In J. Lellmann, M. Burger, J. Modersitzki (Eds.): Scale Space and Variational Methods. Lecture Notes in Computer Science, Vol. 11603, pp. 92–103, Springer, Cham, 2019. The final publication is available at link.springer.com.

Compressing Audio Signals with Inpainting-based Sparsification

Pascal Peter, Jan Contelly, and Joachim Weickert

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

Abstract. Inpainting techniques are becoming increasingly important for lossy image compression. In this paper, we investigate if successful ideas from inpainting-based codecs for images can be transferred to lossy audio compression. To this end, we propose a framework that creates a sparse representation of the audio signal directly in the sample-domain. We select samples with a greedy sparsification approach and store this optimised data with entropy coding. Decoding restores the missing samples with well-known 1-D interpolation techniques. Our evaluation on music pieces in a stereo format suggests that the lossy compression of our proof-of-concept framework is quantitatively competitive to transform-based audio codecs such as mp3, AAC, and Vorbis.

1 Introduction

Inpainting [23] originates from image restoration, where missing or corrupted image parts need to be filled in. This concept can also be applied for compression: Inpainting-based codecs [13] represent an image directly by a sparse set of known pixels, a so-called inpainting mask. This mask is selected and stored during encoding, and decoding involves a reconstruction of the missing image parts with a suitable interpolation algorithm. Such codecs [26, 27] can reach competitive quality to JPEG [25] and JPEG2000 [31], which create sparsity indirectly via cosine or wavelet transforms.

In lossy audio compression, all state-of-the-art codecs use a time-frequency representation of the signal and are thereby also transform-based. This applies to mp3 (MPEG layer-III) [17], advanced audio coding (AAC) [18], and the open source alternative Vorbis [34]. They resemble the classic image codecs, whereas inpainting-based compression has so far not been explored for audio data. Therefore, we propose to select and store samples directly for sparse audio representations that act as known data for inpainting.

The transition from images to audio creates some unique challenges, since visual and audio data differ in many regards. As has been shown for 3-D medical images [27], the effectiveness of inpainting-based codecs increases with the dimensionality of the input data. Audio signals only have a single time dimension, but feature a high dynamic range compared to the 8-bit standard in images. Moreover, more high-frequent changes can be expected in audio files. So far it is unknown how these differences affect the performance of interpolation and data selection strategies. In the following, we want to investigate the potential of inpainting-based audio compression.

Our Contribution. We propose a framework for lossy audio compression that is designed to transfer successful ideas from inpainting-based image compression to the audio setting. Based on this framework, we implement two proof-of-concept codecs that rely on different 1-D inpainting techniques: linear and cubic Hermite spline interpolation. Moreover, we integrate two core concepts from inpainting-based compression: sparsification [22] for the selection of known data locations, and tonal optimisation [22, 27] of the corresponding values. Our input data, music pieces in a stereo format, contain significantly more data than standard test images. Therefore, we need to adapt the optimisation techniques to the audio setting. Localised inpainting allows us to decrease computation time significantly without affecting quality. Moreover, we propose a greedy sparsification approach with global error computation instead of the stochastic, local methods common in image compression. A combination of quantisation, run-length encoding (RLE) and context-mixing for storage of the known audio data complements these optimisation strategies. We compare our new codecs to mp3, AAC, and Vorbis w.r.t. the signal-to-noise ratio.

Related Work. The reconstructing of missing image areas was first investigated by Masnou and Morel [23] who referred to this as a disocclusion problem. Later, Bertalmío et al. [4] coined the term inpainting for this application of interpolation in image restoration. Many successful contemporary inpainting operators rely on partial differential equations (PDEs), for instance homogeneous diffusion [16] or edge-enhancing anisotropic diffusion (EED) [33] inpainting. An overview of methods can be found in the monograph of Schönlieb [28]. These methods achieve a filling-in effect based on physical propagation models. Another popular approach to inpainting are exemplar-based strategies that restore missing values by nonlocal copy-paste according to neighbourhood similarities [12]. For audio, the term inpainting is rarely used. However, interpolation is an important tool for signal restoration (e.g. [20]) or synthesis (e.g. [24]). Adler et al. [2] were the first to apply the core ideas of inpainting to audio signals: They presented a framework for filling in missing audio data from a sparse representation in the time domain. Their dictionary approach relies on discrete cosine and Gabor bases. There is also a vast number of publications that deal with specific scenarios such as removal of artefacts from damaged records [20] or noise in voice recognition applications [14] that rely on interpolation and signal reconstruction in a broader sense. Since a complete review is beyond the scope of this paper, we refer to the overview by Adler et al. [2].

It should be noted that while interpolation is not widely used for audio compression, linear prediction has been applied successfully, for instance by Schuller et al. [29]. However, the core technique behind common codecs are transforms. For instance, MPEG layer-III [17] (mp3) uses a modified discrete cosine transform (MDCT) on a segmented audio signal. The sophisticated non-uniform quantisation strategies and subsequent entropy coding are augmented with psychoacoustic analysis of the signal's Fourier transform. Advanced audio coding (AAC) [18], the successor of mp3, and the open source codec Vorbis [34] rely on the same basic principles. They also combine the MDCT with psychoacoustic modelling, but can achieve a better quality due to an increased flexibility of the encoder. A more detailed discussion of these codecs and a broader overview of the field can be found in the monograph by Spanias et al. [30].

A major competitor to transform-based audio compression arose from the so-called sinusoidal model [20, 24]. It represents an audio signal as the weighted sum of wave functions that have time-adaptive parameters such as amplitude and frequency. Sinusoidal synthesis approaches have also been applied for compression. These use interpolation, but only in the domain of synthesis parameters. Such parametric audio compression [11] also forms the foundation of the MPEG4 HILN (harmonic and individual lines plus noise) standard [19]. HILN belongs to the class of object-based audio codecs: It is able to model audio files as a composition of semantic parts (e.g. chords). Vincent and Plumbley [32] transferred these ideas to a Bayesian framework for decomposing signals into objects.

Since we use ideas from image compression to build our framework, some inpainting-based codecs are closely related to our audio approach. The basic structure of the codec is inspired by the so-called exact mask approach by Peter et al. [26]. It is one of the few codecs that allows to choose and store known data without any positional restrictions. However, we do not use optimal control to find points as in [26]. Instead, we rely on sparsification techniques that resemble the probabilistic approach of Mainberger et al. [22]. Our greedy global sparsification differs from probabilistic sparsification in a few key points: In accordance to the findings of Adam et al. [1], we use global error computation instead of localised errors. Moreover, our approach does not rely on randomisation for spatial optimisation anymore, but is a deterministic, greedy process that always yields the same known data positions. Our choice of PAQ [21] for the storage of the sample data is motivated by several works that have evaluated different entropy coding techniques for inpainting-based compression [26, 27].

Organisation of the Paper. We introduce our framework for inpainting-based audio compression in Section 2. Details on inpainting and data optimisation for two codecs follow in Section 3. These codecs are evaluated in Section 4, and we conclude our paper with a summary and outlook in Section 5.

2 A Framework for Inpainting-based Audio Compression

Our proof-of-concept framework follows the common structure of current inpainting-based compression methods [26, 27]. An input audio file $\boldsymbol{f}\colon \Omega \subset \mathbb{N} \to \mathbb{Z}^c$

```
Input: Original audio signal $\boldsymbol{f} \in \mathbb{Z}^n$, desired final mask density $d$.
Initialisation: Mask $\boldsymbol{m} = (1,1,...,1)^\top \in \{0,1\}^n$ is full. Heap structure $h$ is empty.
    Update set $A = \{1,...,n\}$ contains all samples. Reconstruction error cache $\boldsymbol{e} = \boldsymbol{0}$.
Compute:
    do
        for all $k$ in $A$ do
            Remove the sample $f_k$ temporarily from the mask (i.e. $m_k = 0$).
            Inpaint reconstruction $r(\boldsymbol{m}, \boldsymbol{f})$ with error $e_k$.
            Set $m_k \leftarrow 1$, add sample $f_k$ to heap with reconstruction error $h_{e,k} \leftarrow e_k$.
        end for
        do
            Get the sample $f_j$ from top of heap $h$.
            if $m_j = 1$ and $h_{e,j} = e_j$
            then remove $f_j$ permanently from mask ($m_j \leftarrow 0$).
        while no sample has been permanently removed.
        Define $A$ as set of samples whose error value is affected by removal of $f_j$.
    while pixel density of $\boldsymbol{m}$ larger than $d$.
Output: Mask $\boldsymbol{m}$ of density $d$.
```

Algorithm 1: Greedy Global Mask Sparsification.

maps time coordinates $\Omega = \{1,...,n\}$ to samples of the waveforms from $c \geq 1$ audio channels. Encoding aims to select and store a subset $K \subset \Omega$ of known data. During decoding, inpainting uses these data for a lossy restoration of the missing samples on $\Omega \setminus K$.

In the following, we describe the individual steps of the encoding pipeline: This includes sample selection and optimisation, as well as efficient storage with prediction and entropy coding. Our optimisation are very flexible w.r.t. the inpainting method: We only assume that a deterministic inpainting algorithm computes a reconstruction $r(K, \boldsymbol{g})\colon \Omega \to \mathbb{Z}$ from samples $\boldsymbol{g}\colon K \to \mathbb{Z}$ on the set K. In Section 3 we discuss the actual inpainting methods for our experiments in Section 4. Our codec is designed to be easily extendable with other inpainting techniques, such as the dictionary-based approach of Adler et al. [2].

Step 1: Sample Quantisation. First, we apply a coarse quantisation that reduces the number of sample values to $q \geq 2$. In order to adapt to the coding pipeline from image processing, this involves a global shift to a non-negative sample range $\{0, \ldots, p-1\}$ which is reversed again during decoding. A uniform quantisation partitions this sample range into q subintervals of length p/q, mapping to quantised values $\{0, \ldots, q-1\}$. For inpainting, we assign the quantisation index k to the corresponding quantised value ℓ from the original range:

$$\ell = \left\lfloor \frac{kp}{q} + \frac{1}{2} \right\rfloor . \tag{1}$$

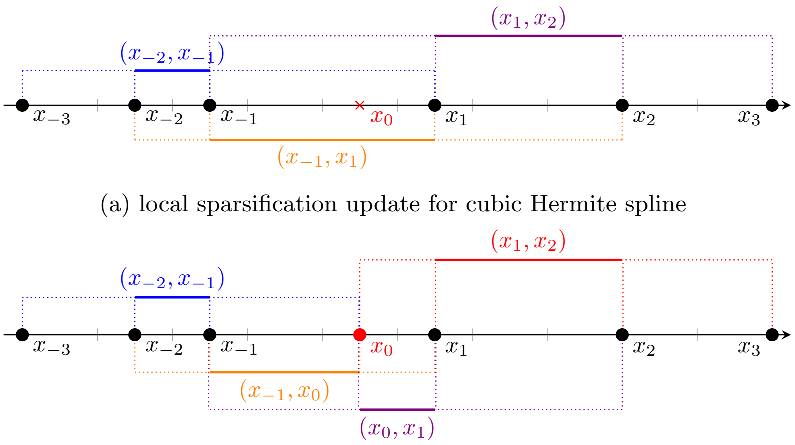
All following optimisation steps use quantised values for inpainting and coding.

Step 2: Greedy Global Sparsification. A popular method for selecting the spatial location of known pixels in inpainting-based compression is probabilistic sparsification [22]: It starts with a full pixel mask, removes a set of randomly selected candidate points. and performs inpainting. A subset of candidates with the lowest local error are then permanently removed, since they are considered easy to reconstruct. We iterate these steps until the desired number of mask points, the target density, is reached. This method is easy to implement and supports all inpainting techniques. However, a recent analysis [1] revealed that the local error computation in this approach yields a suboptimal point selection. Therefore, we use a different sparsification strategy that relies on global error computation as proposed by Adam et al [1]. Moreover, we remove the random component for candidate selection and obtain a *greedy global sparsification* that is described in Algorithm 1.

For each audio sample in the mask we compute the increase in the reconstruction error that would result from its removal. With this global reconstruction error, we sort the samples in a heap. Then we iteratively remove in every step the sample on top of the heap (i.e. with the lowest effect on the error) permanently from the mask. Afterwards, all mask samples that are affected by this removal are updated and reinserted into the heap. Which mask samples need to be updated depends on the inpainting approach (see Section 3). Note that, in order to avoid a costly purging of the heap in each iteration, the unmodified heap elements remain. If the sample at the top has been already removed or its error is not up-to-date, the algorithm moves on to the next one. For image compression, the global impact of individual changes in the mask cannot be considered due to runtime issues. In Section 3 we explain how we can reduce the computational load with an update strategy for the audio setting.

Step 3: Sample Optimisation. It is well-known in inpainting-based image compression that optimising not only the location of known data, but also the function value in the stored pixels can yield large improvements [7, 15, 22, 27]. Since we aim for a flexible framework, we use the technique from [27], as it does not require a specific inpainting technique. It performs a random walk over all mask samples: If a change to the next higher or lower quantisation level improves the reconstruction, it is kept, otherwise it is reverted. As for sparsification, we address runtime questions in Section 3.

Step 4: Location Encoding. Current state-of-the-art codecs [26] employ block coding in 2-D to store *exact masks* with unrestricted placement of known points. A natural substitute for this in 1-D is run-length encoding (RLE) [5]. We represent the mask as a sequences of ones (known samples) and zeroes (unknown samples). In sparse masks, we expect isolated ones with long runs of zeroes in-between. Therefore, we only encode runs of zeroes together with a terminating one. This allows us to store the mask as a sequence of 8bit symbols. Runs up to length 254 require only one symbol while longer runs are split accordingly (e.g. 300 is represented by $255, 45$).



(a) local sparsification update for cubic Hermite spline

(b) local sample update for cubic Hermite spline

Fig. 1: **Local Update Intervals** for the cubic Hermite spline. The coloured solid lines mark the update intervals and the associated dotted lines show which known samples are involved in the corresponding reconstruction.

Step 5: Prediction and Entropy Encoding. Due to recurring patterns in audio files (in particular for music recordings), prediction can be used to achieve higher compression ratios. To this end, many publications on inpainting-based compression (e.g. [26, 27]) apply the context-mixing algorithm PAQ [21]. It predicts the next bit in a stream containing different data types according to numerous predefined and learned contexts. The weighting of these contexts adapts to the local file content with a gradient descent on the coding cost. We use PAQ for an additional joint encoding of the output data from Steps 3–4.

3 Localised Sample Optimisation with 1-D Inpainting

So far, we have not specified concrete inpainting operators for our general framework. In the following, we transfer popular inpainting techniques from image compression to the audio setting. For these inpainting operators, we develop new techniques for the acceleration of the corresponding data optimisation.

Inpainting Techniques. For our first proof-of-concept implementation of the framework, we explore the potential of successful inpainting approaches from image compression. So far, three operators have shown convincing performance [7, 13, 27]: homogeneous diffusion [16], biharmonic [10], and edge-enhancing anisotropic diffusion (EED) inpainting [33]. EED has been particularly successful, since it allows to reconstruct image edges due to a direction dependent propagation. However, due to the 1-D nature of audio data, EED is not an option.

Homogeneous diffusion inpainting keeps all of the known data points on $K \subset \Omega$ unchanged, while the unknown data on $\Omega \setminus K$ must fulfil the Laplace equation $\Delta u = 0$ with $\Delta u = \partial_{xx} u + \partial_{yy} u$. In 1-D, this implies a vanishing second order derivative, which leads to a straightforward linear interpolation between the known data points. This comes down to a minimisation of the energy

$$E_L(u) = \int_{\Omega \setminus K} \left(u'(x)\right)^2 dx. \tag{2}$$

In the following sections we benefit from the *compact support* of the corresponding interpolation function: For the reconstruction $u(x)$ at a location $x \in \Omega \setminus K$ in the inpainting domain, we need a small amount of neighbouring known values (x_k, u_k). In the following, the indices $k = \pm 1, \pm 2, ...$ denote the respective closest known samples in positive/negative x-direction. For linear interpolation, we only require the two known samples (x_{-1}, u_{-1}) and (x_1, u_1) to obtain the reconstruction

$$u(x) = \frac{x - x_{-1}}{x_1 - x_{-1}} u_1 + \left(1 - \frac{x - x_{-1}}{x_1 - x_{-1}}\right) u_{-1}. \tag{3}$$

Biharmonic inpainting is a higher-order approach that imposes the constraint $-\Delta^2 u = 0$ to the inpainted data, thereby providing a smoother reconstruction compared to the homogeneous case. Cubic splines are a natural 1-D counterpart to this approach. They have been originally motivated by a physical elasticity model for draftman's splines [9] and minimise the energy

$$E_{CS}(u) = \int_{\Omega \setminus K} \left(u''(x)\right)^2 dx. \tag{4}$$

However, since we aim to reach a similar locality as for the linear interpolation, we consider a specific variant of cubic splines, the cubic Hermite spline interpolation [6] (Catmull-Rom spline). It yields an interpolant with C^1-smoothness using a finite support. Since it does not require equidistant sampling, it is therefore compatible with sparsification. With $\alpha := \frac{x - x_{-1}}{x_1 - x_{-1}}$, the interpolant of cubic Hermite spline interpolation is

$$\begin{aligned} u(x) = (2\alpha^3 - 3\alpha^2 + 1) u_{-1} + (\alpha^3 - 2\alpha^2 + \alpha) \frac{u_1 - u_{-2}}{x_1 - x_{-2}} \\ + (-2\alpha^3 + 3\alpha^2) u_1 + (\alpha^3 - \alpha^2) \frac{u_2 - u_{-1}}{x_2 - x_{-1}}. \end{aligned} \tag{5}$$

For the interpolation techniques above, we round to the next 16bit integer. Note that this rounding is explicitly not restricted to the quantisation levels according to Step 3 of our compression pipeline from Section 2. From a very small set of quantised values, the inpainting can potentially recover a much broader sample range, if the known data is chosen appropriately. In the following, we discuss how the locality of our inpainting methods can accelerate the data optimisation.

Local Interpolation Updates. Both the greedy sparsification from Algorithm 1 and the sample optimisation from Step 3 require a global reconstruction error. Recomputing the whole reconstruction after a change of a single mask point is a significant drawback of these approaches in 2-D. However, in our 1-D audio signal setting using interpolations methods with finite support, the influence of each mask sample is limited. In the following, we always assume a sample x_0 is changed by an optimisation algorithm, and $x_{-1}, x_{-2}, ...$ denote it left mask neighbours while $x_1, x_2, ...$ are its right mask neighbours.

In a sparsification step, we remove the known sample value y_0 with time coordinate x_0 from the mask. For linear interpolation, this removal affects exactly the reconstruction of the samples $x \in (x_{-1}, x_1)$, which are now reconstructed with the known data x_{-1} and x_1. For the cubic Hermite spline, the situation is similar, but due to the larger support, the interval (x_{-2}, x_2) is affected now. Moreover, it has to be split into three subintervals that are inpainted with different combinations of the known data $x_{-3}, ..., x_3$ (see Fig. 1(a)).

The update strategy for sample optimisation follows the same principle, but more subintervals need to be considered, since we now change the value y_0 at location x_0 instead of removing the sample completely. Thus, for linear interpolation, the optimisation algorithm needs to recompute the intervals (x_{-1}, x_0) and (x_0, x_1) with the new known sample (x_0, y_0). Since the cubic Hermite spline relies on four samples, the sample change affects four intervals: (x_{-2}, x_{-1}), (x_{-1}, x_0), (x_0, x_1), and (x_1, x_2). Fig. 1(b) illustrates the associated samples. Note that we also need the samples x_{-3} and x_3 to compute these reconstructions.

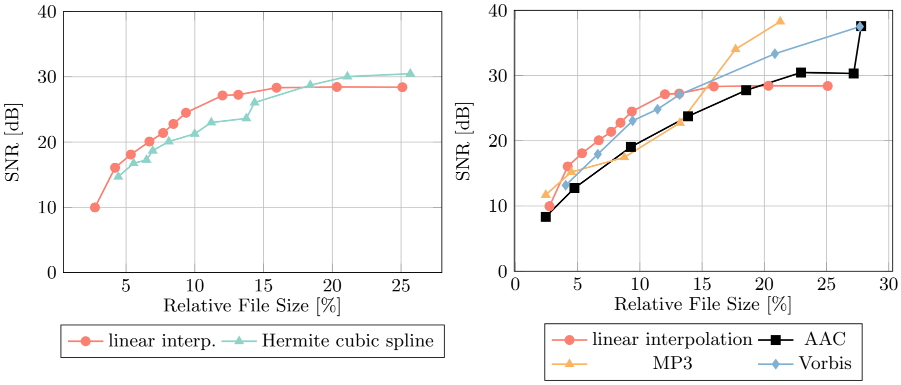
4 Experiments

Experimental Setup. We present detailed results for a royality-free sound file of the song *Exploring the Forest* [8] (linear 16bit pulse coded modulation (PCM) with 44.100 kHz sampling rate and two channels). Results for additional music pieces from a variety of genres as well as playable soundfiles are available online as supplementary material[1]. As a quantitative measure, we use the signal to noise ratio (SNR) that is defined by

$$\mathrm{SNR}(f,g) = 10 \log_{10} \left(\frac{\sum_{i=1}^{n} f_i^2}{\sum_{i=1}^{n} (f_i - g_i)^2} \right) . \tag{6}$$

Comparison of Inpainting Methods. In a first experiment, we compare the performance of the two inpainting methods from Section 3. A quantitative SNR comparison in Fig. 2(a) reveals that linear interpolation yields almost the same SNR and outperforms the Hermite cubic spline for small compressed file sizes. The increased smoothness of the cubic spline comes at the cost of over- and undershoots close to the known samples. These can only be compensated adequately if most of the samples are known. Moreover, for linear interpolation, the

[1] https://www.mia.uni-saarland.de/Publications/peter-ssvm19-supplement.zip



(a) interpolation comparison

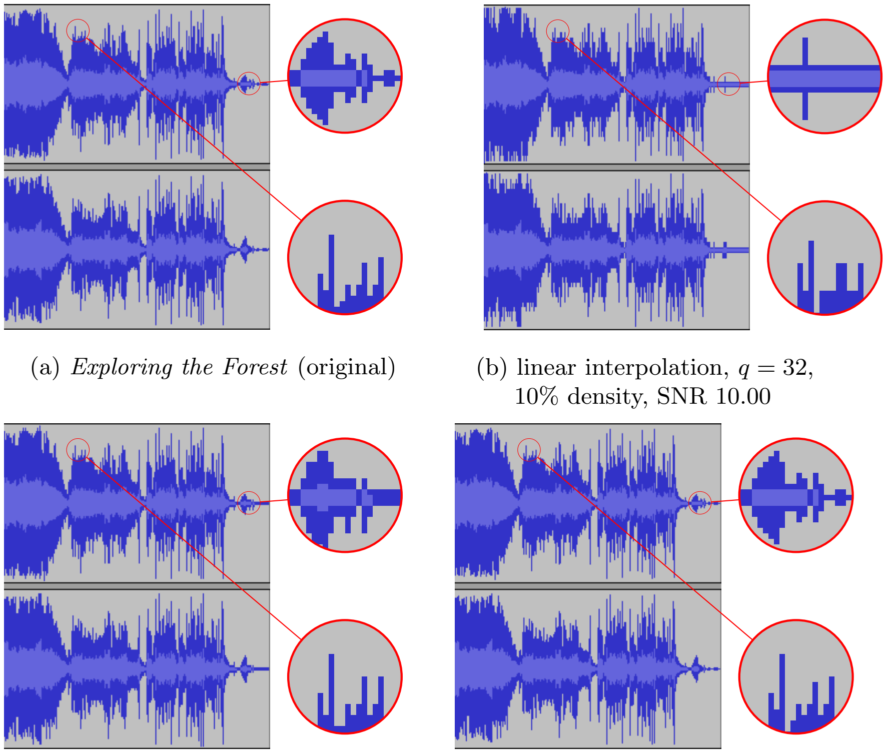
(b) codec comparison

Fig. 2: **SNR Comparisons.** Fig. 2(a) reveals that linear interpolation mostly outperforms the cubic Hermite spline. In Fig. 2(b), our inpainting-based codec with linear interpolation compares favourably with established methods like AAC, mp3, and Vorbis for low to medium compression ratios.

best density and quantisation parameters increase proportionally to the file size. Hermite cubic spline interpolation is more sensitive in this regard. Therefore, we choose linear interpolation for our comparison to established codecs.

Comparison to Established Codecs. Our second series of experiments evaluates the compression performance of our best inpainting codec (with linear interpolation) to the established codecs mp3, Vorbis, and AAC. Our evaluation in Fig. 2(b) yields a surprising result: For medium to high compression ratios, our codec surpasses all three transform-based approaches w.r.t. the SNR. This demonstrates that concepts from inpainting-based compression can be viable in an audio setting, even with simple inpainting methods. For small compression ratios, our codec falls slightly below the SNR of AAC. Inpainting-based methods show similar behaviour for near-lossless coding of images (see e.g. [27]). This is natural, since the impact of inpainting diminishes for dense masks.

Fig. 3 provides a visualisation of the inpainting results for the lowest and highest compression ratios from Fig. 2(a) with linear interpolation. On first glance, even with a low density of 10 % and a very coarse quantisation ($q = 32$) the reconstructed waveform in Fig. 3(b) looks similar to the original in Fig. 3(a). However, some of the peaks are flattened (especially apparent at the end of the signal). On a temporal average, this is still close to the original signal in terms of SNR, but there are some audible artefacts like background noise. A higher density and finer quantisation leads to increasingly improved results in Fig. 3(c) and Fig. 3(d). Simple linear interpolation can reproduce the original waveform from carefully optimised known samples with surprising accuracy.



(a) *Exploring the Forest* (original)

(b) linear interpolation, $q = 32$, 10% density, SNR 10.00

(c) linear interpolation, $q = 64$, 50% density, SNR 22.77

(d) linear interpolation, $q = 1024$, 90% density, SNR 28.40

Fig. 3: **Visual Comparison of Waveforms.** Waveforms of the original file *Exploring the Forest* and the reconstructions corresponding to low, medium, and high compression ratios from Fig. 2.

5 Conclusions and Outlook

Our modular framework for audio compression demonstrates the potential of inpainting with data optimisation for the sparse representation of sample data. Even with fairly simple ingredients, our proof-of-concept codecs are able to compete with established audio codecs w.r.t. quantitative analysis. In particular, this discovery is relevant for recent approaches in inpainting-based video compression [3]: Inpainting-based audio codecs would augment them in a natural way by offering a consistent way of encoding the corresponding audio tracks.

In our future work, we plan to investigate more sophisticated inpainting techniques that have been designed specifically for the audio setting (e.g. [2]), and address practical issues such as random access. Moreover, we will incorporate the psychoacoustic modelling used by transform-based codecs: Prefiltering the signal to eliminate frequencies that are unimportant for human perception might further improve the performance of inpainting-based audio compression.

Acknowledgements. We thank Jan Østergaard (Aalborg University) for the valuable discussions that allowed us to improve our work. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement no. 741215, ERC Advanced Grant INCOVID).

References

1. Adam, R.D., Peter, P., Weickert, J.: Denoising by inpainting. In: Lauze, F., Dong, Y., Dahl, A.B. (eds.) Scale Space and Variational Methods in Computer Vision, Lecture Notes in Computer Science, vol. 10302, pp. 121–132. Springer, Cham (2017)
2. Adler, A., Emiya, V., Jafari, M.G., Elad, M., Gribonval, R., Plumbley, M.D.: Audio inpainting. IEEE Transactions on Audio, Speech, and Language Processing **20**(3), 922–932 (Mar 2012)
3. Andris, S., Peter, P., Weickert, J.: A proof-of-concept framework for PDE-based video compression. In: Proc. 32nd Picture Coding Symposium (PCS 2016). pp. 1–5. Nuremberg, Germany (Dec 2016)
4. Bertalmío, M., Sapiro, G., Caselles, V., Ballester, C.: Image inpainting. In: Proc. SIGGRAPH 2000. pp. 417–424. New Orleans, LI (Jul 2000)
5. Capon, J.: A probabilistic model for run-length coding of pictures. IRE Transactions on Information Theory **5**(4), 157–163 (Dec 1959)
6. Catmull, E., Rom, R.: A class of local interpolating splines. In: Barnhill, R.E., Riesenfeld, R.F. (eds.) Computer Aided Geometric Design, pp. 317–326. Academic Press, New York (1974)
7. Chen, Y., Ranftl, R., Pock, T.: A bi-level view of inpainting-based image compression. In: Proc. 19th Computer Vision Winter Workshop. pp. 19–26. Křtiny, Czech Republic (Feb 2014)
8. Crowley, P.: Exploring the Forest (2013), audio file, available under http://petercrowleyfantasydream.jimdo.com
9. de Boor, C.: A Practical Guide to Splines, Applied Mathematical Sciences, vol. 27. Springer, Berlin (1978)
10. Duchon, J.: Interpolation des fonctions de deux variables suivant le principe de la flexion des plaques minces. RAIRO Analyse Numérique **10**(3), 5–12 (Dec 1976)
11. Edler, B., Purnhagen, H.: Parametric audio coding. In: Proc. International Conference on Communication Technology Proceedings (WCC-ICCT 2000). vol. 1, pp. 614–617. Beijing, China (Aug 2000)
12. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: Proc. Seventh IEEE International Conference on Computer Vision. vol. 2, pp. 1033–1038. Corfu, Greece (Sep 1999)
13. 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 (Jul 2008)
14. Gemmeke, J.F., Van Hamme, H., Cranen, B., Boves, L.: Compressive sensing for missing data imputation in noise robust speech recognition. IEEE Journal of Selected Topics in Signal Processing **4**(2), 272–287 (Feb 2010)

15. Hoeltgen, L., Mainberger, M., Hoffmann, S., Weickert, J., Tang, C.H., Setzer, S., Johannsen, D., Neumann, F., Doerr, B.: Optimising spatial and tonal data for PDE-based inpainting. In: Bergounioux, M., Peyré, G., Schnörr, C. (eds.) Variational Methods in Image Analysis. De Gruyter, Berlin (2016)
16. Iijima, T.: Basic theory on normalization of pattern (in case of typical one-dimensional pattern). Bulletin of the Electrotechnical Laboratory **26**, 368–388 (Jan 1962), in Japanese
17. ISO/IEC: Information technology – coding of moving pictures and associated audio – part 3: Audio (1992), standard, ISO/IEC 11172-3
18. ISO/IEC: Information technology – generic coding of moving pictures and associated audio – part 7: Avanced audio coding (1992), standard, ISO/IEC 13818-7
19. ISO/IEC: Information technology – coding of audio-visual objects – part 3: Audio (2001), standard, ISO/IEC 14496-3
20. Maher, R.: A method for extrapolation of missing digital audio data. Journal of the Audio Engineering Society **42**(5), 350–357 (May 1994)
21. Mahoney, M.: Adaptive weighing of context models for lossless data compression. Tech. Rep. CS-2005-16, Florida Institute of Technology, Melbourne, FL (Dec 2005)
22. Mainberger, M., Hoffmann, S., Weickert, J., Tang, C.H., Johannsen, D., Neumann, F., Doerr, B.: Optimising spatial and tonal data for homogeneous diffusion inpainting. In: Bruckstein, A., ter Haar Romeny, B., Bronstein, A., Bronstein, M. (eds.) Scale Space and Variational Methods in Computer Vision, Lecture Notes in Computer Science, vol. 6667, pp. 26–37. Springer, Berlin (2011)
23. Masnou, S., Morel, J.M.: Level lines based disocclusion. In: Proc. 1998 IEEE International Conference on Image Processing. vol. 3, pp. 259–263. Chicago, IL (Oct 1998)
24. McAulay, R., Quatieri, T.: Computationally efficient sine-wave synthesis and its application to sinusoidal transform coding. In: Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP-88). pp. 370–373. New York, NY (Apr 1988)
25. Pennebaker, W.B., Mitchell, J.L.: JPEG: Still Image Data Compression Standard. Springer, New York (1992)
26. Peter, P., Hoffmann, S., Nedwed, F., Hoeltgen, L., Weickert, J.: Evaluating the true potential of diffusion-based inpainting in a compression context. Signal Processing: Image Communication **46**, 40–53 (Aug 2016)
27. Schmaltz, C., Peter, P., Mainberger, M., Ebel, F., Weickert, J., Bruhn, A.: Understanding, optimising, and extending data compression with anisotropic diffusion. International Journal of Computer Vision **108**(3), 222–240 (Jul 2014)
28. Schönlieb, C.B.: Partial Differential Equation Methods for Image Inpainting, Cambridge Monographs on Applied and Computational Mathematics, vol. 29. Cambridge University Press, Cambridge (2015)
29. Schuller, G.D., Yu, B., Huang, D., Edler, B.: Perceptual audio coding using adaptive pre-and post-filters and lossless compression. IEEE Transactions on Speech and Audio Processing **10**(6), 379–390 (Dec 2002)
30. Spanias, A., Painter, T., Atti, V.: Audio Signal Processing and Coding. Wiley, Hoboken (2006)
31. Taubman, D.S., Marcellin, M.W. (eds.): JPEG 2000: Image Compression Fundamentals, Standards and Practice. Kluwer, Boston (2002)
32. Vincent, E., Plumbley, M.D.: Low bit-rate object coding of musical audio using Bayesian harmonic models. IEEE Transactions on Audio, Speech, and Language Processing **15**(4), 1273–1282 (Apr 2007)

33. Weickert, J.: Theoretical foundations of anisotropic diffusion in image processing. In: Kropatsch, W., Klette, R., Solina, F., Albrecht, R. (eds.) Theoretical Foundations of Computer Vision, Computing Supplement, vol. 11, pp. 221–236. Springer, Vienna, Austria (1996)
34. Xiph.Org Foundation: Vorbis I specification (2015), https://xiph.org/vorbis/doc/Vorbis_I_spec.html