Published in Scale Space and Variational Methods in Computer Vision
The final authenticated version is published by Springer and available online at
https://doi.org/10.1007/978-3-030-75549-2_36

Lossless PDE-based Compression of 3D Medical Images *

Ikram Jumakulyyev[0000−0002−7096−8717] and
Thomas Schultz[0000−0002−1200−7248]

University of Bonn, Germany
{ijumakulyyev,schultz}@cs.uni-bonn.de

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.

1 Introduction

The overall size of neuroimaging data that is acquired each year has been reported to grow exponentially [5], 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 [6,24,23], videos [11,22,1], and audio [21]. 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.

* 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.

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 [20]. Even more importantly, all above-mentioned codecs are for lossy compression, and their benefit relative to established transform-based codecs like JPEG [18] and JPEG2000 [25] tends to be most pronounced at high compression rates [24]. 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 [9,10,16], 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 3, 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 4, 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.

2 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 [12]. 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 [28,20], the lossless JPEG family for lossless compression [10,9].

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 [4] to them. GZIP is based on the Deflate algorithm [3], 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 [9]. In Section 4.1, we will compare our own codec to JPEG-LS, lossless JPEG and JPEG2000, as well as GZIP.

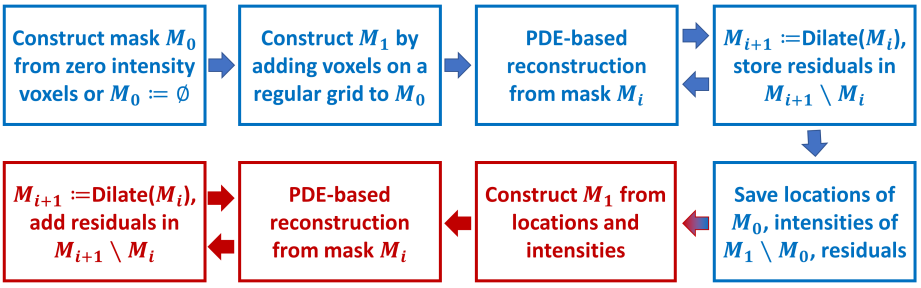

Fig. 1. An overview of the individual steps taken to encode (blue) or decode (red) a 3D image. At the core of our codec is an iteration that alternates between PDE-based reconstruction from an inpainting mask, and a dilation of that mask.

Prior to our work, the only PDE-based image compression codec for 3D medical images was C-EED [20]. It is based on edge-enhancing diffusion (EED) [26] and a cuboid subdivision scheme that extends the rectangular subdivision in the previously proposed R-EED codec [24]. 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 [15]. 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 [17]. We consider this to be a separate line of research which is outside of our scope.

3 Our Proposed Lossless Codec

Figure 1 shows a high-level overview of our PDE-based lossless codec. Section 3.1 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 3.2. Finally, the initial mask and residuals are stored in compressed form (Section 3.4). The decoder (red) mirrors the encoder in that it again alternates between reconstruction and mask dilation.

3.1 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 [14,13], and optimal inpainting masks have been approximated by sophisticated mask selection methods [6,7,24,23].

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.

3.2 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 4.3 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.

3.3 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 [24], and a recently introduced fourth-order generalization of EED [8].

Second-order diffusion can be stated as

$$\partial_t u = \operatorname{div}(\mathbf{D} \cdot \nabla u), \tag{1}$$

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 $t \to \infty$ [7].

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. (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 [26,8] 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,

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

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 [23].

A second parameter in Eq. (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 [24,23].

Empirically optimizing λ causes a noticeable computational expense and, as will be reported in more detail in Section 4.4, 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 [19], 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.

3.4 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).

Table 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%

4 Results

For our experiments, we chose four 3D medical images which are illustrated in Figure 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 [2] 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 [28,27].
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.

4.1 Comparison to Other Codecs

Table 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 2. Results consider different variants of our codec, using linear homogeneous (LH), second-order edge enhancing (EED),

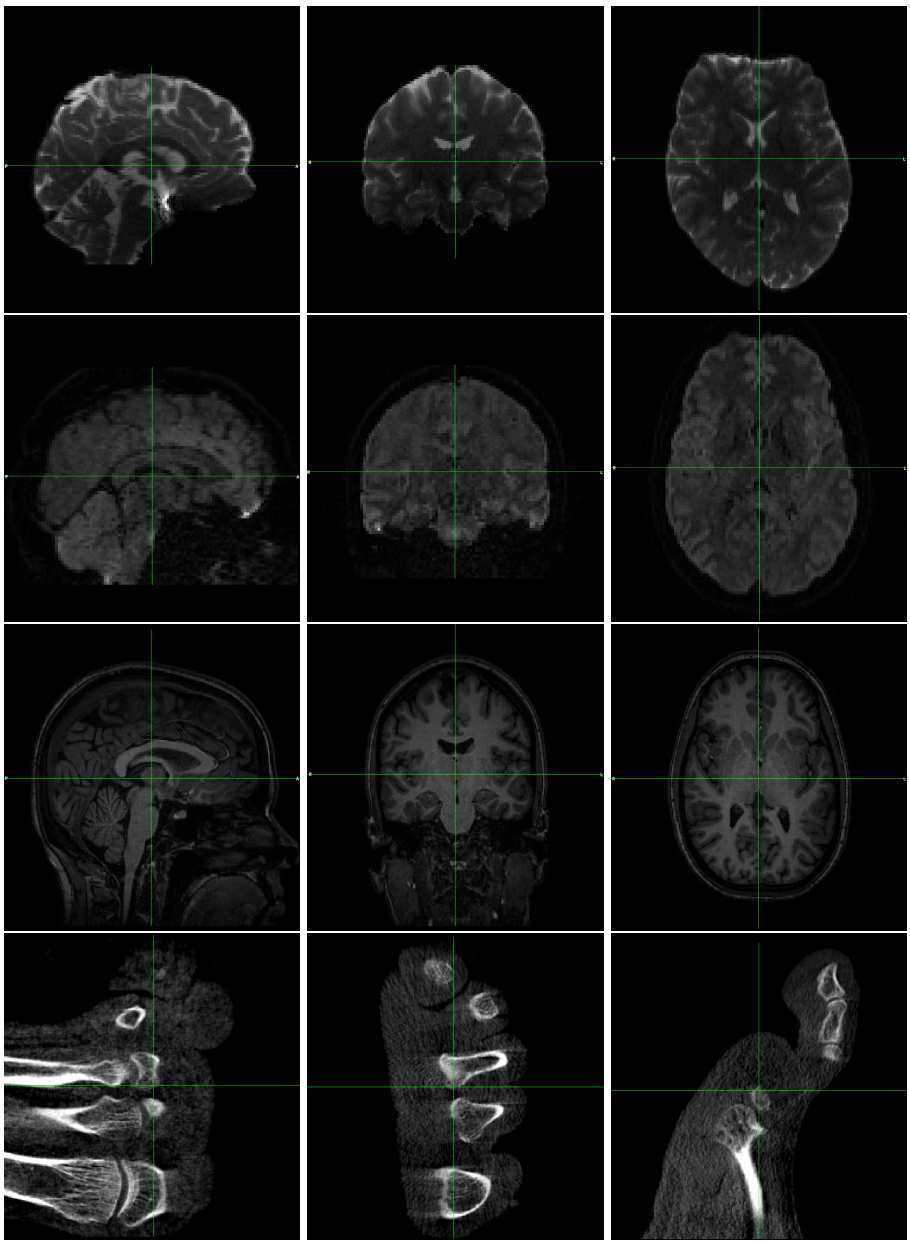

Fig. 2. The four 3D medical images used in our experiments. Top to bottom: Three brain MR images with *B0*, *B700*, and *T1* weighting, and a foot *CT* image. Right to left: The middle slices on the sagittal, coronal, and axial planes.

Table 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

or fourth-order edge enhancing diffusion (FOEED). All cases use iterative reconstruction from a regular grid (R-I) in Mode 1 (see Section 3.2). The effect of using fewer or no iterations will be studied separately in Section 4.3.

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.

4.2 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.

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 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 3 shows that this baseline already improves considerably over GZIP.

Table 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

Table 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 3.2 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%

4.3 Effect of Iterative Construction of Residuals

Table 4 shows how the iterative alternation between reconstruction and residual coding that is described in Section 3.2 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 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.

4.4 Effect of Contrast Parameter

Even though it can be seen from Table 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 explores the effect of varying the ad-hoc threshold value of 90% that was used in Section 3.3 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 used second-order EED with iteration mode 0, because fine-tuning the contrast parameter would be even more costly in mode 1.

Table 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.

	30% Threshold		60% Threshold		90% Threshold	
Image	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%

5 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 [2]. This will require operating on the space of positions and orientation.

References

1. Andris, S., Peter, P., Weickert, J.: A proof-of-concept framework for PDE-based video compression. In: Picture Coding Symposium (PCS). pp. 1–5. IEEE (2016)
2. Basser, P.J., Mattiello, J., Le Bihan, D.: Estimation of the effective self-diffusion tensor from the NMR spin echo. J. of Magnetic Resonance **B**(103), 247–254 (1994)
3. Deutsch, P.: RFC1951: DEFLATE compressed data format specification version 1.3 (1996)
4. Deutsch, P.: RFC1952: GZIP file format specification version 4.3 (1996)
5. Dinov, I.D.: Volume and value of big healthcare data. Journal of medical statistics and informatics **4**, 3 (2016)
6. Galić, I., Weickert, J., Welk, M., Bruhn, A., Belyaev, A., Seidel, H.P.: Towards PDE-based image compression. In: Int'l Workshop on Variational, Geometric, and Level Set Methods in Computer Vision. pp. 37–48. Springer (2005)
7. Galić, I., Weickert, J., Welk, M., Bruhn, A., Belyaev, A., Seidel, H.P.: Image compression with anisotropic diffusion. Journal of Mathematical Imaging and Vision **31**(2-3), 255–269 (2008)
8. Jumakulyyev, I., Schultz, T.: Fourth-order anisotropic diffusion for inpainting and image compression. In: Anisotropy Across Fields and Scales, pp. 99–123. Mathematics and Visualization, Springer (2021)
9. Kil, S.K., Lee, J.S., Shen, D., Ryu, J., Lee, E., Min, H., Hong, S.: Lossless medical image compression using redundancy analysis. International Journal of Computer Science and Network Security **6**(1), 50–56 (2006)

10. Kim, Y.S., Pearlman, W.A.: Lossless volumetric medical image compression. In: Applications of Digital Image Processing XXII. vol. 3808, pp. 305–312. International Society for Optics and Photonics (1999)
11. Köstler, H., Stürmer, M., Freundl, C., Rüde, U.: PDE based video compression in real time. In: Tech. Rep. 07-11, Lehrstuhl für Informatik 10. University Erlangen–Nürnberg (2007)
12. Larobina, M., Murino, L.: Medical image file formats. Journal of digital imaging **27**(2), 200–206 (2014)
13. Liu, D., Sun, X., Wu, F., Li, S., Zhang, Y.Q.: Image compression with edge-based inpainting. IEEE Transactions on Circuits and Systems for Video Technology **17**(10), 1273–1287 (2007)
14. Mainberger, M., Weickert, J.: Edge-based image compression with homogeneous diffusion. In: Int'l Conf. on Computer Analysis of Images and Patterns. pp. 476–483 (2009)
15. Mentzer, F., Agustsson, E., Tschannen, M., Timofte, R., Gool, L.V.: Practical full resolution learned lossless image compression. In: Proc. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). pp. 10629–10638 (2019)
16. Miaou, S.G., Ke, F.S., Chen, S.C.: A lossless compression method for medical image sequences using JPEG-LS and interframe coding. IEEE Transactions on Information Technology in Biomedicine **13**(5), 818–821 (2009)
17. Nasr, M., Shokri, R., Houmansadr, A.: 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. pp. 739–753 (2019)
18. Pennebaker, W.B., Mitchell, J.L.: JPEG: Still Image Data Compression Standard. Springer (1993)
19. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Trans. on Pattern Analysis and Machine Intelligence **12**(7), 629–639 (1990)
20. Peter, P.: Three-dimensional data compression with anisotropic diffusion. In: German Conf. on Pattern Recognition. pp. 231–236. Springer (2013)
21. Peter, P., Contelly, J., Weickert, J.: Compressing audio signals with inpainting-based sparsification. In: Scale Space and Variational Methods. LNCS, vol. 11603, pp. 92–103. Springer (2019)
22. Peter, P., Schmaltz, C., Mach, N., Mainberger, M., Weickert, J.: Beyond pure quality: Progressive modes, region of interest coding, and real time video decoding for PDE-based image compression. Journal of Visual Communication and Image Representation **31**, 253–265 (2015)
23. Schmaltz, C., Peter, P., Mainberger, M., Ebel, F., Weickert, J., Bruhn, A.: Understanding, optimising, and extending data compression with anisotropic diffusion. Int'l Journal of Computer Vision **108**(3), 222–240 (2014)
24. Schmaltz, C., Weickert, J., Bruhn, A.: Beating the quality of JPEG 2000 with anisotropic diffusion. In: Joint Pattern Recognition Symposium. pp. 452–461. Springer (2009)
25. Taubman, D., Marcellin, M.: JPEG2000: Image Compression Fundamentals, Standards and Practice. Springer (2002)
26. Weickert, J.: Anisotropic diffusion in image processing. Teubner Stuttgart (1998)
27. Yee, D., Soltaninejad, S., Hazarika, D., Mbuyi, G., Barnwal, R., Basu, A.: Medical image compression based on region of interest using better portable graphics (BPG). In: IEEE Int'l Conf. on Systems, Man, and Cybernetics. pp. 216–221 (2017)
28. Zukoski, M.J., Boult, T., Iyriboz, T.: A novel approach to medical image compression. Int'l J. of Bioinformatics Research and Applications **2**(1), 89–103 (2006)