Audio Inpainting: Revisited and Reweighted

Ondřej Mokrý and Pavel Rajmic

***Abstract*—We deal with the problem of sparsity-based audio inpainting, i.e. filling in the missing segments of audio. A consequence of the approaches based on mathematical optimization is the insufficient amplitude of the signal in the filled gaps. Remaining in the framework based on sparsity and convex optimization, we propose improvements to audio inpainting, aiming at compensating for such an energy loss. The new ideas are based on different types of weighting, both in the coefficient and the time domains. We show that our propositions improve the inpainting performance in terms of both the SNR and ODG.**

***Index Terms*—Audio inpainting, sparse representations, proximal algorithms, Douglas–Rachford algorithm, Chambolle–Pock algorithm, energy loss compensation, amplitude drop.**

I. Introduction

AUDIO inpainting deals with missing samples in digital audio signals. Different algorithms were developed aiming at the restoration of the lost information. In practice, the typical loss of signal is in the form of a compact gap, for instance due to a dropout in Voice-over-IP communication.

The methods proposed by Janssen [1], [2] and Etter [3] are among the oldest (but most successful!) methods. They are based on autoregressive signal modeling; the missing samples are filled by linear prediction using autoregressive coefficients that are learned using the neighborhood of the gap. For a more comprehensive study into AR-based audio inpainting, see, for example, [4], [5], [6]. Approaches based on statistical methods were presented in [7] for the related problem of click removal, and the Bayesian approach to inpainting/declipping was introduced in [8].

A range of audio processing applications came along with the advent of sparse signal representations [9], [10]. The first work that used sparse signal synthesis for filling the missing samples [11] actually took over the term "audio inpainting" from the image processing field.

For gaps longer than approximately 100 milliseconds, all the above-described approaches start to fail. The main reason is that audio can usually be considered stationary only for a few tens of milliseconds. That is why other modeling approaches have been introduced for longer gaps: (generalized) sinusoidal modeling [12], [13], similarity graph approach [14], [15] or deep neural networks [16], [17].

In the present article, we concentrate on the classic case where the gap does not exceed 50 milliseconds. We emphasize that we assume filling a single, compact gap. Such a task is truly challenging, which explains why not many methods have been published on the topic, at least in comparison with the related field of audio declipping. Clipping is a non-linear distortion that degrades the signal all along its length; there is no compact-in-time loss, and moreover, some information is still available due to the knowledge of the clipping model. For the treatment of clipping, much more literature exists [18], [19], [20], [21], [22], [23], to name but a few. Note that another related problem is filling in the missing samples which are selected randomly, as done, for instance, in [24]. In such a case, the situation is close to clipping; inpainting is then relatively simple, since the occurrence of a significantly long sequence of missing samples is highly improbable.

For our scenario, where we treat short gaps, successful methods are typically model-based; an optimization problem is designed that contains the data fitting term and a regularizer. The regularizer usually penalizes the deviation of transformed signal's coefficients from the model under consideration. The transform used here is typically a kind of short-time spectral transform.

A problem with these approaches is that regularizers make the solution biased. This effect will be discussed later on, but let us reveal right now that in the case of audio inpainting, the bias manifests itself in the form of a signal's energy drop within the filled gap, see Fig. 1. The main goal of the article is to study this effect, design a number of methods that all aim at compensating it, and evaluate them in numerical experiments on a real audio.

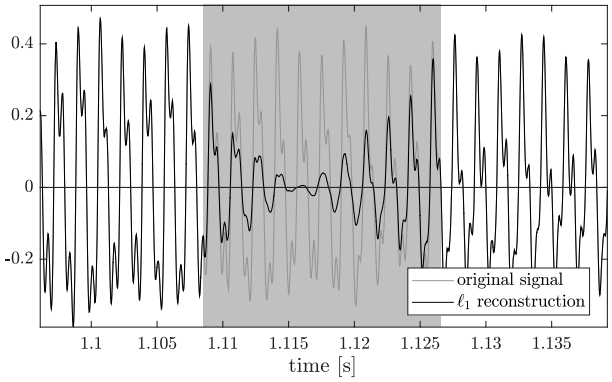


Fig. 1: Illustration of a typical energy loss inside the filled gap. The gap is visualized as a gray area throughout the paper.

In Sec. II, we summarize the basics of Gabor transforms that will be used for window-wise signal processing. Sec. III then inspects the synthesis and analysis models of sparse audio inpainting and introduces different types of weighting the signal coefficients. Sec. IV follows up with the idea that the weights are iteratively recomputed. Sec. V presents the unexpected effect of the shift of the Gabor system in time on the result of inpainting. A gradual approach is described in Sec. VI, where the gap is not filled at once but piece by piece, inspired by image inpainting methods. Sec. VII then closes the

O. Mokrý and P. Rajmic are with the SPLab at the Faculty of Electrical Engineering and Communication, Brno University of Technology, Czech Republic. E-mail: 170583@vutbr.cz, rajmic@feec.vutbr.cz

list of proposed algorithms with the time-domain approach. Finally, experiments are conducted and evaluated in Sec. VIII.

Note that we are not aware of any paper that would study the weighting of signal coefficients, apart from our preliminary study [25]. Maybe even more surprisingly, there is no audio inpainting article that would involve the *analysis* model, except the evaluation study [24].

II. Gabor systems

The audio inpainting model presented hereinafter is based on the time-frequency sparsity of audio signals. As the sparsifying transform, we use the Gabor transform, known also as the Short-time Fourier transform (STFT). The Gabor transform uses a single prototype window function which is translated in time and frequency-modulated [26], [27]. The window serves to localize the signal in time. In discrete time, the translation of the window, denoted $\mathbf{g}$, is characterized by the integer window shift a. The number of modulations is denoted M and will be referred to as the number of frequency channels.

There exist convenient combinations of $\mathbf{g}$ and the parameters a, M such that the resulting Gabor system forms a frame of $\mathbb{C}^L$, i.e. any signal $\mathbf{y} \in \mathbb{C}^L$ is representable in a stable way as the linear combination of the Gabor vectors [28], [29], [30]. Although Gabor bases can be constructed, overcomplete systems which allow non-unique signal representation are preferred. In the overcomplete setup, the synthesis operator $D\colon \mathbb{C}^N \to \mathbb{C}^L$ generates a signal of length L from the $N > L$ coefficients. Its adjoint, the analysis operator $D^*\colon \mathbb{C}^L \to \mathbb{C}^N$, produces coefficients out of the signal. Note that we treat the signal as a complex vector, although we work only with real signals.

In this paper, only Gabor *tight frames* will be used. They have convenient properties from both the theoretical and practical points of view. For example, the windows used in synthesis and analysis are identical (up to a scale factor). We will make use of the so-called Parseval tight frames, which in addition satisfy [27]

$$DD^* = Id, \quad \|D^*\mathbf{y}\|_2 = \|\mathbf{y}\|_2, \tag{1}$$

where Id in general denotes the identity operator; in this particular context, it is the identity on the space $\mathbb{C}^L$.

Going into greater detail, it is natural to require that the signal length L should be divisible by a. In this particular case, the Gabor system consists of $N = \frac{L}{a} \cdot M$ vectors $\{\mathbf{d}_n\}_{n=1,\dots,N}$, $\mathbf{d}_n \in \mathbb{C}^L$. We will refer to these vectors as the (frame) *atoms* and to the whole system as the *Gabor dictionary*.

The window $\mathbf{g}$ is usually identified with its shorter counterpart, built by keeping only the nonzero samples of $\mathbf{g}$. Therefore we refer to the support length $w := |\mathrm{supp}(\mathbf{g})|$ (i.e. the number of non-zero elements of $\mathbf{g}$) as the *window length*. We will work exclusively with finitely-supported windows in this article.

We use the fast implementation of Gabor transforms offered by the LTFAT toolbox [31], [32] in our computations, and we adopt its time-frequency conventions.

III. Synthesis and analysis models with weights

The sparse signal processing literature relied for long on the so-called synthesis model, where one seeks for a small number of coefficients which are then synthesized to produce the resultant signal [10], [33], [34], [35], [36]. More recently, the analysis model has been studied, where one looks directly for the signal, with the requirement that its coefficients after analysis should be sparse [19], [37]. Both approaches are equivalent if the synthesis/analysis operators are bijective, i.e. if the operators correspond to the bases for the signal/coefficient spaces.

In this section we introduce these two approaches in the audio inpainting problem. Besides, we explore several methods for atom weighting, in order to improve the performance of the restoration.

A. Problem formulation

Let $\mathbf{y}$ denote the time-domain signal. Let the indexes of missing (or unreliable) samples be known. This will be referred to as *the gap*. The rest of the samples will be considered non-degraded and will be called *reliable*.

It is natural to require that the recovered signal maintains consistency with the observed signal in the reliable part. To formally handle this requirement, we introduce the (convex) set Γ, the set of all feasible signals

$$\Gamma = \{\mathbf{z} \in \mathbb{C}^L \mid M_\mathrm{R}\mathbf{z} = M_\mathrm{R}\mathbf{y}\}, \tag{2}$$

where $M_\mathrm{R}\colon \mathbb{C}^L \to \mathbb{C}^L$ is the "reliable mask" projection operator. It maps a signal in $\mathbb{C}^L$ to another signal in $\mathbb{C}^L$, keeping the signal samples corresponding to the reliable part intact, while setting the others to zero.

The sparse audio inpainting can be formulated as a minimization problem. In the case of the linear synthesis model, we assume that the synthesis operator D allows any signal from $\mathbb{C}^L$ to be generated from an infinite number of choices of coefficients (due to the overcompleteness of the system). The sparse synthesis model aims at obtaining the highest sparsity representation that fits the reliable signal samples, formally

$$\arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \quad \text{s.t. } D\mathbf{x} \in \Gamma, \tag{3}$$

where $\|\cdot\|_0$ denotes the ℓ_0-pseudonorm, which simply counts the non-zero elements of the argument. Note that this is the same quantity as $|\mathrm{supp}(\cdot)|$. However, we keep both notations throughout the paper, since $|\mathrm{supp}(\cdot)|$ emphasizes the measurment of length in the time domain.

Solving optimization problems that involve $\|\cdot\|_0$ is NP-hard and thus computationally intractable. Therefore approximations of the true solution to (3) must be introduced. Probably the most common way today is to solve a relaxed minimization problem that involves the ℓ_1 norm instead of ℓ_0 [10], [38], allowing the use of convex optimization [33], [39].

In the formulations that follow we include the weighting vector $\mathbf{w} \in \mathbb{R}^N$, $\mathbf{w} > 0$. Its role is to assign potentially different weights to the coefficients, leading to a minimization of the weighted ℓ_1 norm. The relaxed synthesis formulation reads

$$\arg\min_{\mathbf{x}} \|\mathbf{w} \odot \mathbf{x}\|_1 \quad \text{s.t. } D\mathbf{x} \in \Gamma, \tag{4}$$

where $\|\cdot\|_1$ sums the magnitudes of elements of the argument, and $\odot$ denotes the entrywise (Hadamard) product. Problem (4) can be equivalently written in an unconstrained form as

$$\arg\min_{\mathbf{x}} \|\mathbf{w}\odot\mathbf{x}\|_1 + \iota_\Gamma(D\mathbf{x}), \tag{AIs}$$

where ι_C is the indicator function of a set C—it takes on zero value for elements belonging to C and infinity otherwise.

On the other hand, the analysis formulation is

$$\arg\min_{\mathbf{z}} \|\mathbf{w}\odot D^*\mathbf{z}\|_1 \quad \text{s.t. } \mathbf{z}\in\Gamma, \tag{5}$$

or, in an unconstrained form,

$$\arg\min_{\mathbf{z}} \|\mathbf{w}\odot D^*\mathbf{z}\|_1 + \iota_\Gamma(\mathbf{z}). \tag{AIa}$$

It is clear that the output of the analysis minimization (AIa) is directly the restored signal. On the contrary, the synthesis model (AIs) finds the optimal vector of coefficients, and the restored signal is obtained simply via the application of D.

B. Choosing the weights

A consequence of the ℓ_1 minimization is that not only a number of coefficients are pushed to zero, in order to make the coefficient vector sparse, but also the non-zero coefficients are automatically made smaller in magnitude than they could (and should) be. This is a commonly observed problem, which is called *bias* in the statistical community [40], [41], [42], [43].

In the context of the Gabor transform with a system of translated windows, notice that besides the just described global effect of the ℓ_1 norm, the coefficients corresponding to windows that overlap with the gap carry less information about the reliable signal than the coefficients corresponding to the reliable parts. As a consequence, a progressive loss of amplitude is typically observed in the restored signal, as already demonstrated in Fig. 1.

In order to compensate for this local effect, we naturally propose weighting the frame atoms—the less reliable information the atom carries, the lower the corresponding weight should be, resulting in less penalization in either (AIs) or (AIa). Note that such an idea already appeared in [11], where the authors used the weighting such that the ℓ_2 norms of atoms used for inpainting were normalized. In this article, we propose and examine several other ways to determine the weighting coefficients. For the particular choices, see Sec. VIII-C.

C. Proximal algorithms

The proximal splitting methodology is an efficient tool for iterative solution to large-scale convex minimization problems [44], [45], [46], [47]. Certain proximal algorithms are able to find the minimum of a sum of convex functions f_i, with mild assumptions about these functions, even when some of the functions f_i are composed with linear operators. Proximal algorithms perform iterations involving an evaluation of the so-called *proximal operators* related individually to each f_i, which is computationally much simpler than minimizing the composite functional by other means. We will use proximal algorithms to numerically solve our problems (AIs) and (AIa).

The proximal operator of a (convex) function $h\colon \mathbb{C}^N \to \mathbb{R}$ is the mapping $\operatorname{prox}_h\colon \mathbb{C}^N \to \mathbb{C}^N$ [44], [48]. This article will make use of the proximal operators of two particular functions. The proximal operator corresponding to the ℓ_1 norm is the well-known soft thresholding [49]. Conveniently, if the ℓ_1 norm is composed with elementwise weighting, $\operatorname{prox}_{\tau\|\mathbf{w}\odot\cdot\|_1} =: \operatorname{soft}_{\tau\mathbf{w}}$ can be shown to be another elementwise mapping, where each w_n only affects the threshold value, such that we write

$$\operatorname{soft}_{\tau\mathbf{w}}(\mathbf{x}) := \arg(\mathbf{x})\odot\max(|\mathbf{x}|-\tau\mathbf{w},0). \tag{6}$$

The operator $\arg(x)$ denotes the argument of x, where x is a complex number. In (6), this operation is extended elementwise to vectors. When $h=\iota_C$ is the indicator function of a convex set C, the related proximal operator $\operatorname{prox}_{\iota_C}(\mathbf{x})$ finds the vector in C closest to $\mathbf{x}$. Such an operator thus corresponds to the projection onto C and will be denoted proj_C.

The last useful property is related to the Fenchel–Rockafellar conjugate [39]. Given a convex f, the proximal operator of its conjugate f^* can be computed at virtually the same cost as prox_f due to the Moreau identity [46], [47]:

$$\operatorname{prox}_{\alpha f^*}(\mathbf{u}) = \mathbf{u} - \alpha\operatorname{prox}_{f/\alpha}(\mathbf{u}/\alpha) \quad \text{for } \alpha\in\mathbb{R}^+. \tag{7}$$

D. Solving the synthesis problem

Returning to the synthesis-based inpainting (AIs), notice that this problem minimizes the sum of two convex functions

$$f_1 = \iota_\Gamma\circ D, \quad f_2 = \|\mathbf{w}\odot\cdot\|_1. \tag{8}$$

It is convenient to use the Douglas–Rachford (DR) algorithm [44] to find the numerical solution. The building blocks of the DR algorithm are $\operatorname{prox}_{f_1}$ and $\operatorname{prox}_{f_2}$. In light of Sec. III-C, these operators are the projection and the generalized soft thresholding, respectively. In our case, $\operatorname{prox}_{f_1}$ has an explicit form

$$\operatorname{prox}_{\tau f_1}(\mathbf{x}) = \mathbf{x} - D^*M_{\mathrm{R}}D\mathbf{x} + D^*M_{\mathrm{R}}\mathbf{y}, \tag{9}$$

due D being assumed to be a tight Parseval frame [44].

The DR algorithm for inpainting is summarized by Alg. 1. The algorithm converges for any positive τ, but this parameter can largely affect the convergence speed. In the experiments that will follow, we use the usual termination criterion $\|\mathbf{x}^{(i)}-\mathbf{x}^{(i-1)}\| < \varepsilon\|\mathbf{x}^{(i-1)}\|$, where $\varepsilon>0$ is a chosen tolerance.

Algorithm 1: DR algorithm for inpainting

require: tight synthesis operator $D\colon \mathbb{C}^N\to\mathbb{C}^L$, observed signal $\mathbf{y}$, mask M_{R}, weights $\mathbf{w}$

choose parameter $\tau>0$
choose auxiliary variable $\mathbf{q}^{(0)}\in\mathbb{C}^N$ arbitrarily
set iteration counter $i=0$
repeat
  $\mathbf{x}^{(i)} = \operatorname{soft}_{\tau\mathbf{w}}(\mathbf{q}^{(i)})$
  $\mathbf{q}^{(i+1)} = \mathbf{x}^{(i)} + D^*M_{\mathrm{R}}D(2\mathbf{x}^{(i)}-\mathbf{q}^{(i)}) + D^*M_{\mathrm{R}}\mathbf{y}$
  $i\leftarrow i+1$
until *stopping criterion met*
return $\operatorname{proj}_\Gamma(D\mathbf{x}^{(i)})$

E. Solving the analysis problem

For solving (AIa), we use the Chambolle–Pock (CP) algorithm [45] with the assignment

$$f_1 = \iota_\Gamma, \quad f_2 = \|\mathbf{w} \odot \cdot\|_1. \tag{10}$$

The difference from the synthesis variant is that the argument of f_2 is the vector of coefficients of the signal after analysis by D^*. There is no explicit formula like (9) in this case and therefore the DR algorithm is not applicable here.

It is of advantage that $\mathrm{prox}_{f_1}(\cdot) = \mathrm{proj}_\Gamma(\cdot)$, since this is an elementwise operation. Specifically, the current samples in reliable positions are replaced by the corresponding samples of the observed signal while the samples in the gap are preserved. Formally, $\mathrm{proj}_\Gamma(\mathbf{x}) = (Id - M_\mathrm{R})\mathbf{x} + M_\mathrm{R}\mathbf{y}$. By defining $\mathrm{clip}_\mathbf{w}(\mathbf{x}) := \mathbf{x} - \mathrm{soft}_\mathbf{w}(\mathbf{x})$ and using the property $\mathrm{soft}_\tau(\mathbf{x} \cdot \tau) = \tau \cdot \mathrm{soft}_1(\mathbf{x})$ for any $\tau > 0$, we can rewrite Eq. (7) for our $f_2 = \|\mathbf{w} \odot \cdot\|_1$ as

$$\mathrm{prox}_{\sigma f_2^*}(\mathbf{x}) = \mathbf{x} - \sigma \cdot \mathrm{soft}_{\mathbf{w}/\sigma}(\mathbf{x}/\sigma) = \mathrm{clip}_\mathbf{w}(\mathbf{x}). \tag{11}$$

The CP algorithm for inpainting is summarized in Alg. 2. The convergence is guaranteed if the step sizes τ and σ are set such that $\tau\sigma\|D\|^2 \leq 1$ [47]. We would like to use the termination criterion analogously to the synthesis case, and therefore we measure the relative difference of the norms $\|D^*\mathbf{y}^{(i)} - D^*\mathbf{y}^{(i-1)}\| < \varepsilon\|D^*\mathbf{y}^{(i-1)}\|$, where $\varepsilon > 0$ is the tolerance. Notice, however, that the operator D^* is linear and, being a Parseval tight frame, it preserves the vector norms (see Eq. (1)), hence the criterion is equivalent to $\|\mathbf{y}^{(i)} - \mathbf{y}^{(i-1)}\| < \varepsilon\|\mathbf{y}^{(i-1)}\|$.

Algorithm 2: CP algorithm for inpainting

require: tight synthesis operator $D\colon \mathbb{C}^N \to \mathbb{C}^L$, observed signal $\mathbf{y}$, mask M_R, weights $\mathbf{w}$

choose $\tau, \sigma > 0$ satisfying $\tau\sigma\|D\|^2 \leq 1$
choose primal variable $\mathbf{p}^{(0)} \in \mathbb{C}^L$ and dual variable $\mathbf{q}^{(0)} \in \mathbb{C}^N$ arbitrarily
set output variable $\mathbf{y}^{(0)} = \mathbf{p}^{(0)}$
set iteration counter $i = 0$
repeat
  $\mathbf{q}^{(i+1)} = \mathrm{clip}_\mathbf{w}(\mathbf{q}^{(i)} + \sigma D^*\mathbf{y}^{(i)})$
  $\mathbf{p}^{(i+1)} = \mathrm{proj}_\Gamma(\mathbf{p}^{(i)} - \tau D\mathbf{q}^{(i+1)})$
  $\mathbf{y}^{(i+1)} = 2\mathbf{p}^{(i+1)} - \mathbf{p}^{(i)}$
  $i \leftarrow i + 1$
until *stopping criterion met*
return $\mathrm{proj}_\Gamma(\mathbf{y}^{(i)})$

F. Computational complexity

It is clear from the pseudocodes that both the proximal algorithms perform one analysis (operator D^*) and one synthesis (operator D) in each iteration—note that the analysis of the reliable part of the input signal $\mathbf{y}$ also appears in the DR algorithm, but this can be precomputed. Due to the fact that the complexity of D and D^* significantly exceeds the cost of other operations involved, we can summarize that the CP and the DR algorithms for audio inpainting are identically demanding.

IV. Iterative reweighting

In [50], audio declipping using the so-called *re*weighted ℓ_1 minimization was presented. In such an approach, the ℓ_1 norm of the coefficients is weighted by $\mathbf{w}$ as in the previous section, but the idea behind the weights is different here: The restoration task is solved *repeatedly*, and the weights change in the repetitions, based on the values of the coefficients from the current solution. The benefit is that using such a procedure, the significant coefficients can be adaptively penalized less and less, while the insignificant coefficients are more and more pushed towards zero, leading to a better approximation of sparsity (and to avoiding the bias, to some extent). Note also that simple examples can be found where this strategy fails to find the optimal sparse solution [41].

As a matter of fact, [50] applied the reweighting strategy only in the synthesis variant of declipping. We adapted their approach to audio inpainting already in [25] and include it here for the context. Recall that the shift of task from declipping to inpainting is done easily by redefining the set of feasible solutions Γ. The resulting synthesis-based reweighted inpainting is summarized in Alg. 3. Note that step 4 of the algorithm represents the weighted synthesis audio inpainting, and therefore this step is carried out by Alg. 1.

Naturally, the idea of reweighting can be included in the analysis-based recovery, which was proposed already in [41], but not presented in the field of audio restoration. In contrast to the synthesis case, the analysis-based algorithm requires an additional application of the analysis operator, in order to travel from the signal space to the coefficient domain and thus to be able to assign the weights. The algorithm is summarized in Alg. 4. This time, step 4 is solved by Alg. 2.

Algorithm 3: Synthesis reweighted ℓ_1 for inpainting

require: tight synthesis operator $D\colon \mathbb{C}^N \to \mathbb{C}^L$, set of feasible solutions $\Gamma \subset \mathbb{C}^L$, parameters K, ε, $\delta > 0$

set iteration counter $k = 1$
set initial weights $w_i^{(1)} = 1$, $i = 1, \ldots, N$
repeat
  $\mathbf{z}^{(k)} = \arg\min_\mathbf{z} \|\mathbf{w}^{(k)} \odot \mathbf{z}\|_1$ s. t. $D\mathbf{z} \in \Gamma$
  $w_i^{(k)} = 1/(|z_i^{(k)}| + \varepsilon)$, $i = 1, \ldots, N$
  $k \leftarrow k + 1$
until $k > K$ **or** $\|\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\|_2 < \delta$
return $\mathbf{x} = D\mathbf{z}^{(k-1)}$

V. Offset: possibly a strong influencer

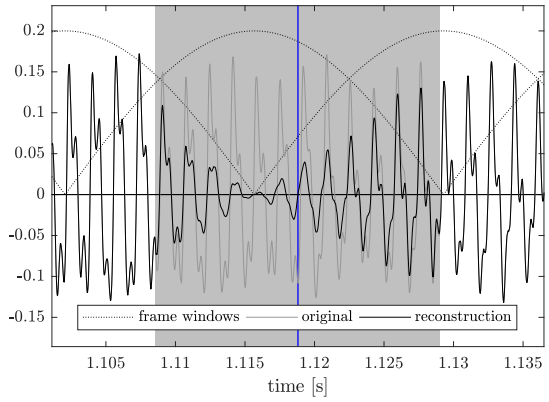
The reader can see from Fig. 1 that the minimum of the amplitude in the reconstructed signal does not appear exactly in the center of the gap. The positioning of the Gabor system with respect to the location of the gap plays a role here. In Fig. 2, ℓ_1 reconstructions using two different Gabor systems are presented. The second system has been shifted such that the "central" Gabor window fits the center of the gap. Consequently, the energy of the reconstructed signal decreases symmetrically within the gap.

We call the amount of the shift of the system *the offset*. We need to centralize the energy loss since methods compensating

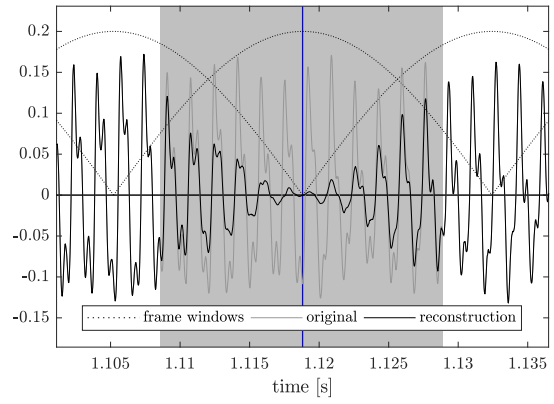
Algorithm 4: Analysis reweighted ℓ_1 for inpainting

require: tight synthesis operator $D\colon \mathbb{C}^N \to \mathbb{C}^L$, set of feasible solutions $\Gamma \subset \mathbb{C}^L$, parameters K, ε, $\delta > 0$

set iteration counter $k = 1$
set initial weights $w_i^{(1)} = 1$, $i = 1, \ldots, N$
repeat
  $\mathbf{x}^{(k)} = \arg\min_{\mathbf{x}} \|\mathbf{w}^{(k)} \odot D^*\mathbf{x}\|_1$ s. t. $\mathbf{x} \in \Gamma$
  $\mathbf{z}^{(k)} = D^*\mathbf{x}^{(k)}$
  $w_i^{(k)} = 1/(|z_i^{(k)}| + \varepsilon)$, $i = 1, \ldots, N$
  $k \leftarrow k + 1$
until $k > K$ **or** $\|\mathbf{z}^{(k)} - \mathbf{z}^{(k-1)}\|_2 < \delta$
return $\mathbf{x}^{(k-1)}$



(a) no offset



(b) full offset

Fig. 2: Visualization of energy drop in the gap and the effect of the offset. The center of the gap is denoted by the blue line.

it in the time domain (Sections VI and VII) require such a symmetric setting. In this section, we will propose two approaches—*half* and *full* offset—that will ensure the energy drop will be symmetric after running the inpainting algorithm described in Sec. III.

There are two ways of choosing the offset: either the center of the gap corresponds to the center of a Gabor window (we refer to this as the *full offset*) or the gap center is just in the middle between two neighboring windows (*half offset*), see the illustrative sequence of Gabor windows in Fig. 3. Note that when we refer to the center of the window with the index $k+1$, we mean the signal index $1 + k \cdot a$ [51], assuming that the indexes of the signal samples start from 1. The selection of the *offset* value is formalized in Alg. 5.

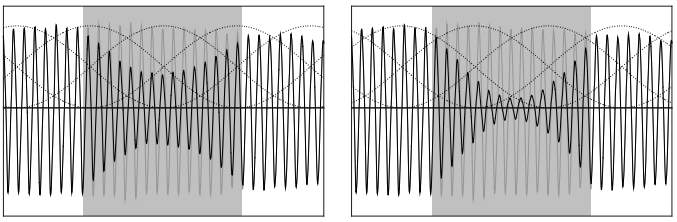
Algorithm 5: Computing the *offset* value

require: indexes s and f of the first and the last missing sample within the original signal, Gabor window shift a

compute the central index of the gap $c = \lceil (s+f)/2 \rceil$
$k = \lfloor (c-1)/a \rfloor$
if *full offset* **then**
  $d = 1 + k \cdot a$
end
if *half offset* **then**
  $d = 1 + k \cdot a \ + \lceil a/2 \rceil$
end
return $\mathit{offset} = c - d$

On line 2, the index of the nearest window preceding the index c is found, i.e. $k \in \mathbb{N} : 1 + k \cdot a \leq c \leq 1 + (k+1) \cdot a$. In the audio inpainting framework, the *offset* value is used as an input of the algorithm for support restriction [51].

Fig. 3 shows that the inclusion of one of the two offset choices in the ℓ_1 inpainting algorithm ensures a symmetric energy drop. We observe in this simulated example that the energy drop differs substantially for the two choices.



(a) full offset (b) half offset

Fig. 3: The two offset settings and their influence on the signal energy inside the filled gap. Original and restored signals are shown together with the individual shifts of Gabor windows.

Does the full offset systematically provide better reconstruction than the half offset? Better performance is *on average* reached with the half offset. Sec. VIII will show an in-depth analysis and the dependence of the answer on the model and on whether the ℓ_1 norm is weighted.

VI. Gradual inpainting

Sec. III proposed a method that used weighting the signal coefficients to reduce the energy drop within the filled gap. Sections VI and VII propose different methods for compensating for the same artifact, but they are, by contrast, based on processing directly in the time domain.

As can be seen in the figures presented, the closer to the borders of the gap we are, the better the reconstruction is. The *gradual* ℓ_1 *inpainting* starts from this observation: Each time the signal in the gap is restored by ℓ_1 minimization, small chunks of the just computed samples from the beginning

and from the end of the filled gap are fixed and treated as reliable from this moment on. At the next grade, inpainting is performed on a gap that is accordingly shorter. The process is repeated until the whole gap is filled, which is formalized in Alg. 6. Step 4 can utilize any of the above-presented signal models/algorithms.

Algorithm 6: Gradual inpainting

require: degraded signal $\mathbf{y}$, indexes s and f of the first and the last missing sample, respectively

1 set grade counter $g = 0$
2 set $\mathbf{y}^{(0)} = \mathbf{y}$
3 **while** $s \leq f$ **do**
4 find $\mathbf{y}^{(g+1)}$ as the solution to inpainting problem with degraded signal $\mathbf{y}^{(g)}$, with samples missing from s to f
5 set step parameter $r \in \mathbb{N}$
6 $s \leftarrow s + r$ `// shrink from the left`
7 $f \leftarrow f - r$ `// shrink from the right`
8 $g \leftarrow g + 1$
9 **end**
10 **return** $\mathbf{y}^{(g)}$

Using the notation in problems (AIs) and (AIa), this algorithm produces a hierarchy of feasible solutions $\Gamma^{(g)} \subset \Gamma^{(g-1)}$ and can use potentially different weighting vectors $\mathbf{w}^{(g)}$ at each grade. Clearly, the solution computed at grade g satisfies

$$\mathbf{y}^{(g)} \in \Gamma^{(g+1)}. \tag{12}$$

Note that if no weighting is used at all, i.e. $\mathbf{w}^{(g)} \odot \mathbf{x} = \mathbf{x}$ at all grades g, the same objective function is minimized at each grade, and (12) induces $\mathbf{y}^{(g+1)} = \mathbf{y}^{(g)}$. Therefore, the gradual approach with no weights does not find a solution any different from the all-at-once approach. The proposed weighting of the atoms thus forms one of the possible approaches, allowing the gradual algorithm to lead to a different (and possibly better) solution to the inpainting problem.

VII. Time domain compensation for energy loss

In this section, we propose a heuristic method for compensating for energy loss after running an ℓ_1 minimization. The idea is to take the result of the ℓ_1 minimization and modify its outcome by entrywise multiplication of the recovered gap in the time domain by a compensation curve, in order to increase its amplitude.

A. Notation and requirements

To formulate the demands on the compensation curve, assume for the moment that the signal is a function on the interval $\langle 0, T\rangle$ with the gap spreading across $\langle s, f\rangle \subset \langle 0, T\rangle$.

Denote $c = (s + f)/2$ the center of the gap. A curve $q(t)$ is suitable for the energy loss compensation, if it satisfies the following natural conditions:

- $q(t)$ is smooth on $\langle 0, T\rangle$,
- $\frac{\mathrm{d}q}{\mathrm{d}t} = 0 \quad$ for $\quad t \in \{s, f\}$,
- $q(t) = 1 \quad$ for $\quad t \notin \langle s, f\rangle$,
- $q(t)$ is non-decreasing on (s, c), non-increasing on (c, f),
- $q(s + t) = q(f - t) \quad$ for $\quad t \in \langle 0, f - s\rangle$.

The first three conditions ensure that the adjustment of amplitude inside the filled gap is smooth. The last two conditions reflect the observation that the greatest energy drop is in the center of the filled gap and it is symmetric. In the discrete setting, the compensation vector $\mathbf{q}$ of length h is obtained by sampling $q(t)$ in the interval $\langle s, f\rangle$.

B. Computing the compensation curve

Finding a good heuristic curve $q(t)$ must be based on reliable information in the neighborhood of the gap. Our approach assumes that in the neighborhood of the gap, signal characteristics do not change too dynamically. Additional gaps are artificially created and inpainted, which provides perfect local information about the energy decrease, since the reference original signal in these artificial gaps is available. This information is then used to compensate for the energy loss in the gap that was originally treated.

Below, we formalize the described idea. See also Fig. 5, which illustrates the main steps of the algorithm.

1) In the (reliable) neighborhood of the gap, create new gaps.
2) Inpaint the original as well as the additional gaps, using the same setting.
3) For all the new gaps, compute how the energy progresses through the gap and through the corresponding portion of the reliable signal. The energy progression is estimated via m overlapping signal segments covering the whole inpainted gap. For the following steps, the segments should be distributed for all the gaps *in the same way*, such that the information is transferable into the time instants $s < t_1 < \cdots < t_m < f$ in the original gap (see Fig. 4).
4) Find the multipliers $\mathbf{m} \in \mathbb{R}^m$, such that the difference in energy between the original signal and the filled parts is minimized (see Eq. (14)).
5) Compute $\mathbf{n} = \sqrt{\mathbf{m}}$ using the entrywise square root[1].
6) Enforce symmetry[2] of $\mathbf{n}$ by updating

$$n_i \leftarrow \frac{n_i + n_{m+1-i}}{2}, \qquad 1 \leq i \leq \left\lfloor \frac{m}{2} \right\rfloor \tag{13a}$$

$$n_i \leftarrow n_{m+1-i}, \qquad \left\lfloor \frac{m}{2} \right\rfloor + 1 \leq i \leq m. \tag{13b}$$

7) The function $q(t)$ is obtained by cubic spline interpolation[3], which for $t_1, t_2, \ldots, t_m$ attains the values of the vector $\mathbf{n}$, and at points s and f its value is 1 and derivative 0, respectively.
8) The vector $\mathbf{q}$ is obtained from $q(t)$ by equidistant sampling in the interval $\langle s, f\rangle$.

To clarify step 4, define two matrices. Let the columns of the matrix $\mathbf{X}$ be formed by the vectors of energy progression from the additional gaps. Let the matrix $\mathbf{Y}$ contain the respective

[1]The vector $\mathbf{m}$ represents the ratios of energy, while we need the ratio of signal amplitude. This corresponds to the square root of the ratios of energy.

[2]We use the offset described in Sec. V, which makes the assumption of symmetric energy drop realistic.

[3]We use the MATLAB function `spline`, see https://www.mathworks.com/help/matlab/ref/spline.html.

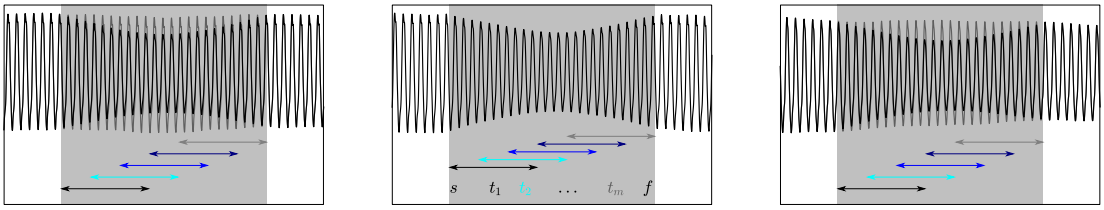
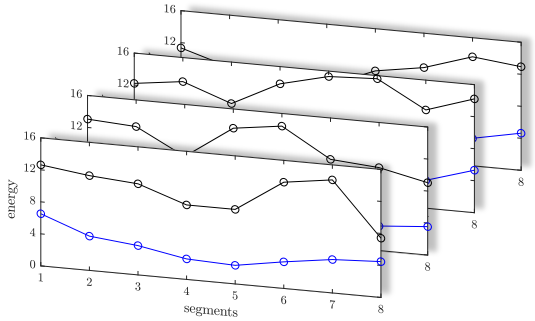
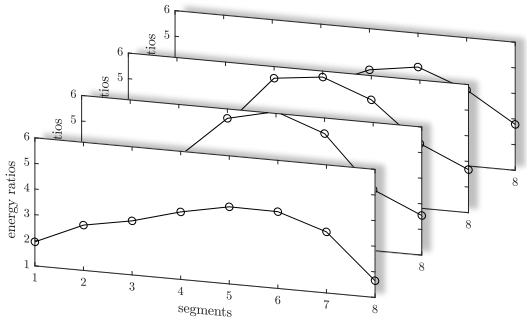


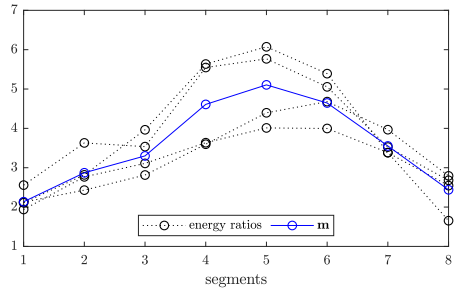
Fig. 4: Illustration of how the segments are distributed inside the gap to compute the energy progression. For the original gap (middle plot), two artificial gaps are created (left and right), where the reference signal (gray) is available. The distribution of $m = 5$ segments, represented by the arrows, is the same in all three cases.



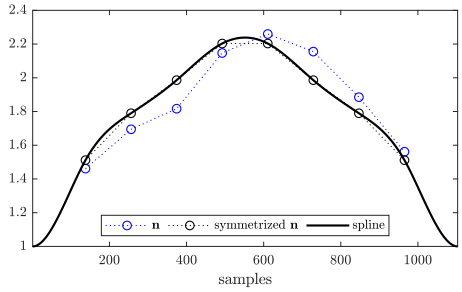
(a) the segmentwise energy progression in the original signal (black) and in the filled gap (blue)



(b) the ratio of the energy of the original signal and the filled gap through the segments



(c) the individual ratios of the energy together with the vector $\mathbf{m}$ computed as in Eq. (14)



(d) the symmetrization of the vector $\mathbf{n} = \sqrt{\mathbf{m}}$ and the resulting vector $\mathbf{q}$ found by spline interpolation

Fig. 5: Visualization of the direct time domain compensation for energy loss on a typical signal excerpt. For all the plots (i.e. for 4 additional new gaps), the energy is computed in $m = 8$ overlapping segments.

values of energy from the original signal. Note that this way, both matrices have m rows and the number of columns is the number of additional gaps (created in step 1). The multipliers $\mathbf{m}$ are then the optimizers of

$$\mathbf{m} = \arg\min_{\mathbf{m}' \in \mathbb{R}^m} \|\mathbf{Y} - \mathrm{diag}(\mathbf{m}') \cdot \mathbf{X}\|_2^2. \tag{14}$$

To understand Eq. (14), imagine a column in $\mathbf{X}$ (the energy of the inpainted signal). Multiplying it elementwise by $\mathbf{m}$, we want to reach as close as possible to the corresponding column of $\mathbf{Y}$ (original signal). The need for the minimization comes from the fact that the vector $\mathbf{m}$ is common to all the columns.

Eq. (14) is a least squares problem and its solution can be written explicitly using the entries of matrices $\mathbf{X}$ and $\mathbf{Y}$ as

$$m_i = \frac{\sum_j y_{ij} x_{ij}}{\sum_j x_{ij}^2}, \quad i = 1, \ldots, m, \tag{15}$$

which corresponds to a very fast procedure.

VIII. Experiments

This section presents a numerical evaluation of the above-described approaches to energy loss compensation. The last experiment shows an overall comparison.

A. Performance measures

As the standard performance measure, we use the signal-to-noise ratio (SNR), defined as

$$\mathrm{SNR}\left(\mathbf{y}_{\mathrm{orig}}, \mathbf{y}_{\mathrm{inp}}\right) = 10 \cdot \log_{10} \frac{\|\mathbf{y}_{\mathrm{orig}}\|_2^2}{\|\mathbf{y}_{\mathrm{orig}} - \mathbf{y}_{\mathrm{inp}}\|_2^2} \quad [\mathrm{dB}], \tag{16}$$

where $\mathbf{y}_{\mathrm{inp}}$ stands for the recovered signal and $\mathbf{y}_{\mathrm{orig}}$ denotes the original signal [11]. Recall that the very last step of our reconstruction algorithms is the projection onto the set Γ. In our implementation, we therefore evaluate the difference in the gap only, since elsewhere $\mathbf{y}_{\mathrm{orig}} - \mathbf{y}_{\mathrm{inp}}$ contains solely zeros. Obviously, higher SNR values reflect better reconstruction.

The SNR formula does not compensate for the length of the filled gap. This is the reason why only gaps of identical lengths will be taken into account in our comparisons. Note that we compute the average SNR by first computing the particular values of SNR in dB, and then taking the average.

For the final, overall comparison, we also included the PEMO-Q criterion [52]. This tool provides an evaluation that takes the model of human auditory system into account, thus being closer to the subjective evaluation than the SNR. The measured quantity called *objective difference grade* (ODG) can be interpreted as the degree of perceptual similarity between $\mathbf{y}_{\text{orig}}$ and $\mathbf{y}_{\text{inp}}$. The ODG attains values from -4 (very annoying) up to 0 (imperceptible), reflecting the effect of audio artifacts in the restored signal.

B. Evaluation setup

We use a collection of ten music recordings sampled at 44.1 kHz, with different levels of sparsity with respect to the Gabor representation. Our signals were chosen from the EBU SQAM dataset [53]. In each test instance, the input was a signal containing 8 gaps at random positions. The lengths of the gaps ranged from 5 ms up to 50 ms.

As the default choice in the tests, we used a tight Gabor frame with the Hann window of length $w = 2800$ samples (approximately 64 ms), window shift $a = 700$ samples and with $M = 2800$ frequency channels.

The Douglas–Rachford (DR) and Chambolle–Pock (CP) algorithms are used for the signal recovery, see Sections III-D and III-E, respectively. The DR algorithm uses $\tau = 0.2$, the CP algorithm uses $\tau = 0.2$ and $\sigma = 5$. Iterations are terminated if the proposed criterion with $\varepsilon = 5 \cdot 10^{-4}$ is satisfied or alternatively after 500 iterations.

For the sake of the overall comparison in Sec. VIII-G, the frame-wise Janssen algorithm [1], which is based on linear prediction, was also included, as used in [11]. Furthermore, the precursor of sparsity-based methods, the Orthogonal Mathching Pursuit (OMP), is included [11]. As the last competitor, we chose the SPAIN algorithm both in its synthesis and in its analysis variant [54]. SPAIN used the same window and overlap as the Gabor transform did.

C. Weighting the atoms and offset

The motivation for weighting the atoms of the Gabor frame was presented in Section III-B. Now we comment in detail on the choices of the weights. Let $\mathbf{d}_n$ be a Gabor atom and let $M_{\text{R}}\mathbf{d}_n$ be its part corresponding to the reliable part of the signal, see Fig. 6. We propose five different formulas for choosing the weights $\mathbf{w}$, including the vector of ones representing the non-weighted case. The proposed formulas are arranged according to the variance of obtained weights, in ascending order.

(a) $w_n = 1$ — no weighting — (`none`)

(b) $w_n = \dfrac{|\text{supp}(M_{\text{R}}\mathbf{d}_n)|}{|\text{supp}(\mathbf{d}_n)|}$ — support-based — (`supp`)

(c) $w_n = \dfrac{\|M_{\text{R}}\mathbf{d}_n\|_1}{\|\mathbf{d}_n\|_1}$ — ℓ_1 norm-based — (`abs`)

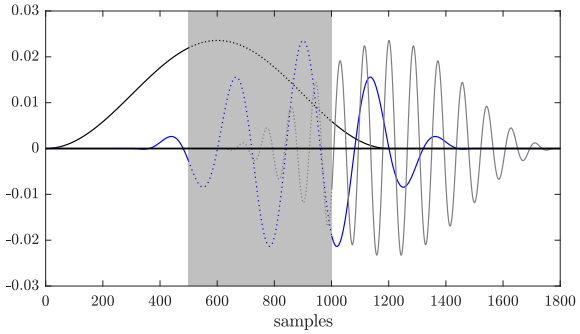


Fig. 6: Three examples of Gabor atoms $\mathbf{d}_n$ with different modulations, overlapping with the gap. Only their real parts are depicted. Solid lines indicate their reliable part $M_{\text{R}}\mathbf{d}_n$.

(d) $w_n = \dfrac{\|M_{\text{R}}\mathbf{d}_n\|_2}{\|\mathbf{d}_n\|_2}$ — ℓ_2 norm-based — (`norm`)

(e) $w_n = \dfrac{\|M_{\text{R}}\mathbf{d}_n\|_2^2}{\|\mathbf{d}_n\|_2^2}$ — energy-based — (`energy`)

Note that [11] used the weighting based on the ℓ_2 norm in their synthesis model, such that the ℓ_2 norms of atoms $M_{\text{R}}\mathbf{d}_n$ were made identical. This corresponds to our case (d). The difference from [11] is that we use the ℓ_1 approach instead of the greedy solver.

Fig. 7 plots the values of $\mathbf{w}$ for the proposed methods (b)–(e) for a fixed gap length. It illustrates that different options provide different weights. Furthermore, observe that all the values are strictly greater than zero, which follows from the window length being greater than the gap length. Should the opposite situation occur, some weights could be set to zero using any of the formulas (b)–(e). This would result in some coefficients not being penalized at all, while not being bound by the reliable signal parts either. In the iterative solver, the outcome would be that these coefficients would keep their initial value.

Before proceeding to the performance evaluation based on choosing the weighting types, we analyze the effect of the offset (Sec. V). The motivation is that if a preferable offset option is found, then this type of offset will be used in all of the experiments focusing on other parameters. Recall that Figures 2 and 3 illustrate not only the connection between the value of offset and the position of minimal amplitude (inside the gap), but also the difference that the choice of offset variant makes.

To analyze the difference between the choice of full or half offset, the SNR values for weighted ℓ_1 inpainting are plotted in Fig. 8. Two observations are clear from the figure. First, there is no obvious way of choosing the offset value, since no clear dependence on the SNR value or weighting type is observed. However, the results slightly favor the half offset (approx. 54 % of data points in Fig. 8a and 58 % in Fig. 8b lie above the diagonal line). Second, the SNR values exhibit much more variation in the synthesis case, compared to the analysis case. The two observations suggest that the choice of offset is not crucial in the analysis-based inpainting, whereas it could affect the results based on the synthesis model (due to the larger variance).

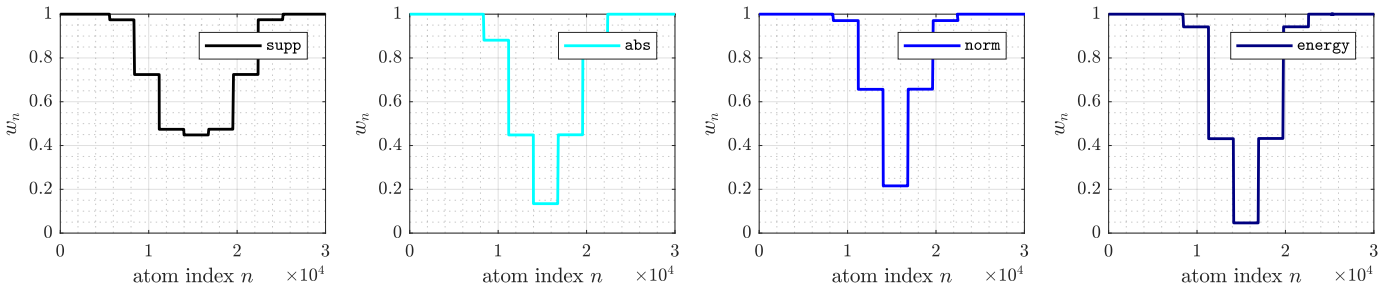


Fig. 7: Illustration of the weighting variants (besides the obvious case of constant weights). The weights were computed using the Gabor frame with a 64 ms long Hann window, overlap 75 % and the number of frequency channels equal to the length of the window in samples. The gap length was 35 ms.

Since it is convenient to fix as many parameters as possible for the following comparisons, and also the whole evaluation is based on an average performance of the algorithms, half offset will be the default choice in the subsequent experiments. For the results with full offset, see the accompanying repository (link in Sec. IX).

Fig. 9 shows the influence of weighting (Sec. III-B) on the results. Regarding the basic, non-weighted models, one may observe that the analysis-based inpainting performs slightly better compared to the synthesis-based case, especially for the middle-length gaps. It is, however, more interesting to notice that introducing the weights does not have the same effect in both models. The performance of the analysis model improves as the variance of the involved weights grows (see Fig. 7). In the synthesis model, on the other hand, weighting by the ℓ_1 or ℓ_2 norm leads to a consistent improvement, whereas the other choices may even decrease the value of SNR. Also note that in the synthesis model, the weighting is most beneficial for middle length gaps, while in the analysis model, the improvement increases with the gap length.

D. Iterative reweighting

Recall that in the iterative approach (Sec. IV), the iterative weights (denoted `iterative`) depend on both the signal and the time-frequency transform, and that they are computed differently in the synthesis- and analysis-based models. In the analysis-based Alg. 4, the new weights are computed from coefficients in the range space of D^*, which is generally not the case of the synthesis-based Alg. 3.

For these reasons, Fig. 8 and the earlier decision to use half offset are irrelevant in the case of iterative reweighting. Fig. 10 thus shows the average values of SNR for both the half and the full offset, in comparison with the simple non-weighted approach. Fig. 10a shows that in the synthesis model, the iterative reweighting provides consistent (however small) improvement for longer gaps and it is not much dependent on the offset choice. Contrarily, Fig. 10b suggests that the choice of offset is crucial in the reweighted analysis case. Note that this result is in contrast to the observation for the non-iterative weighting, where the choice of the offset did not significantly affect the results in the analysis case. Note also that although the iterative reweighting is highly beneficial for longer gaps with half offset, it decreases the average performance for the shortest gaps, independently of the offset variant.

E. Gradual inpainting

The average SNR values for gradual inpainting introduced in Sec. VI are presented in Fig. 11. As mentioned above, the gradual approach needs to be fused with another modification of the ℓ_1 inpainting to produce sensible results. In this experiment, weighting the atoms was chosen with the weighting formula based on the results described in Sec. VIII-C. The only other parameter of the method is the number of samples r taken as reliable from the left and right sides of the gap at each grade.

The results are presented for different choices of r as a fraction of the gap length h. In the synthesis case, Fig. 11a shows that the gradual algorithm is beneficial compared to the reference (the non-gradual method) for long gaps. In the analysis case, on the other hand, Fig. 11b clearly indicates that its performance via the gradual approach does not improve. Fig. 11c illustrates that the variance in the results in Fig. 11b is explained by the variance due to the approximate solution to the optimization problems involved. When any minor computational error appears, it is amplified at each grade of the gradual algorithm. Such a problem does not occur in the synthesis case, which was not expected.

Finally, note that even with the positive effect of the gradual approach, the synthesis model did not reach the quality of the analysis model.

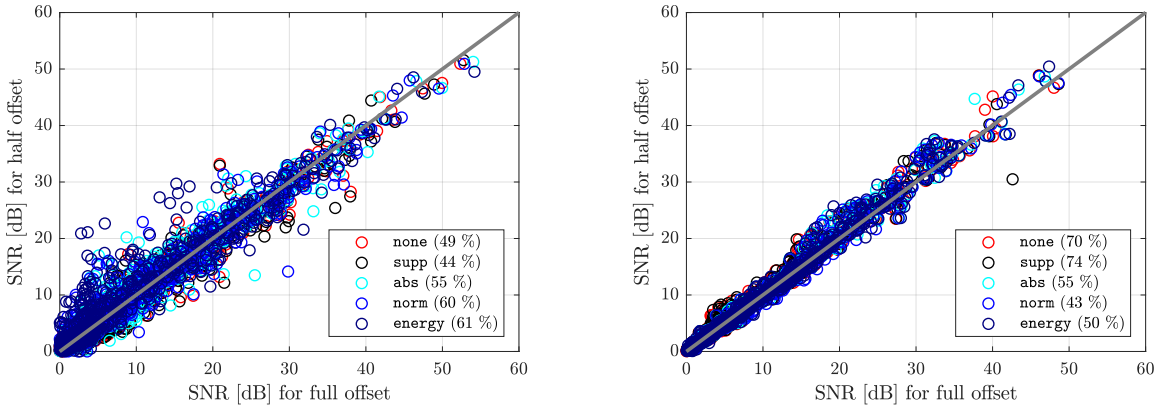
F. Direct time domain compensation for energy loss

The method of direct time domain compensation (Sec. VII) depends on a larger number of parameters, compared to the previous techniques. They are:

- the number of additional artificial gaps (denoted `gaps`),
- the positions of these gaps in the signal,
- the number of segments from which the evolution of energy is computed (denoted m),
- the length of these segments.

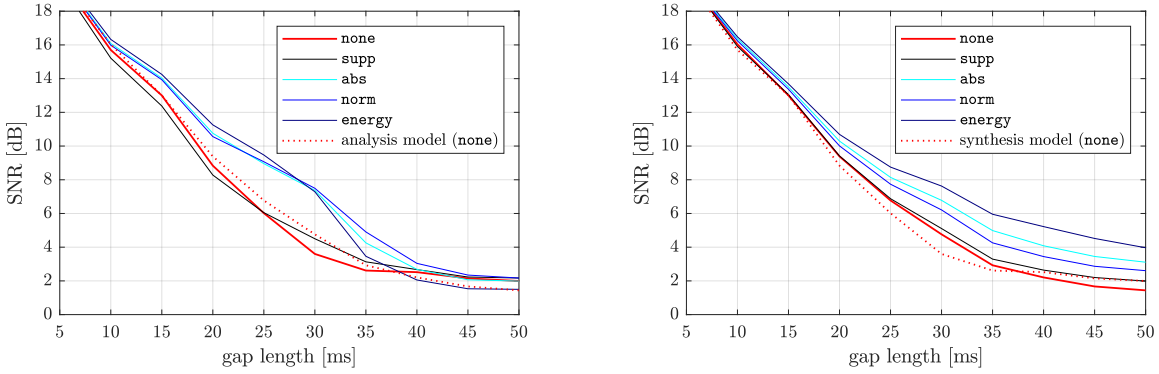
In our experiment, all the parameters except `gaps` have been fixed as follows:

- the additional gaps are symmetrically distributed around the initial gap, starting w samples away from the edge of the gap and then shifted by $w/2$ samples,
- the number of segments $m = 10$ and the length of each segment is $h/4$ (i.e. the segments are overlapping).



(a) synthesis model (b) analysis model

Fig. 8: Comparison of the two offset approaches, combined with weighting the frame atoms. Every point represents a single test instance. For better orientation, the diagonal line is shown. The percentage in brackets represents the fraction of instances above the diagonal line, given the weighting type.



(a) synthesis model (b) analysis model

Fig. 9: Evaluation of the methods for weighting atoms. The simple (non-weighted) variant of the concurrent model is also depicted by dots as a reference.

Fig. 12 shows the results of the experiment. Two approaches are tested in both the synthesis and the analysis model. First, the time domain approach is meant to be a competitor of weighting the atoms in the coefficient domain (plots 12a and 12b). Second, the weighting in the coefficient and the time domains is combined to provide even more energy compensation (plots 12c and 12d).

One can conclude that except the synthesis model combined with weighting the atoms (Fig. 12c), the time domain compensation leads to an improvement by a few dB. The results are remarkable, especially when the analysis model is used. Although Fig. 12b shows that the time domain compensation does not surpass the weighting of the atoms (compare with the reference in Fig. 12d), the combination of both approaches proves to be the best choice.

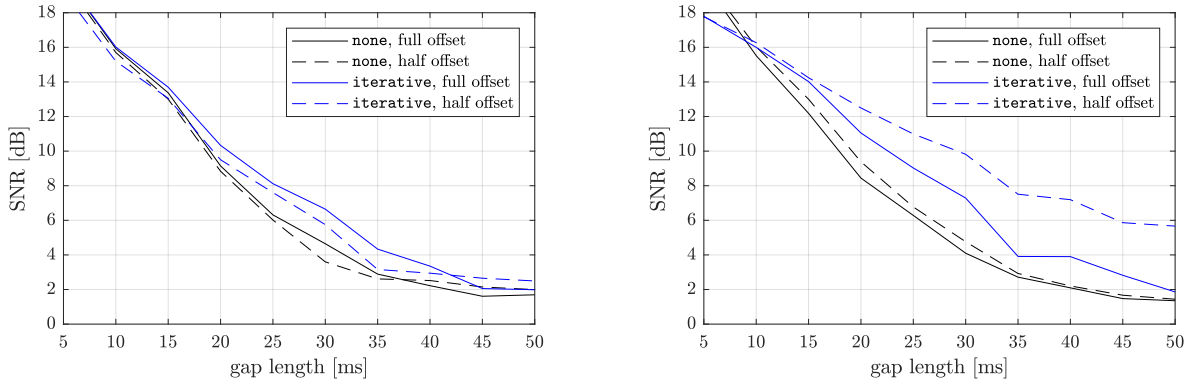
Note also that all the plots suggest that the number of additional gaps does not crucially affect the results.

G. Overall comparison

For the sake of an overall comparison, the time domain compensation for energy loss was applied with `gaps` $= 4$ while the SPAIN and OMP algorithms used the frame-wise DFT dictionary with redundancy 4. Finally, the frame-wise Janssen algorithm was applied according to [11] with the order of the autoregressive model $p = \min(3H + 2, w/3)$, with H denoting the number of missing samples within the current frame (window), and the number of iterations was set to 50.

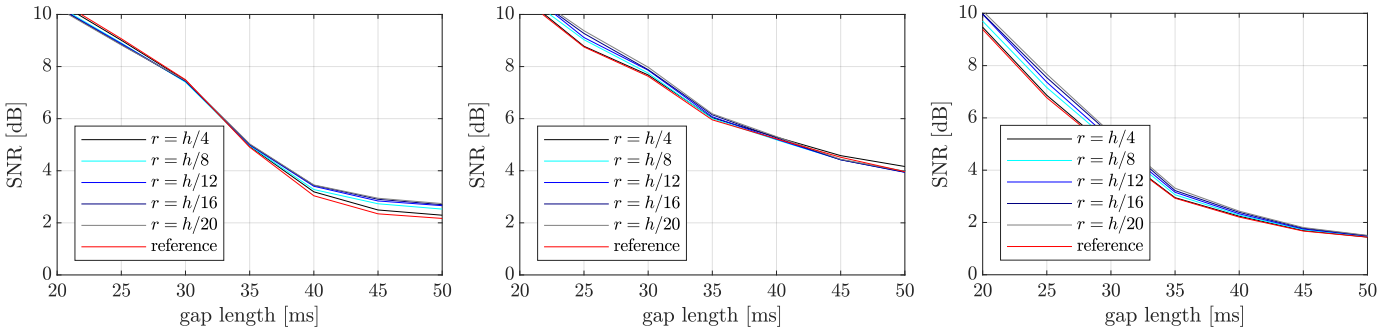
The evaluation is shown in Fig. 13. A comparison based on SNR (Fig. 13a) reveals the success of the described analysis model combining weighting the atoms and the time domain processing (abbreviated as tdc in the figure). It even outperforms the rather high values of SNR of the iterative reweighting approach. Such results suggest that the techniques developed are beneficial for the task of compensating the energy drop. This is apparent for longer gaps in particular, where the low SNR of the simple ℓ_1 method is mainly caused by the disproportion of energy of the original and the restored signal. For gaps of up to 25 ms, on the other hand, the non-convex approaches and the Janssen algorithm remained unsurpassed.

Fig. 13b shows that the values of SNR in our case mostly coincide with the perceptual measure, the ODG. Note, however,



(a) synthesis model (b) analysis model

Fig. 10: Evaluation of the methods for the iterative approach to weighting atoms. In this case, results for both the full and the half offset are shown. The parameters of the reweighting (see Alg. 3 and 4) are $K = 10$, $\varepsilon = 0.001$ and $\delta = 0.01$.



(a) synthesis model (`norm`) (b) analysis model (`energy`) (c) analysis model (`none`)

Fig. 11: Evaluation of the gradual approach. The results are plotted for different choices of the parameter r. The weighting option is chosen based on Figures 11a and 11b. The non-gradual approach with the corresponding weighting is taken as the reference and plotted in red.

the differences regarding the modifications of the ℓ_1 relaxation. First, the difference between the analysis and the synthesis model becomes more pronounced in terms of ODG compared to SNR. Second, the combined compensation methods do not lead to any major improvement in terms of ODG compared to SNR. The interpretation here is that although the time domain compensation is able to provide energy for the inpainted segment, the multiplication in the time domain introduces spectral components that are not present in the reference signal. This effect is then reflected in a lower ODG value.

Finally, a surprising exception is the iterative reweighting, in which case the values of ODG indicate an *opposite* result compared to SNR values, especially in the analysis case. The possible reason is that during the iterations, a coefficient mistakenly taken as significant in the early phase of the algorithm is amplified by the reweighting procedure in the later phases. This leads to (audible) artifacts in the restored signal, which is then reflected by the ODG. However, testing this hypothesis is beyond the scope of the paper.

IX. Software & Reproducible Research

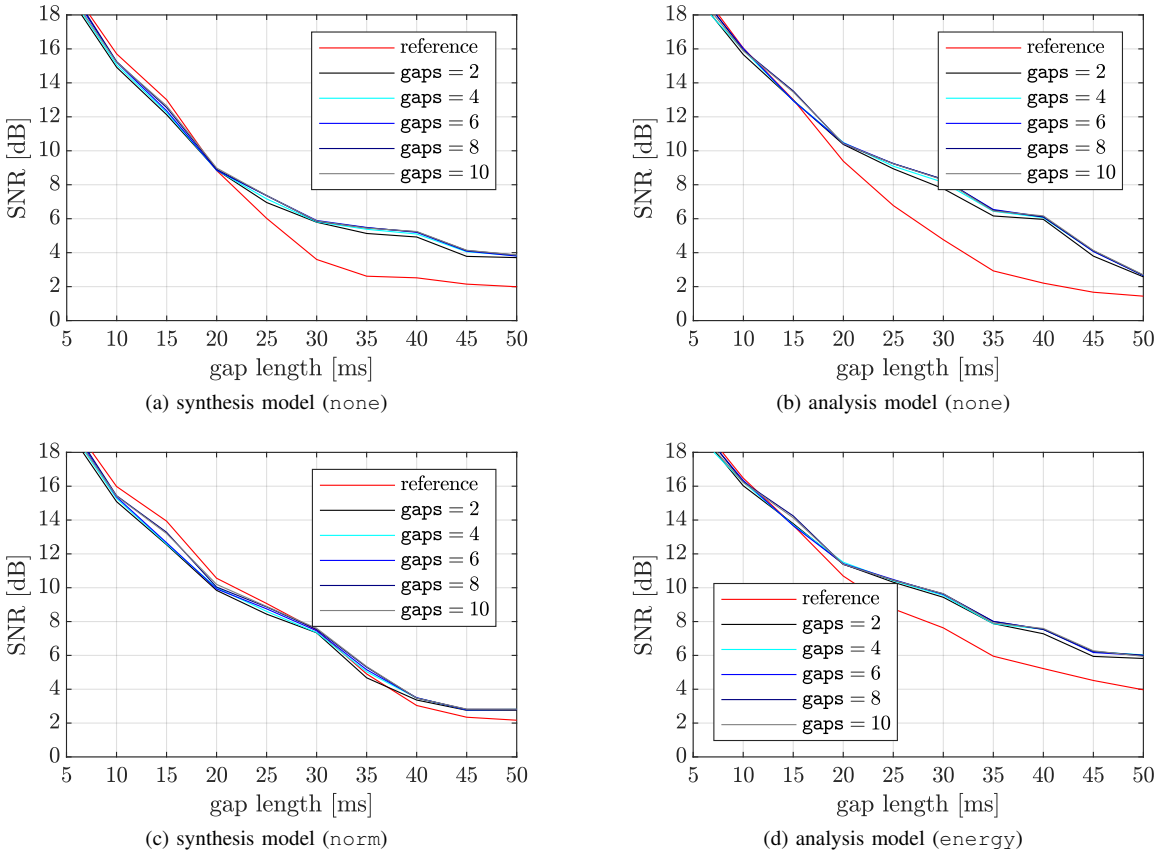
The implementation of the Janssen algorithm was taken from the Audio Inpainting Toolbox [11]. OMP was implemented using the Sparsify Toolbox [55]. The MATLAB codes needed for the experiments and all the data and supplemental figures are available at https://github.com/ondrejmokry/InpaintingRevisited. The codes were run in MATLAB versions 2019a and 2020a.

X. Conclusion

We have described the problem of modern optimization-based methods for audio inpainting, which consists in the lower signal energy in the center of the filled gap. We have presented a number of ideas which can effectively deal with this problem and improve the performance of ℓ_1-based restoration by bringing more energy to the gap.

The sparse analysis model appears to be more stable in performance with respect to altering the settings of the methods. Moreover, the analysis model is superior to the synthesis model in most of the cases presented. Nevertheless, in terms of the SNR, the autoregressive Janssen algorithm can outdo the presented variations in half of the cases, while in terms of ODG, it remains unsurpassed.

In the future, formal listening tests should confirm the numerical results. A modification of the Janssen algorithm based on selected ideas from this paper should also be considered.



(a) synthesis model (`none`)

(b) analysis model (`none`)

(c) synthesis model (`norm`)

(d) analysis model (`energy`)

Fig. 12: Evaluation of the direct time domain compensation for energy loss. The results are plotted for different choices of the number of additional gaps (denoted `gaps`). The reference uses the same weighting of atoms but no additional compensation method. The weighting for the plots 12c and 12d is based on the previous experiments (Fig. 9).

ACKNOWLEDGMENT

The work was supported by the joint project of the FWF and the Czech Science Foundation (GAČR), numbers I 3067-N30 and 17-33798L, respectively, and GAČR project number 20-29009S. Research described in this paper was financed by the National Sustainability Program under grant LO1401. Infrastructure of the SIX Center was used.

REFERENCES

[1] A. J. E. M. Janssen, R. N. J. Veldhuis, and L. B. Vries, "Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes," *IEEE Trans. Acoustics, Speech and Sig. Proc.*, 1986.

[2] L. Oudre, "Interpolation of missing samples in sound signals based on autoregressive modeling," *Image Proc. On Line*, 2018.

[3] W. Etter, "Restoration of a discrete-time signal segment by interpolation based on the left-sided and right-sided autoregressive parameters," *IEEE Trans. Sig. Proc.*, 1996.

[4] P. Esquef et al., "Interpolation of long gaps in audio signals using the warped burg's method," in *Int. Conf. on Digital Audio Effects*, 2003.

[5] I. Kaupinnen and K. Roth, "Audio Signal Extrapolation – Theory and Applications," in *Int. Conf. on Digital Audio Effects*, 2002.

[6] I. Kauppinen and J. Kauppinen, "Reconstruction method for missing or damaged long portions in audio signal," *J. Audio Eng. Soc.*, 2002.

[7] S. J. Godsill and P. J. Rayner, *Digital Audio Restoration: A Statistical Model Based Approach*. Springer, 1998.

[8] G. Chantas et al., "Sparse audio inpainting with variational bayesian inference," in *IEEE Int. Conf. on Consumer Electronics*, 2018.

[9] M. Elad, *Sparse and Redundant Representations: From Theory to Applications in Sig. and Image Proc.*. Springer, 2010.

[10] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization," *Proc. of The National Academy of Sciences*, 2003.

[11] A. Adler et al., "Audio Inpainting," *IEEE Trans. Audio, Speech, and Language Proc.*, 2012.

[12] M. Lagrange, S. Marchand, and J.-B. Rault, "Long interpolation of audio signals using linear prediction in sinusoidal modeling," *J. Audio Eng. Soc.*, 2005.

[13] J. Lindblom and P. Hedelin, "Packet loss concealment based on sinusoidal modeling," in *Speech Coding*. IEEE, 2002.

[14] Y. Bahat, Y. Y. Schechner, and M. Elad, "Self-content-based audio inpainting," *Sig. Proc.*, 2015.

[15] N. Perraudin, N. Holighaus, P. Majdak, and P. Balazs, "Inpainting of long audio segments with similarity graphs," *IEEE/ACM Trans. Audio, Speech, and Language Proc.*, 2018.

[16] A. Marafioti et al., "A context encoder for audio inpainting," *IEEE/ACM Trans. Audio, Speech, and Language Proc.*, 2019.

[17] A. Marafioti et al., "Audio inpainting of music by means of neural networks," in *Audio Engineering Society Convention*, 2019.

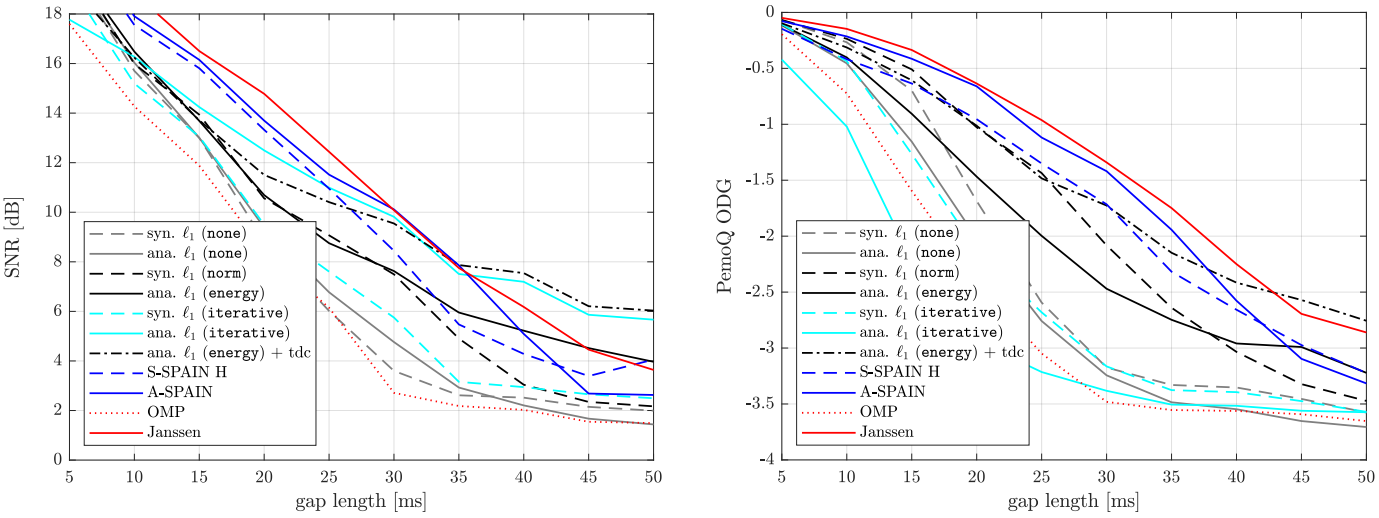
[18] A. Adler et al., "A constrained matching pursuit approach to audio declipping," in *IEEE Int. Conf. on Acoustics, Speech and Sig. Proc.*, 2011.

[19] S. Kitić, N. Bertin, and R. Gribonval, "Sparsity and cosparsity for audio declipping: a flexible non-convex approach," in *Int. Conf. on Latent Variable Analysis and Sig. Separation*, 2015.

[20] Ç. Bilen, A. Ozerov, and P. Pérez, "Audio declipping via nonnegative matrix factorization," in *IEEE Workshop on Applications of Sig. Proc. to Audio and Acoustics*, 2015.

[21] L. Rencker et al., "Consistent dictionary learning for signal declipping," in *Latent Variable Analysis and Sig. Separation*. Springer, 2018.

[22] P. Záviška et al., "Revisiting synthesis model in sparse audio declipper," in *Latent Variable Analysis and Sig. Separation*, 2018.



(a) evaluation by SNR

(b) evaluation by ODG

Fig. 13: Overall comparison of the methods.

[23] P. Záviška et al., "A proper version of synthesis-based sparse audio declipper," in *IEEE Int. Conf. on Acoustics, Speech and Sig. Proc.*, 2019.

[24] F. Lieb and H.-G. Stark, "Audio inpainting: Evaluation of time-frequency representations and structured sparsity approaches," *Sig. Proc.*, 2018.

[25] O. Mokrý and P. Rajmic, "Reweighted l1 minimization for audio inpainting," in *2019 SPARS workshop*, 2019.

[26] K. Gröchenig, *Foundations of time-frequency analysis*. 2001.

[27] O. Christensen, *Frames and Bases, An Introductory Course*. 2008.

[28] ——, *An Introduction to Frames and Riesz Bases*. 2003.

[29] C. Heil, *A Basis Theory Primer: Expanded Edition (Applied and Numerical Harmonic Analysis)*. 2010.

[30] H. G. Feichtinger and T. Strohmer, *Advances in Gabor Analysis*. 2001.

[31] Z. Průša et al., "The Large Time-Frequency Analysis Toolbox 2.0," in *Sound, Music, and Motion*, 2014.

[32] P. L. Søndergaard. LTFAT webpage. URL: http://ltfat.sourceforge.net.

[33] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM Review*, 2001.

[34] R. Tibshirani, "Regression shrinkage and selection via the LASSO," *Journal of the Royal Statistical Society*, 1996.

[35] A. M. Bruckstein et al., "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Review*, 2009.

[36] M. Aharon et al., "K-SVD: An algorithm for designing of overcomplete dictionaries for sparse representations," *IEEE Trans. Sig. Proc.*, 2006.

[37] M. Elad, P. Milanfar, and R. Rubinstein, "Analysis versus synthesis in signal priors," in *Inverse Problems*, 2005.

[38] M. Fornasier, Ed., *Theoretical Foundations and Numerical Methods for Sparse Recovery*. Berlin, Boston: De Gruyter, 2010.

[39] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[40] T. Hastie, R. Tibshirani, and M. Wainwright, *Statistical learning with sparsity*. Boca Raton: CRC Press, 2015.

[41] E. J. Candes et al., "Enhancing sparsity by reweighted ℓ_1 minimization," *Journal of Fourier Analysis and Applications*, 2008.

[42] P. Rajmic, "Exact risk analysis of wavelet spectrum thresholding rules," in *IEEE Int. Conf. on Electronics, Circuits and Systems*, 2003.

[43] M. Daňková and P. Rajmic, "Low-rank model for dynamic MRI: joint solving and debiasing," in *Annual Scientific Meeting*. Springer, 2016.

[44] P. Combettes and J. Pesquet, "Proximal splitting methods in signal processing," *Fixed-Point Algorithms for Inverse Problems in Science and Engineering*, 2011.

[45] A. Chambolle and T. Pock, "A first-order primal-dual algorithm for convex problems with applications to imaging," *Journal of Mathematical Imaging and Vision*, 2011.

[46] N. Komodakis and J. Pesquet, "Playing with duality: An overview of recent primal-dual approaches for solving large-scale optimization problems," *IEEE Sig. Proc. Magazine*, 2015.

[47] L. Condat, "A generic proximal algorithm for convex optimization—application to total variation minimization," *Sig. Proc. Letters*, 2014.

[48] J. J. Moreau, "Proximité et dualité dans un espace hilbertien," *Bulletin de la société mathématique de France*, 1965.

[49] D. Donoho, "De-noising by soft-thresholding," *IEEE Trans. Information Theory*, 1995.

[50] A. J. Weinstein and M. B. Wakin, "Recovering a clipped signal in sparseland," *Sampling Theory in Sig. and Image Proc.*, 2013.

[51] P. Rajmic, H. Bartlová, Z. Průša, and N. Holighaus, "Acceleration of audio inpainting by support restriction," in *Int. Congress on Ultra Modern Telecommunications and Control Systems*, 2015.

[52] R. Huber and B. Kollmeier, "PEMO-Q—A new method for objective audio quality assessment using a model of auditory perception," *IEEE Trans. Audio Speech Language Proc.*, 2006.

[53] EBU SQAM CD: Sound quality assessment material recordings for subjective tests. online. URL: https://tech.ebu.ch/publications/sqamcd.

[54] O. Mokrý et al., "Introducing SPAIN (SParse Audio INpainter)," in *European Sig. Proc. Conf.* IEEE, 2019.

[55] T. Blumensath, "Sparsify toolbox," online, URL: https://www.southampton.ac.uk/engineering/about/staff/tb1m08.page#software.

Ondřej Mokrý did his MSc. in Mathematical Engineering at the Faculty of Mechanical Engineering, BUT. Since 2019 he has been pursuing doctoral studies at the Faculty of Electrical Engineering and Communication at the same university. He focuses on applications of sparse regularization, especially on the problem of audio restoration.

Pavel Rajmic finished his Ph.D. studies in signal processing in 2004 and since then he has been employed at the Faculty of Electrical Engineering and Communication, Brno University of Technology. As a member of the SPLab team, his interests include signal processing, applied and computational mathematics, frame theory and applications, sparse signal modeling, and compressed sensing.