending on which gives the smaller output file size.

In cases where medical images contain a substantial amount of empty space, e.g., a background region with exactly zero image intensity, our previous work [50] found that coding it separately using run length encoding (RLE) can provide an additional benefit. Unfortunately, in dMRI, the background is perturbed by measurement noise, which renders this approach ineffective. Therefore, our current work does not include any dedicated empty space coding.

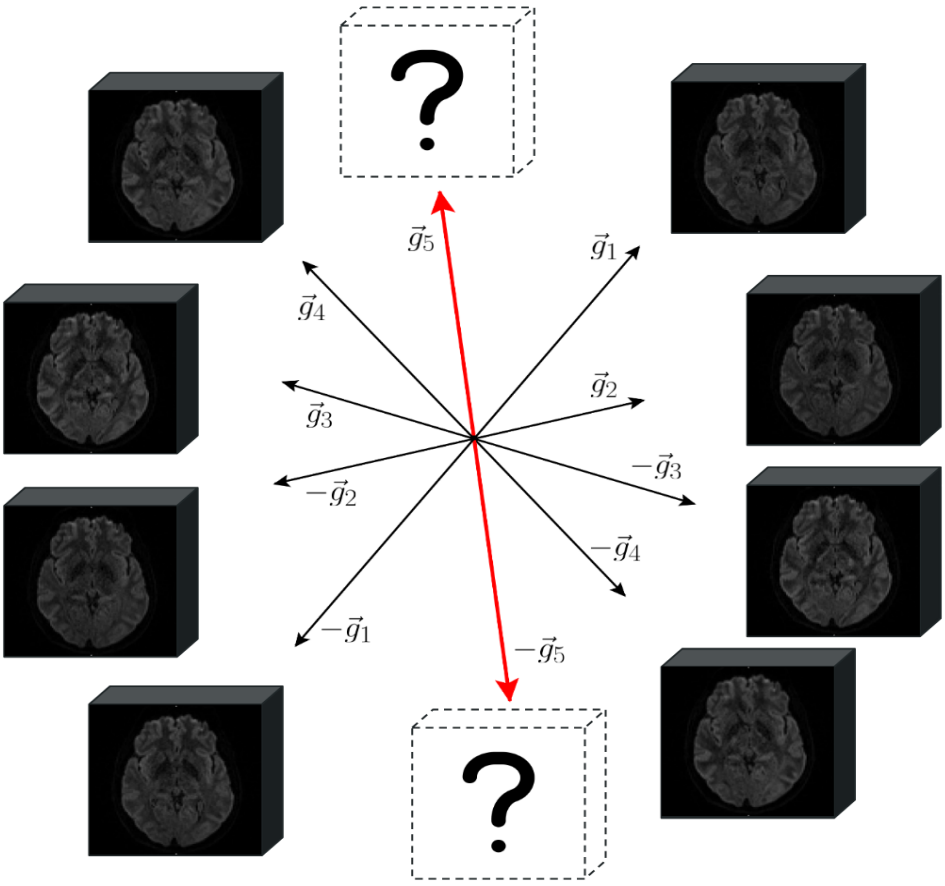


FIGURE 6.2: An example of q-space inpainting to predict a diffusion-weighted volume in gradient direction $\vec{g}_5$ (red double arrow) based on four known volumes, illustrated as filled volumes in directions $\vec{g}_i$ (black arrows).

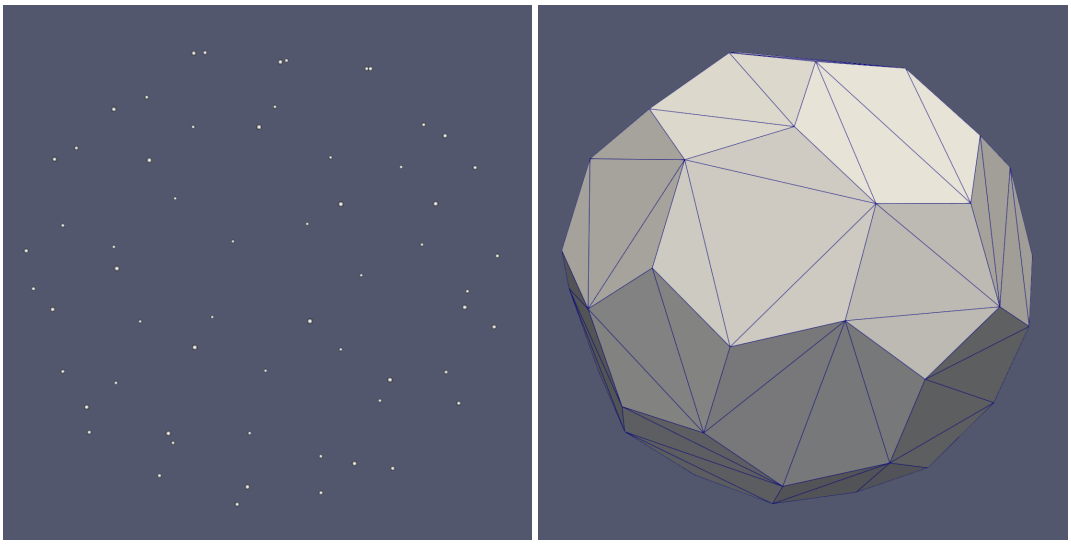


FIGURE 6.3: The q space sampling of the dMRI data used in our experiments (left), and the resulting triangulation that is used within the Finite Element Method.

6.4.2 PDE-based q-Space Inpainting

The general idea of q-space inpainting is illustrated in Figure 6.2: Once a certain number of diffusion-weighted images with different gradient directions are known, we can use them to predict images that correspond to a new direction. This happens at the voxel level, so that the prediction at a given location is entirely determined by values at the same location in the known images.

This can be understood as "flipping" the setup from Section 6.4.1, where the mask consisted of pixel locations, and the inpainting was repeated with an identical mask for each channel. Instead, the mask now specifies the known channels, and inpainting is repeated for each voxel in the volume.

Compressing Diffusion-Weighted Images

Since we assume that diffusion-weighted measurements are on spherical shells in q-space (Section 6.3.2), we inpaint with second-order linear homogeneous (LH) diffusion

$$\partial_t u = \Delta u \tag{6.6}$$

or fourth-order biharmonic (BH) smoothing

$$\partial_t u = -\Delta^2 u \tag{6.7}$$

on the sphere, where Δ is the Laplace-Beltrami operator.

Given that our samples do not form a regular grid, we numerically solve these equations using Finite Element Methods (FEM) [21, 66]. For this, we first construct a 3D Delaunay tessellation from the set of all gradient vectors $\mathbf{g}_i$ and their antipodal counterparts $-\mathbf{g}_i$, and then extract a triangular surface mesh from it. Figure 6.3 shows an example of the given vectors (left), and the resulting triangular mesh (right).

Similar to PDE-based inpainting in the image domain, we fix the known values by imposing Dirichlet boundary conditions at the vertices corresponding to the previously coded diffusion-weighted images, again accounting for antipodal symmetry

so that each known image determines the values of two vertices. Once a steady state has been reached, the values at locations corresponding to diffusion-weighted images that are yet to be coded can serve as predictions. Similar as before, we compute residuals with respect to those predictions in modular arithmetic, and apply Huffman coding or Deflate to them.

We found that, once a sufficient number of diffusion weighted images are available as a basis of q-space inpainting, its residuals become more compressible than those from iterative image space inpainting. Our codec adaptively determines a suitable point for switching from spatial to q-space predictions. After the first diffusion-weighted volume, it starts comparing the sizes of compressing subsequent volumes with the spatial codec (Section 6.4.1) to the size when using q-space inpainting and switches to it on the first volume where it is beneficial. To limit computational effort, the spatial codec is no longer tried for subsequent volumes.

Accelerated Computation

Since q-space inpainting happens at a voxel level, it should be repeated for each voxel of the 3D image. However, the computational cost of running the FEM solver for each voxel separately is extremely high. Fortunately, linearity of the PDEs and the fact that the Dirichlet boundary conditions are imposed on the same vertices for each voxel permit a significant speedup.

Formally, we can consider one time step of numerically solving Equation (6.6) or (6.7), at time t, as applying a discrete linear differential operator $\mathbf{D}$, which is determined by the vertices and connectivity of our triangular mesh, on a discrete input $\mathbf{u}^{(t)} \in \mathbb{R}^c$,

$$\mathbf{u}^{(t+1)} = \mathbf{D}\left[\mathbf{u}^{(t)}\right], \tag{6.8}$$

where c is the number of channels (q-space samples per voxel). The boundary conditions ensure that $u_{k_j}^{(t+1)} = u_{k_j}^{(t)}$ at positions k_j that correspond to the fixed (previously coded) channels.

The inpainting result is obtained as the fixed point $\mathbf{u}^{(\mathrm{FP})}$ as $t \to \infty$. It can be approximated by iterating $\mathbf{D}$ a sufficient number of times, resulting in an operator $\mathbf{D}_{\mathrm{FP}}$ that directly maps

$$\mathbf{u}^{(\mathrm{FP})} = \mathbf{D}_{\mathrm{FP}}\left[\mathbf{u}^{(0)}\right]. \tag{6.9}$$

$\mathbf{D}_{\mathrm{FP}}$ is still linear, and we observe that its kernel is the subspace corresponding to the unknown q-space samples, so that their initialization in $\mathbf{u}^{(0)}$ does not influence the steady state [89]. Therefore, we can rewrite Equation 6.9 as

$$\begin{aligned}\mathbf{D}_{\mathrm{FP}}\left[\mathbf{u}^{(0)}\right] =& u_{k_1}^{(0)}\mathbf{D}_{\mathrm{FP}}[\mathbf{e}_{k_1}] + u_{k_2}^{(0)}\mathbf{D}_{\mathrm{FP}}[\mathbf{e}_{k_2}] \\ &+ \ldots + u_{k_n}^{(0)}\mathbf{D}_{\mathrm{FP}}[\mathbf{e}_{k_n}],\end{aligned} \tag{6.10}$$

where $\mathbf{e}_{k_j}$ are the indicator vectors of the known samples k_j.

In other words, by computing $\mathbf{w}_{k_j} = \mathbf{D}_{\mathrm{FP}}[\mathbf{e}_{k_j}]$, we can obtain weight vectors that specify how the known values are combined to predict the unknown ones. They are analogous to the "inpainting echoes" that have been computed in previous work [69] for the purpose of optimizing tonal data. Omitting the irrelevant initialization of the unknown values from the input $\mathbf{u}^{(0)}$, and the known values from the output $\mathbf{u}^{(\mathrm{FP})}$ yields a weight matrix $\mathbf{W}$ of shape $m \times n$ for n known and m unknown values.

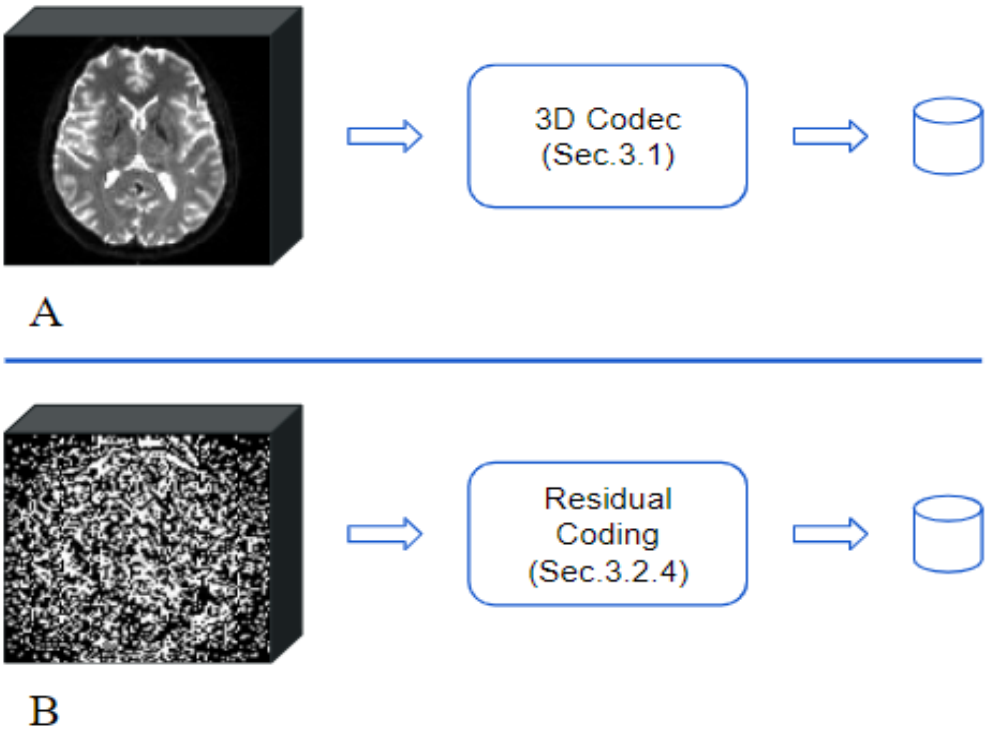


FIGURE 6.4: Top (A): Compression of the first $b = 0$ volume using the recently proposed lossless 3D codec [50]. Bottom (B): Compression of the remaining $b = 0$ volumes using residuals in modular arithmethic. Residuals are taken with respect to the first $b = 0$ volume, after motion correction.

We compute the coefficients of $\mathbf{W}$ by running the FEM n times. In the jth run, we set the value corresponding to k_j to one, all remaining values to zero. After numerically solving the Laplace or Biharmonic PDE, the values at the m unique vertices that correspond to unknown DWIs yield the jth column of $\mathbf{W}$.

Finally, $\mathbf{W}$ allows us to make efficient predictions in each voxel, by simply multiplying it to a vector that contains the intensities in that voxel from the previously coded diffusion gradients.

Implementation Details and Running Times

We numerically solve Equations (6.6) and (6.7) via the open-source FEM solver package FEniCSx [66]. For implementation details, we refer to its tutorials [58]. Applying this solver to each voxel of a $104 \times 104 \times 72$ volume takes close to two and four hours, respectively, for LH and BH PDEs on a single 3.3 GHz CPU core.

The acceleration from the previous section reduces this to only 1.6 s and 2.4 s per volume, respectively. This includes the time for building a Delaunay tessellation, which is computed with SciPy [108], and extracting a surface mesh from it using the `BoundaryMesh` method from FEniCSx.

Compressing $b = 0$ Images

Our general approach simplifies for unweighted volumes with $b = 0$. Again, the first of them is compressed using the spatial codec. If multiple $b = 0$ images were

acquired to increase the signal-to-noise ratio, our codec compresses the remaining ones by taking the respective residuals with respect to the first $b = 0$ volume, as illustrated in Figure 6.4.

6.4.3 Motion Correction

Subject motion commonly occurs during the lengthy dMRI acquisitions, and is typically accounted for by applying motion correction based on image registration [43]. We also include this step in our codec, since inpainting in q-space requires a correct spatial alignment of all 3D volumes so that predictions are based on information from the same location within the brain.

We implement motion correction as follows:

1. We perform affine registration of each volume to the same $b = 0$ volume, which is used as a common reference. This yields a transformation matrix $\mathbf{T}_{X \to b_0}$ which aligns DWI volume X to the $b = 0$ reference.

2. When predicting a DWI volume P, we transform all known volumes X via the affine transformation $\mathbf{T}^{-1}_{P \to b_0} \circ \mathbf{T}_{X \to b_0}$, which can be computed from the transformations in Step 1.

3. In addition to resampling each known volume X, we re-orient its diffusion gradient direction $\mathbf{g}_X$ according to the rotational part of the transformation in Step 2. Omitting this step would lead to incorrect relative orientations of diffusion gradient directions [61], which could again reduce accuracy of q-space inpainting.

Transforming images via a common reference allows us to align them without having to perform image registration on all pairs of volumes. This saves considerable computational effort. Combining the two transformations and applying them in a single step also reduces computational effort, and simultaneously reduces image blurring compared to a two-stage implementation that would involve interpolating twice.

In addition to the computational expense, motion correction incurs the cost of having to store the affine matrices $\mathbf{T}_{X \to b_0}$ along with the compressed data. Experimental results in Section 6.5.4 will demonstrate that this storage cost is outweighed by the increase in compression rate when q-space inpainting properly accounts for motion.

Subject movement correction and B-matrix reorientation are done using the freely available FSL tools [43] and the DIPY imaging library [31], respectively. A practically relevant implementation detail concerns boundary effects. As illustrated in Figure 6.5, missing information can enter the field of view when applying image transformations. We found that q-space inpainting near the boundary of the domain works more reliably if we resolve these cases with nearest neighbor extrapolation, rather than with zero padding.

6.4.4 Compressed File Format

In our current implementation, the relevant data is spread over multiple files whose sizes are added when computing compression rates.

The volumes that are compressed with the 3D lossless codec (Section 6.4.1) are stored with the same header as in [50]. Stated briefly, it contains the original minimum and maximum voxel values (4 bytes), sizes of the compressed data streams

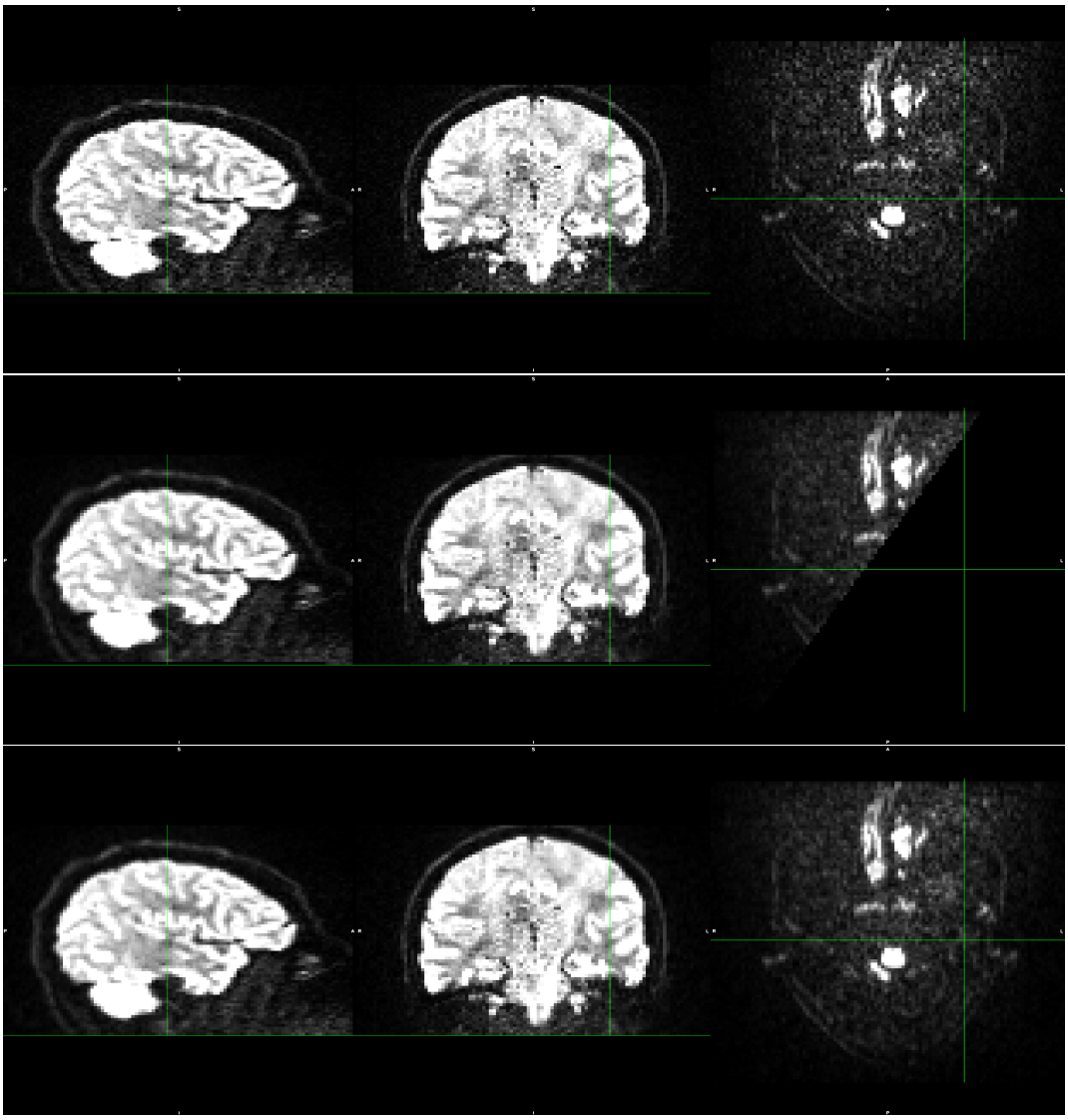


FIGURE 6.5: Boundary effects in volume alignment. Top: Original DWI volume. Center and Bottom: Motion corrected with zero padding and nearest neighbor extrapolation, respectively.

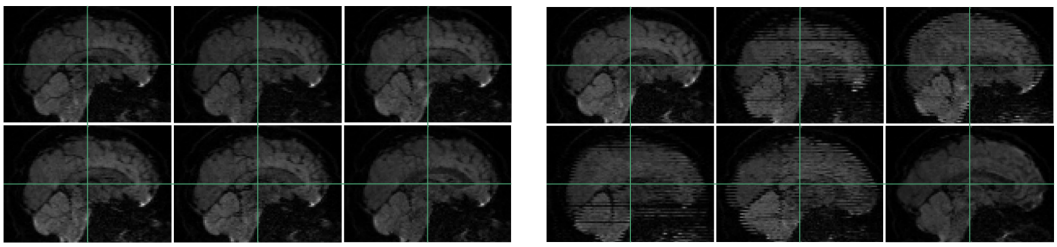


FIGURE 6.6: Example images from our two dMRI datasets, without deliberate head motion (left) and with strong motion artifacts (right). In each case, six corresponding sagittal slices from different diffusion weighted images (DWIs) are shown. Note that subject motion leads to spatial misalignments between DWIs, but also to artifacts within individual images.

for zero voxel binary mask and mask intensities (8 bytes), the diffusivity contrast parameter (4 bytes), the type of PDE (2 bits), the dilation mode (1 bit), and the types of encoding for mask intensities and residuals (2 bits).

For each volume that is compressed with q-space inpainting, the header contains the original minimum and maximum voxel values (4 bytes), the type of PDE (2 bits), the type of encoding for the residuals (1 bit), and the volume number in the original order (2 bytes).

Mask and residual values themselves are stored after compression with pure Huffman coding or Deflate, depending on what gave a smaller file size.

In addition, we store the NIfTI header (348 bytes) as well as files containing b values and gradient vectors in their original ASCII formats. For simplicity, affine transformations for motion correction are also kept in the ASCII format generated by FSL FLIRT [43].

6.5 Results and Discussion

6.5.1 Data

We evaluate our codec on two dMRI datasets that were made publicly available by Koch et al. [55], and are specifically suited to investigate the impact of subject motion compensation. Both datasets have been collected from the same subject (male, 34 years) in the same scanner, a 3T MAGNETOM Skyra (Siemens Healthcare, Erlangen, Germany), with an identical measurement sequence. For the first scan, the subject received the usual instruction of staying as still as possible during the acquisition. For the second scan, the subject was asked to move his head, to deliberately introduce motion artifacts.

From these datasets, we use the five non-diffusion weighted ($b = 0$) MRI scans each, as well as 30 diffusion weighted images ($b = 700\,\mathrm{s/mm^2}$, diffusion gradient duration $\delta = 334\,\mathrm{ms}$, spacing $\tau = 445\,\mathrm{ms}$). Each image consists of $104 \times 104 \times 72$ voxels with a resolution of $2 \times 2 \times 2\,\mathrm{mm^3}$. The data, and the effects of subject motion, are illustrated in Figure 6.6.

6.5.2 DTI Baseline

We compare the signal predictions from our q-space PDE to a simple baseline, which is derived from the Diffusion Tensor Imaging (DTI) model. DTI is widely used in practice, due to its relative simplicity and modesty in terms of scanner hardware and measurement time.

It rests on the assumption that the diffusion propagator $\bar{P}(\mathbf{R}, \tau)$ is a zero-mean Gaussian whose covariance matrix is proportional to the diffusion tensor $\mathbf{D}$, a symmetric positive definite 3×3 matrix that characterizes the local diffusion [7]. The signal model in DTI relates the diffusion-weighted signal $S(\hat{\mathbf{g}}, b)$ for a given b-value and gradient vector direction $\hat{\mathbf{g}} = \mathbf{g}/\|\mathbf{g}\|$ to the unweighted signal S_0 according to

$$S(\hat{\mathbf{g}}, b) = S_0 \mathrm{e}^{-b\hat{\mathbf{g}}^T \mathbf{D} \hat{\mathbf{g}}}. \tag{6.11}$$

Fitting $\mathbf{D}$ requires at least one reference MR image S_0, plus diffusion-weighted images in at least six different directions, which are usually taken with a fixed non-zero b-value. Equation (6.11) can then be used to predict the diffusion-weighted signal in any desired direction. In our experiments, we compare our PDE-based to DTI-based predictions that account for the same set of known measurements.

TABLE 6.1: Compressed file sizes from separate PDE-based compression of each 3D scan (baseline), from different variants of our proposed lossless codec, as well as from GZIP and lossless codecs from the JPEG family. For hybrid codecs, the split indicates the number of volumes coded with q-space or spatial inpainting, respectively.

Codec Variant	Scan 1: No deliberate motion			Scan 2: Strong head motion		
	Split	Size (bytes)	Over R-IEED-1	Split	Size (bytes)	Over R-IEED-1
R-IEED-1		16022666	-		16082537	-
R-IFOEED-1		15955826	+0.42%		16019913	+0.39%
qLH ∘ R-IFOEED-1	27/4	14984472	+6.50%	26/5	15570493	+3.18%
qLH ∘ R-IEED-1	27/4	14991732	+6.43%	26/5	15578604	+3.13%
qBH ∘ R-IFOEED-1	27/4	15032384	+6.18%	26/5	15681354	+2.50%
qBH ∘ R-IEED-1	27/4	15039644	+6.14%	26/5	15689465	+2.44%
DTI ∘ R-IFOEED-1	24/7	15099213	+5.76%	24/7	15854216	+1.42%
DTI ∘ R-IEED-1	24/7	15108244	+5.71%	24/7	15861176	+1.38%
GZIP		21841701	-36.32%		21819641	-35.67%
JPEG		17885953	-11.63%		17905933	-11.34%
JPEG-LS		17921807	-11.85%		17893931	-11.26%
JPEG2000		15993453	+0.18%		15980005	+0.64%

6.5.3 Comparing Lossless Codecs for Diffusion MRI

A comparison of file sizes that can be achieved on our two test datasets with different lossless codecs is provided in Table 6.1. As a baseline, the first two rows show results from coding each 3D volume independently with our recently proposed PDE-based codec [50], using second-order (R-IEED-1) and fourth-order anisotropic diffusion (R-IFOEED-1). Additional savings of other codecs with respect to R-IEED-1 are given in percent.

The second block in Table 6.1 shows results from several variants of our proposed new codec, which adaptively combines inpainting in q-space and image space. Highest compression rates were achieved when combining linear homogeneous (LH) diffusion in q-space with R-IFOEED-1 in image space, closely followed by R-IEED-1. Biharmonic (BH) diffusion in q-space also produced useful, but slightly weaker results.

Both q-space diffusion approaches achieved better compression than predictions from DTI (Section 6.5.2). This could be due to the fact that the quadratic model of diffusivities in Equation (6.11) is known to be an oversimplification in many parts of the brain [2], and the PDE-based approaches provide more flexibility.

DTI requires independent coding of at least seven 3D volumes, which led us to fix this split in our experiments. PDE-based imputation makes it possible to switch to q-space inpainting earlier, and our adaptive selection does so after four volumes in the low-motion data, after five volumes in the data with strong motion.

Switching to q-space inpainting also speeds up our codec. Our implementation of R-IEED-1 and R-IFOEED-1 requires approximately 478 s and 6185 s, respectively, for one volume on a single 3.3 GHz CPU core. Even though it would be possible

TABLE 6.2: Compressed file sizes when omitting motion compensation, and the relative benefit from motion correction.

Codec Variant	Scan 1: No deliberate motion			Scan 2: Strong head motion		
	Split	Without Correction	Benefit	Split	Without Correction	Benefit
qLH ∘ R-IFOEED-1	27/4	15113407	+0.85%	16/15	16456557	+5.38%
qLH ∘ R-IEED-1	27/4	15122438	+0.86%	16/15	16478088	+5.46%
qBH ∘ R-IFOEED-1	27/4	15302805	+1.77%	16/15	16582150	+5.43%
qBH ∘ R-IEED-1	27/4	15311836	+1.78%	16/15	16603681	+5.51%
DTI ∘ R-IFOEED-1	24/7	15396194	+1.93%	24/7	16946648	+6.45%
DTI ∘ R-IEED-1	24/7	15405225	+1.93%	24/7	16955494	+6.45%

to further optimize this, exploiting linearity in qLH and qBH, as described in Section 6.4.2, significantly lowers the intrinsic computational complexity, so that even a straightforward implementation only requires 1.64 s and 2.4 s per volume, respectively.

It can be seen in Figure 6.6 that subject motion during different phases of the acquisition leads to different types of artifacts. Results in Table 6.1 include the motion correction described in Section 6.4.3, which compensates spatial misalignments of different scans. However, motion can also lead to signal dropouts or to distortions within scans, which our current codec does not explicitly account for. This explains why q-space inpainting is less effective on the second as compared to the first scan. However, even on this challenging dataset that exhibits unusually strong artifacts, q-space inpainting still provides a benefit compared to all other alternatives.

Finally, Table 6.1 shows results from several other lossless codecs for comparison. GZIP is most widely used in practice, but the resulting files are more than 35% larger than those from our proposed codec. Among the lossless codecs from the JPEG family, JPEG2000 is the only one that outperforms R-IEED-1 for per-volume compression, and only by a small margin. Our new hybrid methods that combine image space and q-space inpainting always performed best.

6.5.4 Benefit from Motion Correction

Table 6.2 investigates the benefit of motion correction (Section 6.4.3) by showing file sizes when removing motion correction from our codec, and comparing the results to ones with motion correction (Table 6.1), indicating the benefit in percent.

Even on the first scan, in which the subject tried to keep his head still, compensating for small involuntary movements yields a slight benefit. The effect is largest when imputing via qBH and DTI. This might be explained by the fact that qLH satisfies the min-max principle, which makes it more robust against inaccuracies in its inputs, and provides another argument in its favor.

When strong head motion is present (second scan), restoring a correct voxel alignment via motion correction becomes essential for q-space inpainting. Without it, the switch to q-space imputation happens much later, and the overall file size is larger than when coding each volume independently. This is explained by the fact that our codec always applies difference coding to the $b = 0$ images, and that this becomes detrimental when those images are strongly misaligned.

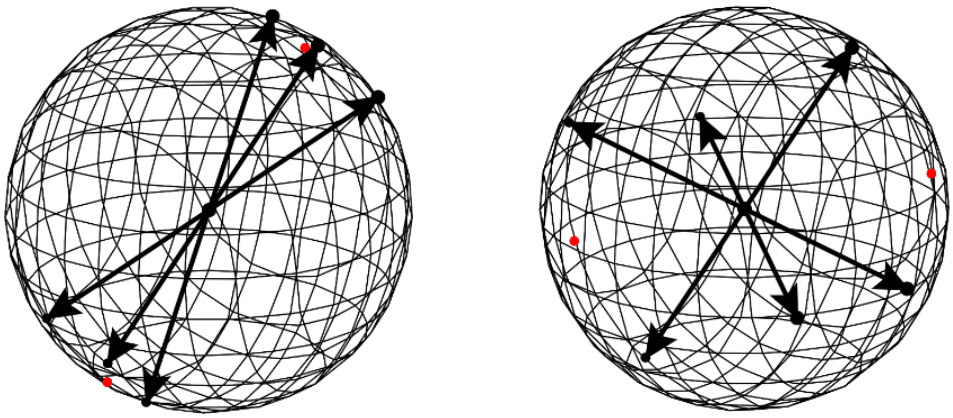


FIGURE 6.7: Given a set of previously coded DWIs, the *closest* strategy (left) selects the volume whose gradient vector has the smallest angular distance from the known ones, to maximize expected prediction accuracy. The *furthest* strategy (right) maximizes the angular distance, aiming for a more uniform coverage of the sphere for subsequent steps. The sketch shows the directions selected in the first three steps as black double arrows, the fourth direction as a red dot.

TABLE 6.3: Compressed file size for scan 1 (without strong motion) when ordering the diffusion-weighted images differently. This affects the accuracy of q-space imputation.

Codec Variant	Closest Selection	Furthest Selection
qLH ∘ R-IEED-1	15031703	14991732
qBH ∘ R-IEED-1	15153333	15039644
qDTI ∘ R-IEED-1	15290105	15108244

6.5.5 Effect of Re-ordering DWIs

Since q-space imputation relies on the previously (de)coded diffusion weighted images, its accuracy depends on the order in which we process the gradient directions.

Two contradictory greedy strategies are illustrated in Figure 6.7: Always selecting the *closest* gradient direction, i.e., the one with the smallest angular distance from the already known ones, can be expected to result in the most accurate prediction, in the same spirit as our spatial codec (Section 6.4.1) iteratively grows a mask of known pixels around an initial set of seed points.

On the other hand, the spatial codec starts with a seed mask that covers the full domain sparsely, but uniformly. Achieving something similar motivates selecting the gradient direction that is *furthest* from any of the known ones. Even though this strategy can be expected to lead to lower accuracy, and therefore to less compressible residuals in the first few iterations, later iterations might benefit from the more uniform coverage of the overall (spherical) domain.

Table 6.3 presents the effect of these two selection strategies on final file sizes. The results are from the first scan, without strong motion. Overall, greedily selecting the

furthest gradient vector gives slightly smaller overall file sizes. Therefore, this is the strategy that we followed in all other experiments.

6.6 Conclusion

In this work, we introduced a PDE-based lossless image compression codec that explicitly exploits both the spatial and the q-space structure in diffusion MRI. To our knowledge, it is the first codec that has been tailored to this type of data. We demonstrated a clear improvement over PDE-based codecs that treat each volume separately, and over other established baselines including GZIP and spatial codecs from the JPEG family.

We evaluated several variants of our codec, and found that q-space predictions with linear homogeneous diffusion permitted the highest compression rates among them. With our proposed method for accelerated computation, it could also be applied at a very reasonable computational cost. We further demonstrated the importance of including motion correction, and propose an efficient implementation that is based on affine image transformations via a common reference. Finally, we found that the order of coding the diffusion-weighted volumes had a relatively minor effect, but that a greedy strategy that strives to cover the sphere as uniformly as possible provides a small benefit.

In the future, one might attempt to replace the switching between image space and q-space inpainting with a PDE that jointly operates on the product space. However, this is likely to substantially increase the computational effort, and introduces the issue of properly balancing image space and q-space diffusion. Similarly, employing nonlinear PDEs for q-space predictions might further increase compression rates, but is likely to cause a high computational cost.

Chapter 7

Conclusion

7.1 Contributions

7.1.1 Image inpainting

Image inpainting is a technique used in image processing to fill in missing or corrupted parts of an image based on the surrounding areas or known subset of pixels. It's a form of image restoration where the goal is to generate visually coherent and realistic information in the areas where data is missing. Diffusion PDEs provide a mathematical foundation for modeling the process of propagating information and can be adapted for image inpainting. By leveraging the principles of diffusion and adapting PDEs to guide the inpainting process, algorithms can effectively estimate and fill in missing or corrupted parts of an image while preserving important structural features.

As a contribution to this topic we introduced a novel fourth-order PDE for edge enhancing diffusion (FOEED), steered by a fourth-order diffusion tensor. The diffusion tensor is designed from the regularized local image gradient information. The ingredients of the diffusion tensor are the diffusivity function and contrast parameter. They define the direction and magnitude of the diffusion. We provide a theoretical unifying framework to define all anisotropic fourth-order diffusion based methods which have been proposed so far. In implementation, we used a fast semi-iterative scheme, and demonstrated that it achieved improved accuracy in several inpainting tasks, including reconstructing images from a small fraction of pixels, or removing scratches.

Our main motivation for using fourth-order diffusion in this context is the increased smoothness of results compared to second-order PDEs [114], which we expected to result in visually more pleasant reconstructions. We also demonstrated that the reconstruction quality difference between the second-order edge enhancing diffusion-based PDE (EED) increased as we decrease the fraction of known pixels.

7.1.2 3D medical image compression

In medical image compression, preserving every detail of an image is crucial for accurate diagnosis and analysis by healthcare professionals. The lossless compression ensures that the original image can be perfectly reconstructed from the compressed version, making it superior for medical image storage and transmission.

PDE-based inpainting has previously been shown to have a strong potential for lossy image compression, especially at high compression rates. We demonstrated that this approach also holds promise for lossless compression of 3D medical images. In particular, we propose a codec that beats state-of-the-art alternatives by

combining a simple yet efficient to code initial inpainting mask with iterative reconstruction and coding of residuals, as well as a separate coding of empty space.

As an initial mask we use the hexahedral grid consisting of voxels of every fourth in each of all three spatial dimensions. It releases us from substantial effort in selecting a suitable small subset of voxel locations as an inpainting mask from which the original volume can be approximately reconstructed and the cost of coding the initial mask, i.e., initial voxel locations and voxel values. Our codec exploits also the fact that many medical images contain a substantial amount of empty space, which typically yields the lowest possible intensity, and can be coded efficiently as a run length encoded binary mask. We perform residual coding and iterative reconstruction to achieve a lossless codec and to compensate coarse reconstruction with our initial mask.

7.1.3 Diffusion MR image compression

Diffusion MRI visualizes and quantifies the diffusion of water molecules within biological tissues. The dMRI data are 4D data with many repeated 3D volumetric scans. Volumetric scans, diffusion MRI volumes are three dimensional spatial data. The spatial redundency is exploited in our 3D medical image codec [50]. The 3D codec [50] can be applied for each 3D volume individually to compress 4D dMRI data. We observed that exploiting redundancy in the fourth dimension brings more compression benefit than applying the 3D spatial codec [50] individually. The fourth dimension in dMRI, q-space is for gradient sample indicating the direction and strength of a diffusion sensitizing gradient in MRI during the measurement of water molecules in a biological tissue. Q-space redundancy comes with the fact that measurements for nearby gradient directions are usually similar. It results in very similar MRI volumes which can be exploited.

We introduced a PDE-based lossless image compression codec that explicitly exploits both the spatial and the q-space structure in diffusion MRI. To our knowledge, it is the first codec that has been tailored to this type of data. We demonstrated a clear improvement over PDE-based codecs that treat each volume separately, and over other established baselines including GZIP and spatial codecs from the JPEG family.

We evaluated several variants of our codec, and found that q-space predictions with linear homogeneous diffusion permitted the highest compression rates among them. With our proposed method for accelerated computation, it could also be applied at a very reasonable computational cost. We further demonstrated the importance of including motion correction, and propose an efficient implementation that is based on affine image transformations via a common reference. Finally, we found that the order of coding the diffusion-weighted volumes had a relatively minor effect, but that a greedy strategy that strives to cover the sphere as uniformly as possible provides a small benefit.

Last but not least, the potential of deep learning for lossless compression of natural images has been demonstrated in the recent works [73, 118]. However, training deep learning based models to develop a codec for medical images will have to account for data privacy, which makes it difficult to obtain large-scale training data and raises concerns about inference attacks [77]. Hence, we kept deep learning based approches out of this thesis scope.

7.2 Future work

7.2.1 Anisotropic fourth-order diffusion

Our proposed FOEED anisotropic fourth-order diffusion model is specific for image inpainting and compression purposes. The FOEED model is based on a single edge direction at each pixel, extracted via a traditional second-order structure tensor. It can be investigated as a potential future research work to combine this with approaches for the estimation of complex structures such as crossings or bifurcations [1, 98], and with their improved reconstruction, e.g., by operating on the space of positions and orientations [11, 22, 28].

Although it is designed specifically for inpainting and compression, the FOEED can be applied for image denoising purposes. For denoising, it is suggested to investigate the FOEED model's diffusivity function. There is a set of different diffusivity functions investigated for inpainting with the FOEED model in our work which can be investigated for denoising applications as a starting point. The FOEED model employs the Charbonnier diffusivity function as it has been proved to be useful for inpainting applications.

Anisotropic fourth-order diffusion has the potential for super-resolution and image reconstruction. For example, this work [39] employs isotropic fourth-order diffusion mode as a part of its super-resolution model. The isotropic model can be replaced by an anisotropic counterpart with a specifically designed diffusion tensor and diffusivity function.

In work [91], the second-order anisotropic diffusion is employed to design a 3D seismic data fusion and filtering. The PDE model can be extended to the fourth-order anisotropic model and its performance can be investigated compared to the proposed second-order counterpart.

7.2.2 Medical image compression

Both our proposed 3D lossless medical image and 4D dMRI codecs have the potential to extend to almost/near lossless medical image compression which focuses on measurement noise as a loss. We use a regular mask in our proposed codecs as it is simple and requires no location signaling to the final bitstream as an overhead. With almost lossless schemes one can re-design regular masks based on tree structure so that more mask locations are gathered to the region of importance in a medical image. Then the obtained mask locations can be signaled as coding tree blocks which consist of 0s and 1s. This might be further compressed by Run length encoding. Given that dMR images change due to gradient direction, ROI is expected to be on the same locations. Finding ROI in one sample of q-space should be efficient enough in almost lossless compression of dMRI data. However, here one should introduce a metric to make sure that the diagnostically relevant structures are not affected. This metric can be structural similarity index (SSIM) which should be higher than pre-defined threshold during the encoding process of codec.

Further, for the work on dMRI codec, as a future work, one might attempt to replace the switching between image space and q-space inpainting with a PDE that jointly operates on the product space. However, this is likely to substantially increase the computational effort, and introduces the issue of properly balancing image space and q-space diffusion. Similarly, employing nonlinear PDEs for q-space predictions might further increase compression rates, but is likely to cause a high computational cost.

Our 4D dMRI codec is specifically designed for a single shell diffusion MRI dataset. In single shell diffusion MRI, only one set of diffusion gradients with a particular diffusion weightings, described by b-values, is used during image acquisition. The b-value represents the strength and timing of the diffusion-sensitizing gradients applied in the MRI sequence. Our dMRI codec can be extended to multi-shell diffusion MRI. Multi-shell diffusion MRI employs multiple sets of diffusion gradients with varying b-values during image acquisition. Different b-values probe different scales of diffusion in biological tissues. For each different b-values 4D dMRI codec can be applied seperately.

Bibliography

[1] Til Aach, Cicero Mota, Ingo Stuke, Matthias Mühlich, and Erhard Barth. "Analysis of Superimposed Oriented Patterns". In: *IEEE Trans. on Image Processing* 15.12 (2006), pp. 3690–3700.

[2] D. C. Alexander, G. J. Barker, and S. R. Arridge. "Detection and Modeling of Non-Gaussian Apparent Diffusion Coefficient Profiles in Human Brain Data". In: *Magnetic Resonance in Medicine* 48 (2002), pp. 331–340.

[3] Naoufal Amrani, Joan Serra-Sagristà, Pascal Peter, and Joachim Weickert. "Diffusion-based inpainting for coding remote-sensing data". In: *IEEE Geoscience and Remote Sensing Letters* 14.8 (2017), pp. 1203–1207.

[4] Jesper L.R. Andersson and Stamatios N. Sotiropoulos. "An integrated approach to correction for off-resonance effects and subject movement in diffusion MR imaging". In: *NeuroImage* 125 (2016), pp. 1063–1078.

[5] Sarah Andris, Pascal Peter, and Joachim Weickert. "A proof-of-concept framework for PDE-based video compression". In: *Picture Coding Symposium (PCS)*. IEEE. 2016, pp. 1–5.

[6] Johannes Ballé, Valero Laparra, and Eero P. Simoncelli. "End-to-end Optimized Image Compression". In: *Int'l Conf. on Learning Representations (ICLR)*. 2017.

[7] Peter J. Basser, James Mattiello, and Denis Le Bihan. "Estimation of the Effective Self-Diffusion Tensor from the NMR Spin Echo". In: *J. of Magnetic Resonance* B.103 (1994), pp. 247–254.

[8] Christian Beaulieu. "The basis of anisotropic water diffusion in the nervous system – a technical review". In: *NMR in Biomedicine* 15.7–8 (2002), pp. 435–455.

[9] Leila Belhadef and Zoulikha Mekkakia Maaza. "Lossless 4D medical images compression with motion compensation and lifting wavelet transform". In: *Int. J. Signal Process. Syst* 4.2 (2016), pp. 168–171.

[10] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. "Image inpainting". In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. 2000, pp. 417–424.

[11] Ugo Boscain, Roman A Chertovskih, Jean-Paul Gauthier, and AO Remizov. "Hypoelliptic diffusion and human vision: a semidiscrete new twist". In: *SIAM Journal on Imaging Sciences* 7.2 (2014), pp. 669–695.

[12] Susanne C Brenner, L Ridgway Scott, and L Ridgway Scott. *The mathematical theory of finite element methods*. Vol. 3. Springer, 2008.

[13] Marco Cagnazzo, Giovanni Poggi, and Luisa Verdoliva. "Region-based transform coding of multispectral images". In: *IEEE Trans. on Image Processing* 16.12 (Dec. 2007), pp. 2916–2926.

[14] P. T. Callaghan, C. D. Eccles, and Y. Xia. "NMR microscopy of dynamic displacements: k-space and q-space imaging". In: *Journal of Physics E* 21.8 (1988), pp. 820–822.

[15] Francine Catté, Pierre-Louis Lions, Jean-Michel Morel, and Tomeu Coll. "Image selective smoothing and edge detection by nonlinear diffusion". In: *SIAM Journal on Numerical analysis* 29.1 (1992), pp. 182–193.

[16] Tony F Chan and Jianhong Shen. "Nontexture inpainting by curvature-driven diffusions". In: *Journal of Visual Communication and Image Representation* 12.4 (2001), pp. 436–449.

[17] Pierre Charbonnier, Laure Blanc-Féraud, Gilles Aubert, and Michel Barlaud. "Deterministic edge-preserving regularization in computed imaging". In: *IEEE Trans. on Image Processing* 6.2 (1997), pp. 298–311.

[18] Pierre Charbonnier, Laure Blanc-Féraud, Gilles Aubert, and Michel Barlaud. "Two Deterministic Half-Quadratic Regularization Algorithms for Computed Imaging". In: *Int'l Conf. on Image Processing*. 1994, pp. 168–172.

[19] Yunjin Chen, René Ranftl, and Thomas Pock. "A bi-level view of inpainting-based image compression". In: *Proc. Computer Vision Winter Workshop*. Ed. by Zuzana Kúkelová and Jan Heller. 2014, pp. 19–26.

[20] Jian Cheng, Dinggang Shen, Pew-Thian Yap, and Peter J Basser. "Single-and multiple-shell uniform sampling schemes for diffusion MRI using spherical codes". In: *IEEE Transactions on Medical Imaging* 37.1 (2017), pp. 185–199.

[21] Vassillen Chizhov and Joachim Weickert. "Efficient Data Optimisation for Harmonic Inpainting with Finite Elements". In: *Int'l Conf. on Computer Analysis of Images and Patterns*. 2021, pp. 432–441.

[22] Giovanna Citti and Alessandro Sarti. "A cortical based model of perceptual completion in the roto-translation space". In: *Journal of Mathematical Imaging and Vision* 24.3 (2006), pp. 307–326.

[23] P. Deutsch. *RFC1951: DEFLATE Compressed Data Format Specification Version 1.3*. USA, 1996.

[24] P. Deutsch. *RFC1952: GZIP File Format Specification Version 4.3*. USA, 1996.

[25] Stephan Didas, Joachim Weickert, and Bernhard Burgeth. "Properties of Higher Order Nonlinear Diffusion Filtering". In: *Journal of Mathematical Imaging and Vision* 35.3 (2009), pp. 208–226.

[26] Ivo D Dinov. "Volume and value of big healthcare data". In: *Journal of medical statistics and informatics* 4 (2016), p. 3.

[27] Jarek Duda, Khalid Tahboub, Neeraj J. Gadgil, and Edward J. Delp. "The use of asymmetric numeral systems as an accurate replacement for Huffman coding". In: *Proc. IEEE Picture Coding Symposium (PCS)*. 2015, pp. 65–69.

[28] Erik Franken and Remco Duits. "Crossing-preserving coherence-enhancing diffusion on invertible orientation scores". In: *Int'l Journal of Computer Vision* 85.3 (2009), p. 253.

[29] Irena Galić, Joachim Weickert, Martin Welk, Andrés Bruhn, Alexander Belyaev, and Hans-Peter Seidel. "Image compression with anisotropic diffusion". In: *Journal of Mathematical Imaging and Vision* 31.2-3 (2008), pp. 255–269.

[30] Irena Galić, Joachim Weickert, Martin Welk, Andrés Bruhn, Alexander Belyaev, and Hans-Peter Seidel. "Towards PDE-based image compression". In: *Int'l Workshop on Variational, Geometric, and Level Set Methods in Computer Vision*. Springer. 2005, pp. 37–48.

[31] Eleftherios Garyfallidis, Matthew Brett, Bagrat Amirbekian, Ariel Rokem, Stefan Van Der Walt, Maxime Descoteaux, Ian Nimmo-Smith, and Dipy Contributors. "Dipy, a library for the analysis of diffusion MRI data". In: *Frontiers in Neuroinformatics* 8 (2014), p. 8.

[32] Josselin Gautier, Olivier Le Meur, and Christine Guillemot. "Efficient depth map compression based on lossless edge coding and diffusion". In: *2012 Picture Coding Symposium*. IEEE. 2012, pp. 81–84.

[33] Donald Geman and George Reynolds. "Constrained restoration and the recovery of discontinuities". In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 14.3 (1992), pp. 367–383.

[34] John B. Greer and Andrea L. Bertozzi. "H^1 Solutions of a class of fourth order nonlinear equations for image processing". In: *Discrete and Continuous Dynamical Systems* 10 (2003), pp. 349–366.

[35] David Hafner, Peter Ochs, Joachim Weickert, Martin Reißel, and Sven Grewenig. "FSI Schemes: Fast Semi-Iterative Solvers for PDEs and Optimisation Methods". In: *German Conf. on Pattern Recognition (GCPR)*. 2016, pp. 91–102.

[36] Mohammad Reza Hajiaboli. "A self-governing hybrid model for noise removal". In: *Pacific-Rim Symposium on Image and Video Technology*. Springer. 2009, pp. 295–305.

[37] Mohammad Reza Hajiaboli. "An anisotropic fourth-order diffusion filter for image noise removal". In: *International Journal of Computer Vision* 92.2 (2011), pp. 177–191.

[38] Sebastian Hoffmann, Markus Mainberger, Joachim Weickert, and Michael Puhl. "Compression of depth maps with segment-based homogeneous diffusion". In: *Int'l Conf. on Scale Space and Variational Methods in Computer Vision*. Springer. 2013, pp. 319–330.

[39] Shuying Huang, Yong Yang, and Guoyu Wang. "An Improved PDE Based Super-Resolution Reconstruction Algorithm". In: *Procedia Engineering* 29 (2012), pp. 2838–2842.

[40] David A Huffman. "A method for the construction of minimum-redundancy codes". In: *Proceedings of the IRE* 40.9 (1952), pp. 1098–1101.

[41] David A. Huffman. "A Method for the Construction of Minimum-Redundancy Codes". In: *Proceedings of the Institute of Radio Engineers* 40.9 (1952), pp. 1098–1101.

[42] Taizo Iijima. "Basic theory on normalization of a pattern (in case of typical one-dimensional pattern)". In: *Bulletin of the Electrotechnical Laboratory* 26 (1962), pp. 368–388.

[43] Mark Jenkinson, Peter Bannister, Michael Brady, and Stephen Smith. "Improved optimization for the robust and accurate linear registration and motion correction of brain images". In: *Neuroimage* 17.2 (2002), pp. 825–841.

[44] Derek K. Jones, ed. *Diffusion MRI: Theory, Method, and Applications*. 1st. United Kingdom: Oxford University Press, 2011.

[45] Ferdinand Jost, Pascal Peter, and Joachim Weickert. "Compressing flow fields with edge-aware homogeneous diffusion inpainting". In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 2198–2202.

[46] Ikram Jumakulyyev and Thomas Schultz. "Combining Image Space and q-Space PDEs for Lossless Compression of Diffusion MR Images". In: *Journal of Mathematical Imaging and Vision* 65 (2023), pp. 644–656.

[47] Ikram Jumakulyyev and Thomas Schultz. "Fourth-order anisotropic diffusion for inpainting and image compression". In: *Anisotropy Across Fields and Scales*. Springer, Cham, 2021, pp. 99–124.

[48] Ikram Jumakulyyev and Thomas Schultz. "Fourth-Order Anisotropic Diffusion for Inpainting and Image Compression". In: *Anisotropy Across Fields and Scales*. Mathematics and Visualization. Springer, 2021, pp. 99–123.

[49] Ikram Jumakulyyev and Thomas Schultz. "Fourth-Order Anisotropic Diffusion for Inpainting and Image Compression". In: *Anisotropy Across Fields and Scales*. Ed. by Evren Özarslan, Thomas Schultz, Eugene Zhang, and Andrea Fuster. Cham: Springer International Publishing, 2021, pp. 99–124. ISBN: 978-3-030-56215-1.

[50] Ikram Jumakulyyev and Thomas Schultz. "Lossless PDE-based Compression of 3D Medical Images". In: *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer. 2021, pp. 450–462.

[51] Ashraf A Kassim, Pingkun Yan, Wei Siong Lee, and Kuntal Sengupta. "Motion compensated lossy-to-lossless compression of 4-D medical images using integer wavelet transforms". In: *IEEE Trans. on Information Technology in Biomedicine* 9.1 (2005), pp. 132–138.

[52] Se-Kee Kil, Jong-Shill Lee, D Shen, J Ryu, E Lee, H Min, and S Hong. "Lossless medical image compression using redundancy analysis". In: *International Journal of Computer Science and Network Security* 6.1 (2006), pp. 50–56.

[53] Young-Seop Kim and William A Pearlman. "Lossless volumetric medical image compression". In: *Applications of Digital Image Processing XXII*. Vol. 3808. International Society for Optics and Photonics. 1999, pp. 305–312.

[54] Nahum Kiryati and Yuval Landau. "Dataset growth in medical image analysis research". In: *Journal of imaging* 7.8 (2021), p. 155.

[55] Alexandra Koch, Andrei Zhukov, Tony Stöcker, Samuel Groeschel, and Thomas Schultz. "SHORE-based detection and imputation of dropout in diffusion MRI". In: *Magnetic Resonance in Medicine* 82.6 (2019), pp. 2286–2298.

[56] H Köstler, M Stürmer, Ch Freundl, and U Rüde. "PDE based video compression in real time". In: *Tech. Rep. 07-11, Lehrstuhl für Informatik 10*. University Erlangen–Nürnberg, 2007.

[57] Hariharan G Lalgudi, Ali Bilgin, Michael W Marcellin, and Mariappan S Nadar. "Compression of fMRI and ultrasound images using 4D SPIHT". In: *IEEE Int'l Conf. on Image Processing (ICIP)*. Vol. 2. 2005, pp. 746–749.

[58] Hans Petter Langtangen and Anders Logg. *Solving PDEs in python: the FEniCS tutorial I*. Springer Nature, 2017.

[59] Michele Larobina and Loredana Murino. "Medical image file formats". In: *Journal of digital imaging* 27.2 (2014), pp. 200–206.

[60] Denis Le Bihan, Eric Breton, Denis Lallemand, Philippe Grenier, Emmanuel Cabanis, and Maurice Laval-Jeantet. "MR Imaging of Intravoxel Incoherent Motions: Application to Diffusion and Perfusion in Neurologic Disorders". In: *Radiology* 161.2 (1986), pp. 401–407.

[61] Alexander Leemans and Derek K Jones. "The B-matrix must be rotated when correcting for subject motion in DTI data". In: *Magnetic Resonance in Medicine* 61.6 (2009), pp. 1336–1349.

[62] Peng Li, Shuai-Jie Li, Zheng-An Yao, and Zu-Jin Zhang. "Two anisotropic fourth-order partial differential equations for image inpainting". In: *IET Image Processing* 7.3 (2013), pp. 260–269.

[63] Yun Li, Mårten Sjöström, Ulf Jennehag, and Roger 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 (3DTV-CON)*. IEEE. 2012, pp. 1–4.

[64] Dong Liu, Xiaoyan Sun, Feng Wu, Shipeng Li, and Ya-Qin Zhang. "Image compression with edge-based inpainting". In: *IEEE Transactions on Circuits and Systems for Video Technology* 17.10 (2007), pp. 1273–1287.

[65] Ying Liu and William A Pearlman. "Four-dimensional wavelet compression of 4-D medical images using scalable 4-D SBHP". In: *Proc. Data Compression Conference (DCC)*. IEEE. 2007, pp. 233–242.

[66] Anders Logg, Kent-Andre Mardal, and Garth Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*. Vol. 84. Springer Science & Business Media, 2012.

[67] Marius Lysaker, Arvid Lundervold, and Xue-Cheng Tai. "Noise removal using fourth-order partial differential equation with applications to medical magnetic resonance images in space and time". In: *IEEE Trans. on Image Processing* 12.12 (2003), pp. 1579–1590.

[68] Markus Mainberger, Andrés Bruhn, Joachim Weickert, and Søren Forchhammer. "Edge-based compression of cartoon-like images with homogeneous diffusion". In: *Pattern Recognition* 44.9 (2011), pp. 1859–1873.

[69] Markus Mainberger, Sebastian Hoffmann, Joachim Weickert, Ching Hoo Tang, Daniel Johannsen, Frank Neumann, and Benjamin Doerr. "Optimising spatial and tonal data for homogeneous diffusion inpainting". In: *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer. 2011, pp. 26–37.

[70] Markus Mainberger and Joachim Weickert. "Edge-Based Image Compression with Homogeneous Diffusion". In: *Int'l Conf. on Computer Analysis of Images and Patterns*. 2009, pp. 476–483.

[71] David Marwood, Pascal Massimino, Michele Covell, and Shumeet Baluja. "Representing images in 200 bytes: Compression via triangulation". In: *2018 25th IEEE International Conference on Image Processing (ICIP)*. IEEE. 2018, pp. 405–409.

[72] Simon Masnou and J-M Morel. "Level lines based disocclusion". In: *Proceedings 1998 International Conference on Image Processing. ICIP98 (Cat. No. 98CB36269)*. IEEE. 1998, pp. 259–263.

[73] Fabian Mentzer, Eirikur Agustsson, Michael Tschannen, Radu Timofte, and Luc Van Gool. "Practical full resolution learned lossless image compression". In: *Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2019, pp. 10629–10638.

[74] Sylvain Merlet. "Compressive sensing in diffusion MRI". PhD thesis. Université Nice Sophia Antipolis, 2013.

[75] Shaou-Gang Miaou, Fu-Sheng Ke, and Shu-Ching Chen. "A lossless compression method for medical image sequences using JPEG-LS and interframe coding". In: *IEEE Transactions on Information Technology in Biomedicine* 13.5 (2009), pp. 818–821.

[76] Omniah H Nagoor, J Whittle, J Deng, Benjamin Mora, and Mark W Jones. "Lossless Compression For Volumetric Medical Images Using Deep Neural Network With Local Sampling". In: *Proc. IEEE Int'l Conf. on Image Processing (ICIP)*. 2020, pp. 2815–2819.

[77] Milad Nasr, Reza Shokri, and Amir Houmansadr. "Comprehensive Privacy Analysis of Deep Learning: Passive and Active White-box Inference Attacks against Centralized and Federated Learning". In: *Proc. IEEE Symp. on Security and Privacy*. 2019, pp. 739–753.

[78] Binh P Nguyen, Chee-Kong Chui, Sim-Heng Ong, and Stephen Chang. "An efficient compression scheme for 4-D medical images using hierarchical vector quantization and motion compensation". In: *Computers in Biology and Medicine* 41.9 (2011), pp. 843–856.

[79] Stanley Osher and Leonid I Rudin. "Feature-oriented image enhancement using shock filters". In: *SIAM Journal on numerical analysis* 27.4 (1990), pp. 919–940.

[80] William B. Pennebaker and Joan L. Mitchell. *JPEG Still Image Data Compression Standard*. 1st. USA: Kluwer Academic Publishers, 1992. ISBN: 0442012721.

[81] William B. Pennebaker and Joan L. Mitchell. *JPEG: Still Image Data Compression Standard*. Springer, 1993.

[82] Pietro Perona and Jitendra Malik. "Scale-space and edge detection using anisotropic diffusion". In: *IEEE Trans. on Pattern Analysis and Machine Intelligence* 12.7 (1990), pp. 629–639.

[83] Pascal Peter. "Fast inpainting-based compression: Combining Shepard interpolation with joint inpainting and prediction". In: *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE. 2019, pp. 3557–3561.

[84] Pascal Peter. "Three-dimensional data compression with anisotropic diffusion". In: *German Conf. on Pattern Recognition*. Springer. 2013, pp. 231–236.

[85] Pascal Peter, Jan Contelly, and Joachim Weickert. "Compressing Audio Signals with Inpainting-based Sparsification". In: *Scale Space and Variational Methods*. Vol. 11603. LNCS. Springer, 2019, pp. 92–103.

[86] Pascal Peter, Sebastian Hoffmann, Frank Nedwed, Laurent Hoeltgen, and Joachim Weickert. "Evaluating the true potential of diffusion-based inpainting in a compression context". In: *Signal Processing: Image Communication* 46 (2016), pp. 40–53.

[87] Pascal Peter, Sebastian Hoffmann, Frank Nedwed, Laurent Hoeltgen, and Joachim Weickert. "From optimised inpainting with linear PDEs towards competitive image compression codecs". In: *Image and Video Technology*. Springer. 2015, pp. 63–74.

[88] Pascal Peter, Lilli Kaufhold, and Joachim Weickert. "Turning diffusion-based image colorization into efficient color compression". In: *IEEE Transactions on Image Processing* 26.2 (2016), pp. 860–869.

[89] Pascal Peter, Christian Schmaltz, Nicolas Mach, Markus Mainberger, and Joachim Weickert. "Beyond pure quality: Progressive modes, region of interest coding, and real time video decoding for PDE-based image compression". In: *Journal of Visual Communication and Image Representation* 31 (2015), pp. 253–265.

[90] Carlo Pierpaoli and Peter J. Basser. "Toward A Quantitative Assessment of Diffusion Anisotropy". In: *Magnetic Resonance in Medicine* 36 (1996), pp. 893–906.

[91] Sorin Pop, Romulus Terebes, Monica Borda, Sebastien Guillon, Naamen Keskes, Pierre Baylou, and Olivier Lavialle. "3D seismic data fusion and filtering using a PDE-based approach". In: *2007 IEEE International Conference on Image Processing*. Vol. 4. IEEE. 2007, pp. IV–117.

[92] David Salomon. "Data compression". In: *Handbook of massive data sets*. Springer, 2002, pp. 245–309.

[93] David Salomon and Giovanni Motta. *Handbook of data compression*. Springer, 2010.

[94] Victor Sanchez, Panos Nasiopoulos, and Rafeef Abugharbieh. "Efficient lossless compression of 4-D medical images based on the advanced video coding scheme". In: *IEEE Trans. on Information Technology in Biomedicine* 12.4 (2008), pp. 442–446.

[95] Otmar Scherzer. "Denoising with higher order derivatives of bounded variation and an application to parameter estimation". In: *Computing* 60.1 (1998), pp. 1–27.

[96] Christian Schmaltz, Pascal Peter, Markus Mainberger, Franziska Ebel, Joachim Weickert, and Andrés Bruhn. "Understanding, optimising, and extending data compression with anisotropic diffusion". In: *Int'l Journal of Computer Vision* 108.3 (2014), pp. 222–240.

[97] Christian Schmaltz, Joachim Weickert, and Andrés Bruhn. "Beating the quality of JPEG 2000 with anisotropic diffusion". In: *Joint Pattern Recognition Symposium*. Springer. 2009, pp. 452–461.

[98] Thomas Schultz, Joachim Weickert, and Hans-Peter Seidel. "A Higher-Order Structure Tensor". In: *Visualization and Processing of Tensor Fields – Advances and Perspectives*. Ed. by David H. Laidlaw and Joachim Weickert. Springer, 2009, pp. 263–280.

[99] Claude E Shannon. "A mathematical theory of communication". In: *The Bell system technical journal* 27.3 (1948), pp. 379–423.

[100] Edward O Stejskal and John E Tanner. "Spin diffusion measurements: spin echoes in the presence of a time-dependent field gradient". In: *The journal of chemical physics* 42.1 (1965), pp. 288–292.

[101] David Taubman and Michael Marcellin. *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Springer, 2002.

[102] S Thayammal and D Selvathi. "A Review On Segmentation Based Image Compression Techniques." In: *Journal of Engineering Science & Technology Review* 6.3 (2013).

[103] Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. "Lossy Image Compression with Compressive Autoencoders". In: *Int'l Conf. on Learning Representations (ICLR)*. 2017.

[104] Ch Thum. "Measurement of the entropy of an image with application to image focusing". In: *Optica Acta: International Journal of Optics* 31.2 (1984), pp. 203–211.

[105] Alexandra Tobisch, Rüdiger Stirnberg, Robbert L. Harms, Thomas Schultz, Alard Roebroeck, Monique M. Breteler, and Tony Stöcker. "Compressed Sensing Diffusion Spectrum Imaging for Accelerated Diffusion Microstructure MRI in Long-Term Population Imaging". In: *Frontiers in Neuroscience* 12 (2018), p. 650.

[106] George Toderici, Sean M. O'Malley, Sung Jin Hwang, Damien Vincent, David Minnen, Shumeet Baluja, Michele Covell, and Rahul Sukthankar. "Variable Rate Image Compression with Recurrent Neural Networks". In: *Int'l Conf. on Learning Representations (ICLR)*. Ed. by Yoshua Bengio and Yann LeCun. 2016.

[107] George Toderici, Damien Vincent, Nick Johnston, Sung Jin Hwang, David Minnen, Joel Shor, and Michele Covell. "Full resolution image compression with recurrent neural networks". In: *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 5306–5314.

[108] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. "SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python". In: *Nature Methods* 17 (2020), pp. 261–272.

[109] J. Weickert. "Theoretical Foundations of Anisotropic Diffusion in Image Processing". In: *Theoretical Foundations of Computer Vision*. Ed. by W. Kropatsch, R. Klette, F. Solina, and R. Albrecht. Vienna: Springer Vienna, 1996, pp. 221–236. ISBN: 978-3-7091-6586-7.

[110] Joachim Weickert. *Anisotropic diffusion in image processing*. Teubner Stuttgart, 1998.

[111] Joachim Weickert, Martin Welk, and Marco Wickert. "L 2-stable nonstandard finite differences for anisotropic diffusion". In: *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer. 2013, pp. 380–391.

[112] Ian H. Witten, Radford M. Neal, and John G. Cleary. "Arithmetic Coding for Data Compression". In: *Communications of the ACM* 30.6 (1987), pp. 520–540.

[113] David Yee, Sara Soltaninejad, Deborsi Hazarika, Gaylord Mbuyi, Rishi Barnwal, and Anup Basu. "Medical image compression based on region of interest using better portable graphics (BPG)". In: *IEEE Int'l Conf. on Systems, Man, and Cybernetics*. 2017, pp. 216–221.

[114] Yu-Li You and Mostafa Kaveh. "Fourth-order partial differential equations for noise removal". In: *Image Processing, IEEE Transactions on* 9.10 (2000), pp. 1723–1730.

[115] Shekoufeh Gorgi Zadeh, Stephan Didas, Maximilian WM Wintergerst, and Thomas Schultz. "Multi-scale Anisotropic Fourth-Order Diffusion Improves Ridge and Valley Localization". In: *Journal of Mathematical Imaging and Vision* 59.2 (2017), pp. 257–269. DOI: 10.1007/s10851-017-0729-1. URL: https://doi.org/10.1007/s10851-017-0729-1.

[116] Li Zeng, Christian P Jansen, Stephan Marsch, Michael Unser, and Patrick R Hunziker. "Four-dimensional wavelet compression of arbitrarily sized echocardiographic data". In: *IEEE Transactions on Medical Imaging* 21.9 (2002), pp. 1179–1187.

[117] Fang Zhang, Ying Chen, Zhitao Xiao, Lei Geng, Jun Wu, Tiejun Feng, Ping Liu, Yufei Tan, and Jinjiang Wang. "Partial differential equation inpainting method based on image characteristics". In: *Int'l Conf. on Image and Graphics*. Springer. 2015, pp. 11–19.

[118] Honglei Zhang, Francesco Cricri, Hamed R Tavakoli, Nannan Zou, Emre Aksu, and Miska M Hannuksela. "Lossless image compression using a multi-scale progressive statistical model". In: *Proceedings of the Asian Conference on Computer Vision*. 2020.

[119] Matthew J Zukoski, Terrance Boult, and Tunç Iyriboz. "A novel approach to medical image compression". In: *Int'l J. of Bioinformatics Research and Applications* 2.1 (2006), pp. 89–103.