Article

New generalized projection speeds up audio declipping

Pavel Rajmic [1,*,†], Pavel Záviška [1], Vítězslav Veselý [2] and Ondřej Mokrý [1,2]

[1] Signal Processing Laboratory, Brno University of Technology, Brno, Czech Republic
[2] Faculty of Mechanical Engineering, Brno University of Technology, Brno, Czech Republic
* Correspondence: rajmic@vutbr.cz
† Current address: Technická 12, 616 00 Brno, Czech Republic

Abstract: In theory and applications, it is often inevitable to work with projectors onto convex sets, where a linear transform is involved. In this article, a novel projector is presented, which generalizes previous results in that it admits a broader family of linear transforms, but on the other hand it is limited to box-type convex sets in the transformed domain. The new projector has an explicit formula and it can be interpreted within the framework of proximal optimization. The benefit of the new projector is demonstrated on an example from signal processing, where it was possible to speed up the convergence of a signal declipping algorithm by a factor of more than two.

Keywords: projection; optimization; generalization; box constraints; declipping; desaturation; proximal splitting; sparsity

1. Introduction

Proximal algorithms are a modern branch of mathematical optimization, with the principal building blocks being the so-called proximal operators. When proximal optimization is required in applications, the composition of a function with a linear operator occurs often. This necessarily leads to the knowledge of the proximal operator of such a composition. Generally, this requirement leads to iterative solutions, but there are special cases in which the proximal operator can be expressed explicitly.

In the paper, we introduce and prove a new projection lemma. It is inspired by an already known property of proximal operators and, actually, our lemma generalizes the previous result in two directions. On the other hand, the new property holds only for projectors on box-type sets. Projectors represent a special case of proximal operators, and thus our lemma falls into the general concept of proximal minimization; nevertheless, our result is valuable per se, since projections are used in many other contexts and fields of science.

As an example, we apply the novel projection to the problem of signal declipping, i.e. to the restoration of unknown parts of a signal degraded by clipping/saturation. The declipping model will be quite simple, since it is not the goal of this paper to develop a method that outperforms the state of the art, but rather to illustrate the use of the proposed lemma.

In Sec. 2, the proximal and projection operators are reviewed and the state of the art in this specific area is presented. Right next, the novel generalized projection is introduced including the proof. Section 3 then discusses the result and emphasizes its advantages over the current methods, and also points out its limitations. Based on the new result, an example experiment in the signal processing field is performed in Sec. 4, where we show that audio declipping benefits from our approach by the computation being accelerating by a factor of at least two.

2. Materials and Methods

2.1. Proximal algorithms

The proximal algorithm is an iterative algorithm that, under certain conditions, provides a sequence of vectors $\{\mathbf{x}_k\}$ converging to the minimizer of a sum of convex functions $\sum_i f_i(\mathbf{x})$. This is achieved by sequential evaluation of the so-called proximal operators associated with each of the summands, f_i. The most often used and studied are proximal algorithms that can minimize the sum of two functions, such as the Forward-backward algorithm or the Douglas-Rachford algorithm [1,2]; however, there exist algorithms for minimizing the sum of arbitrarily many functions [3].

Algorithms that can minimize a sum involving a linear operator have been presented only recently. Chambolle and Pock [4] presented a primal-dual algorithm that minimizes the sum of two functions $f_1(\mathbf{x}) + f_2(L\mathbf{x})$. The FBB-PD algorithm [5] can cope with the sum of three functions, one of which involves L. The most general algorithm published by Condat [6,7] admits any finite number of functions and any of them can involve a linear operator.

2.2. Proximal operators

Let $\tilde{\mathbb{R}}$ denote the extended real line, i.e. $\tilde{\mathbb{R}} = \mathbb{R} \cup \{-\infty, \infty\}$. Proximal operators are the key components in proximal algorithms. Recall that the proximal operator of a function $h : \mathbb{C}^N \to \tilde{\mathbb{R}}$ maps $\mathbf{z} \in \mathbb{C}^N$ to another vector in $\mathbb{C}^N$, such that [8,9]

$$\operatorname{prox}_h(\mathbf{z}) = \underset{\mathbf{x}\in\mathbb{C}^N}{\arg\min} \left\{ h(\mathbf{x}) + \frac{1}{2}\|\mathbf{z}-\mathbf{x}\|_2^2 \right\}. \tag{1}$$

The proximal operator is a generalization of a projection operator onto a set — this fact is clear if we identify h with the indicator function ι_K of such a set K; in such a case [10],

$$h(x) = \iota_K(x) = \begin{cases} 0 & \text{for } x \in K \\ \infty & \text{for } x \notin K. \end{cases}$$

For a convex h, the map (1) is uniquely defined [3]. A characterization of proximal operators of convex lower semicontinuous functions is the work of J.-J. Moreau [8].

In this article, we will be interested in the proximal operator of the map $f(L\cdot)$, i.e. of such a composition that now $h(\mathbf{x}) = (f \circ L)(\mathbf{x}) = f(L\mathbf{x})$, with $L : \mathbb{C}^N \to \mathbb{C}^M$. If f is convex, then $f \circ L$ is also convex, which is straightforward to show. Therefore, our goal is to find

$$\operatorname{prox}_{f\circ L}(\mathbf{z}) = \underset{\mathbf{x}\in\mathbb{R}^N}{\arg\min} \left\{ f(L\mathbf{x}) + \frac{1}{2}\|\mathbf{z}-\mathbf{x}\|_2^2 \right\}. \tag{2}$$

While explicit formulas exist for many functions h in (1), the situation gets complicated for the compositions in (2).

In many practical situations, one can benefit from the following lemma, occurring, for example, in [3, Tab. 10.1.x]. Let Id denote the identity operator, i.e. $Id(\mathbf{x}) = \mathbf{x}$; the possible subscript indicates the dimensionality of $\mathbf{x}$.

Lemma 1. *Let L be a linear operator, $L : \mathbb{R}^N \to \mathbb{R}^M$, such that $LL^\top = \nu\, Id_M$, $\nu > 0$. Then*

$$\operatorname{prox}_{f\circ L} = Id_N + \nu^{-1} L^\top \circ (\operatorname{prox}_{\nu f} - Id_M) \circ L. \tag{3}$$

This property can be found (possibly including various styles of proof) in [1,9,11,12], but limited to real space setting. The property says that to evaluate the composite proximal operator, it is sufficient to know the proximal operator of f and to be able to perform L and $L^\top$. For an arbitrary $\mathbf{z} \in \mathbb{R}^N$, this means $\mathrm{prox}_{f\circ L}(\mathbf{z}) = \mathbf{z} + \nu^{-1}L^\top(\mathrm{prox}_{\nu f}(L\mathbf{z}) - L\mathbf{z})$. The case $LL^\top = \nu\, Id$ corresponds to L being the so-called tight frame (synthesis) operator known from the field of functional analysis [13–16]. Note that as a consequence, it automatically holds $M < N$. Equation (3) drastically reduces the complexity of computations when prox_f is known. The tight frame acts the same way as an orthonormal operator does.

Example 1. *As an example, the projection onto an ellipsoid* $\{\mathbf{z} \mid \|L\mathbf{z} - \mathbf{y}\|_2 \leq \delta\}$ *has to be computed iteratively in general, however, when L is a tight frame, such a projection can be evaluated explicitly, using a simple projection onto the ball* $\{\mathbf{z} \mid \|\mathbf{z} - \mathbf{y}\|_2 \leq \delta\}$.

Another, more common case of projection is the projection onto an affine subspace, see for example [17]. The motivation for the following lemma is twofold: First, the described projection belongs to the "family" of proximal operators we are considering, and second, this result will be later utilized in the experiment. For the context, this lemma holds for any *complex* $\mathbf{b}$, but on the other hand, its application is limited to affine spaces only.

Lemma 2. *Let L be any linear operator,* $L : \mathbb{C}^N \to \mathbb{C}^M$ *and* $\mathbf{b} \in \mathbb{C}^M$*. Then the projection of a vector* $\mathbf{z} \in \mathbb{C}^N$ *onto the affine space* $\Gamma = \{\mathbf{u} \in \mathbb{C}^N \mid L\mathbf{u} = \mathbf{b}\}$ *can be expressed as*

$$\mathrm{proj}_\Gamma(\mathbf{z}) = \mathbf{z} + L^+(\mathbf{b} - L\mathbf{z})\,. \tag{4}$$

2.3. The new relation of projections

In this section, we present and prove the proposed property of the proximal operator of a composite function. Let * denote the adjoint of a bounded linear operator, for a matrix coinciding with its Hermitian transpose.

First, two concepts known from matrix algebra are adopted. We do this to emphasize that our result could be generalized to an infinite-dimensional setting (although below we work with finite-dimensional spaces, for which a linear operator uniquely coincides with its matrix). Equalities in the infinitedimensional spaces are understood "in norm".

Definition 1. *Let* X, Y *be Hilbert spaces and let* $L : X \to Y$ *be a continuous (bounded) linear operator. The operator L is called full-rank if the range space of L is Y, i.e. if L is surjective.*

Definition 2. *A linear operator D defined on a Hilbert space X with basis* $\{\mathbf{e}_k\}_{k\in\mathbb{N}}$ *is called diagonal if it holds*

$$D\mathbf{e}_k = \lambda_k \mathbf{e}_k \quad \textit{for all } k \in \mathbb{N}, \tag{5}$$

where λ_k *are complex scalars.*[1]

The same way as a diagonal matrix does, a diagonal operator performs entry-wise, since $X \ni \mathbf{x} = \sum_k x_k \mathbf{e}_k$ and thus $D\mathbf{x} = \sum_k x_k D\mathbf{e}_k = \sum_k (\lambda_k x_k)\mathbf{e}_k$. The novel projection utilizes the pseudoinverse operator, which might be recalled now:

Lemma 3. *The linear operator* $L : \mathbb{C}^N \to \mathbb{C}^M$, $M \leq N$, *full-rank,* $\mathbf{y} \in \mathbb{R}^M$*. Then* $L^+\mathbf{y}$ *is the least-norm solution to the system* $L\mathbf{x} = \mathbf{y}$*, i.e.* $\|L^+\mathbf{y}\|_2 \leq \|\mathbf{x}\|_2$*, for any* $\mathbf{x}$ *satisfying* $L\mathbf{x} = \mathbf{y}$*. The pseudoinverse operator* L^+ *is in this full-rank case defined through* $L^+ = L^*(LL^*)^{-1}$.

[1] N. K. Nikolskii, B. S. Pavlov (originator): Diagonal operator. Encyclopedia of Mathematics. URL: http://www.encyclopediaofmath.org/index.php?title=Diagonal_operator

Lemma 4. *Let there be the linear operator $L : \mathbb{C}^N \to \mathbb{C}^M$, $M \leq N$, full-rank, LL^* diagonal. Let the multidimensional interval bounds $\mathbf{b}_1, \mathbf{b}_2 \in \tilde{\mathbb{R}}^M$ such that $\mathbf{b}_1 \leq \mathbf{b}_2$. Then the projection of a vector $\mathbf{z} \in \mathbb{C}^N$,*

$$\mathrm{proj}_\Gamma(\mathbf{z}) = \arg\min_{\mathbf{u}\in\Gamma} \|\mathbf{z}-\mathbf{u}\|_2, \tag{6a}$$

$$\textit{where} \quad \Gamma = \{\mathbf{u} \in \mathbb{C}^N \,|\, \Re(L\mathbf{u}) \in [\mathbf{b}_1, \mathbf{b}_2],\ \Im(L\mathbf{u}) = 0\}, \tag{6b}$$

can be evaluated as

$$\mathrm{proj}_\Gamma(\mathbf{z}) = \mathbf{z} + L^+\left(\mathrm{proj}_{[\mathbf{b}_1,\mathbf{b}_2]}(L\mathbf{z}) - L\mathbf{z}\right), \tag{7}$$

where

$$\mathrm{proj}_{[\mathbf{b}_1,\mathbf{b}_2]}(\mathbf{y}) = \min(\max(\mathbf{b}_1, \Re(\mathbf{y})), \mathbf{b}_2), \quad \mathbf{y} \in \mathbb{C}^M. \tag{8}$$

Here, Eq. (8) is the projection of the complex vector onto a real multidimensional interval $[\mathbf{b}_1, \mathbf{b}_2]$ with min *and* max *functions returning pairwise extremes entry-by-entry.*

The proposed lemma says that the projection onto the box-type set Γ that involves L can be made simpler and faster using projection onto the plain $[\mathbf{b}_1, \mathbf{b}_2]$. In (7), the application of L^+ amounts to entrywise multiplication by the inverse diagonal of LL^*, followed by L^*; therefore, the cost of L^+ will typically be in the order of the cost of L^*.

Before we prove Lemma 4, we present several lemmas that will be needed later.

Lemma 5. *Let $L : \mathbb{C}^N \to \mathbb{C}^M$, $M \leq N$, full-rank, LL^* diagonal. Then $(L^+)^*L^+$ is diagonal.*

Proof. The operator L is full-rank and therefore its pseudoinverse can be computed as in Lemma 3. Then

$$(L^+)^*L^+ = \left((LL^*)^{-1}\right)^* LL^*\,(LL^*)^{-1} = \left((LL^*)^{-1}\right)^* = (LL^*)^{-1}, \tag{9}$$

which is the inverse of the diagonal operator LL^*, and thus it is diagonal as well. □

Lemma 6. *Given any linear operator $L : \mathbb{C}^N \to \mathbb{C}^M$ and vector $\mathbf{y} \in \mathbb{C}^N$, the vector $L^+L\mathbf{y} \in \mathbb{C}^N$ is the shortest vector that is mapped by L to $L\mathbf{y}$.*

Proof. Consider the linear system $L\mathbf{x} = L\mathbf{y}$ with $\mathbf{y}$ fixed. Using the properties of the pseudoinverse (Lemma 3), the shortest solution to such a system is found by pseudoinverting the right hand side, i.e. by $L^+(L\mathbf{y})$. □

Lemma 7. *Let $W : \mathbb{C}^M \to \mathbb{C}^N$ be such an operator that W^*W is diagonal and let $\mathbf{x}, \mathbf{y} \in \mathbb{C}^M$. Then from $|x_m| \leq |y_m|, \forall m$, it follows that $\|W\mathbf{x}\|_2 \leq \|W\mathbf{y}\|_2$.*

Proof. Let $\{\mathbf{e}_n\}$ be an orthonormal basis of $\mathbb{C}^N$. Then, since W^*W is a diagonal operator, according to Definition 2 we have $W^*W\mathbf{e}_m = \lambda_m \mathbf{e}_m$ for a set of scalars $\{\lambda_m\}$, which in this case are real nonnegative thanks to the positive semidefiniteness of W^*W. Now $\|W\mathbf{x}\|_2^2 = \langle W\mathbf{x}, W\mathbf{x}\rangle = \langle W^*W\mathbf{x}, \mathbf{x}\rangle$ using the basic property of the adjoint operator. Thus for $\mathbf{x} = \sum_m x_m \mathbf{e}_m$, it holds

$$\|W\mathbf{x}\|_2^2 = \langle W^*W\sum_m x_m\mathbf{e}_m, \sum_m x_m\mathbf{e}_m\rangle = \langle \sum_m x_m W^*W\mathbf{e}_m, \sum_m x_m\mathbf{e}_m\rangle = \langle \sum_m x_m\lambda_m\mathbf{e}_m, \sum_m x_m\mathbf{e}_m\rangle = \sum_m \lambda_m |x_m|^2,$$

where in the last equality we have used the linearity of inner product and the orthonormality of $\{\mathbf{e}_m\}$. If $|x_m| \leq |y_m|$, then also their squares satisfy $|x_m|^2 \leq |y_m|^2$ for each m and therefore $\|W\mathbf{x}\|_2^2 = \sum_m \lambda_m |x_m|^2 \leq \sum_m \lambda_m |y_m|^2 = \|W\mathbf{y}\|_2^2$, since all λ_m are nonnegative. □

Proof of Lemma 4. First of all, we will show that (8) is indeed a projector onto $[\mathbf{b}_1, \mathbf{b}_2]$. Due to the fact that the set $[\mathbf{b}_1, \mathbf{b}_2]$ is a multidimensional box (although possibly open towards plus or

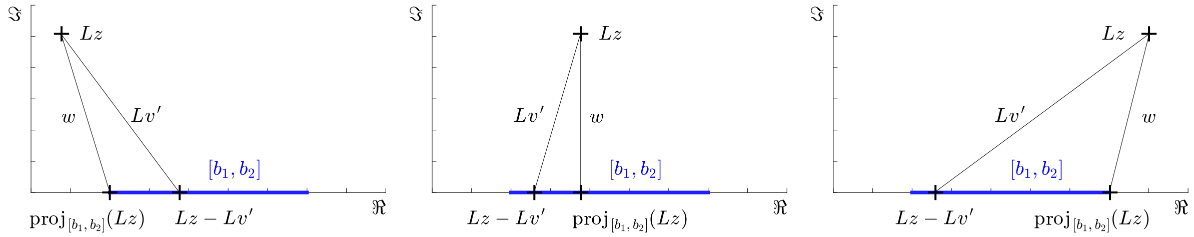

Figure 1. Illustration of the inequality (14) for different cases of the relative position of $L\mathbf{z}$ and the interval $[\mathbf{b}_1, \mathbf{b}_2]$. Only a single entry is depicted for each vector, i.e. the meaning of Lz in the plot is $(L\mathbf{z})_m$ and similarly for the other vectors. The point $L(\mathbf{z} - \mathbf{v}')$ represents an arbitrary point in the interval $[\mathbf{b}_1, \mathbf{b}_2]$, as assumed.

minus infinity), the projection $\text{proj}_{[\mathbf{b}_1,\mathbf{b}_2]}(\mathbf{y})$ can be evaluated entry-by-entry. Consider the m-th entry $\mathbf{y}_m =: y$, $(\mathbf{b}_1)_m =: b_1$, $(\mathbf{b}_2)_m =: b_2$. In accordance with the definition of projection,

$$\begin{aligned}\text{proj}_{[b_1,b_2]}(y) &= \underset{b_1 \leq x \leq b_2}{\arg\min}\, |x - y| = \underset{b_1 \leq x \leq b_2}{\arg\min} \sqrt{(x - \Re(y))^2 + \Im(y)^2} \\ &= \underset{b_1 \leq x \leq b_2}{\arg\min} \left((x - \Re(y))^2 + \Im(y)^2\right) = \underset{b_1 \leq x \leq b_2}{\arg\min}\, |x - \Re(y)|, \end{aligned} \tag{10}$$

i.e. the original task reduces to projecting $\Re(y)$ onto $[b_1, b_2]$. We prove (8) by evaluating the only three possibilities (for a single entry but applied to all of them):

$$\text{proj}_{[b_1,b_2]}(y) = \min(\max(b_1, \Re(y)), b_2) = \begin{cases} b_1 & \text{for } \Re(y) < b_1, \\ b_2 & \text{for } \Re(y) > b_2, \\ y & \text{for } b_1 \leq \Re(y) \leq b_2. \end{cases} \tag{11}$$

Now return to formula (7). Suppose that $\mathbf{z} \in \Gamma$; then $\Re(L\mathbf{z}) \in [\mathbf{b}_1, \mathbf{b}_2]$ and from (8) we have $\text{proj}_{[\mathbf{b}_1,\mathbf{b}_2]}(L\mathbf{z}) = L\mathbf{z}$ and from (7) we get $\text{proj}_\Gamma(\mathbf{z}) = \mathbf{z}$, which is correct. In the complementary case, $\mathbf{z} \notin \Gamma$, due to the fullrank of L, there exists at least one $\mathbf{v} \in \mathbb{C}^N$ for which

$$L\mathbf{v} = L\mathbf{z} - \text{proj}_{[\mathbf{b}_1,\mathbf{b}_2]}(L\mathbf{z}) =: \mathbf{w}. \tag{12}$$

Trivially, any such vector $\mathbf{v}$ satisfies $L(\mathbf{z} - \mathbf{v}) = \text{proj}_{[\mathbf{b}_1,\mathbf{b}_2]}(L\mathbf{z}) \in [\mathbf{b}_1, \mathbf{b}_2]$. In line with (6a), we are looking for the shortest $\mathbf{v}$, which is found using the pseudoinverse:

$$\mathbf{v}^+ := L^+\mathbf{w}. \tag{13}$$

Suppose now that there is another vector $\mathbf{v}' \in \mathbb{C}^N$ such that $L(\mathbf{z} - \mathbf{v}') \in [\mathbf{b}_1, \mathbf{b}_2]$. We will show that $\|\mathbf{v}'\|_2 \geq \|\mathbf{v}^+\|_2$. The projection (8) is performed entrywise and thus it must hold

$$|w_m| \leq |(L\mathbf{v}')_m| \quad \text{for } m = 1, \ldots, M, \tag{14}$$

see Fig. 1 for illustration. From Lemma 5 it follows that $(L^+)^* L^+$ is diagonal. Therefore, using Lemma 7 we have

$$\|\mathbf{v}^+\|_2 = \|L^+\mathbf{w}\|_2 \leq \|L^+(L\mathbf{v}')\|_2. \tag{15}$$

Finally, Lemma 6 states that $\|L^+(L\mathbf{v}')\|_2 \leq \|\mathbf{v}'\|_2$, which together with Eq. (15) concludes the proof. □

3. Discussion on the new result

Our Lemma 4 allows us to cover a broader scope of operators than Lemma 1 does. Specifically, L can be *an arbitrary complex* operator, and, moreover, LL^* can be diagonal, not just a multiple of the identity. On the other hand, the generalized relation holds only for an projector onto a box-type set and not for a generic proximal operator.

Example 1 showed that projecting onto an ellipsoid could be performed fast in the special case of $LL^* = \nu\, Id$. In the case of the diagonal LL^*, this is no longer possible.

Remark 1. *Lemma 4 clearly holds when L is the synthesis operator of a tight frame (in such a case, $LL^* = \nu\, Id$ is a multiple of the identity). The property is still valid when such a synthesis operator is weighted column-by-column by a diagonal operator D that is real and positive. Substituting DL for L still meets the assumptions of the theorem, since $(DL)(DL)^* = DLL^*D^* = \nu DD^*$, which is again diagonal.*

Remark 2. *On the contrary, consider that L is an analysis operator. First of all, LL^* cannot be diagonal (while actually LL^* is diagonal); even if such a condition in the assumption is disregarded, the theorem still cannot be formulated for such L. The problem appears early in the proof, in (12). This linear system does not need to have any solution $\mathbf{v}$ for fixed $\mathbf{z}$, and therefore by using (8), we do not obtain a vector in Γ.*

Remark 3. *It is clear that if we choose L full-rank, but not obeying that LL^* is diagonal, the proof is ruined in Eq. (15), since Lemmas 7 and 5 cannot be used. In such a situation, formula (7) clearly provides a feasible solution to the problem (6), but it is not guaranteed to be the optimal one.*

Remark 4. *Consider the common discrete Fourier transform (DFT). The related synthesis and analysis operators, L and L^*, respectively, are complex; however, recall that they have a special complex-conjugate structure, such that when $\mathbf{x}$ is a real signal, then $LL^*\mathbf{x} = \mathbf{x}$ is real again, while $L^*\mathbf{x}$ bears the same complex-conjugate structure. In such a case, when the coefficient vector $\mathbf{z}$ is fed into (7), $L\mathbf{z}$ is real and so is its projection $\mathrm{proj}_{[\mathbf{b}_1,\mathbf{b}_2]}(L\mathbf{z})$. Thus, L^+ is applied to a real vector. Since $LL^* = Id$ for the DFT, it follows that the pseudoinverse $L^+ = L^* = L^{-1}$ bears the structure of L^*. As a consequence, the resulting projection is also complex-conjugate. Therefore, in theory, taking the real part in (8) is redundant for the DFT. It will be shown later that this feature will be kept for time-frequency operators that will be used in the experiment.*

4. Experiment

The so-called *clipping* is an artefact occurring in signal acquisition or processing, when the dynamic range of a signal is larger than the representation range of a device which is used to record or process the signal. In audio processing, this typically happens when an analog signal is captured with an analog-to-digital converter whose pre-amplifier is set such that the largest signal values are beyond the maximum digital representation level. Clipping is a psychoacoustically annoying effect, and it introduces a plenty of unwanted higher harmonic components to the signal.

The process of restoring as much audio information as possible, based on the degraded signal, is usually called *declipping*. Such a restoration task is clearly ill-conditioned, since there are infinitely many possible solutions to the problem. Declipping methods thus have to rely on additional information that characterizes the signal.

Besides other types of signal modeling [18–22], sparse priors have recently become very popular. Such a type of regularization assumes that the signal can be well approximated by only a few synthesis coefficients. As the transforms, the cosine (DCT, MDCT) [23,24] or the short-time Fourier transform (STFT) [25,26] are typically used. Since finding truly sparse solutions is NP-hard, various strategies are used to approximate the solution. Two major groups of algorithms can be identified: the greedy algorithms [27–31] and the convex-relaxation algorithms [25,26,32], or algorithms combining both approaches [24].

In our experiment, a convex formulation of the declipping problem is utilized with (synthesis) sparsity in the STFT (also termed Gabor) domain. It is shown that when the Gabor frame fulfills the requirements of Lemma 4, as a consequence the projection onto the time-domain constraints can be done in a single step, leading to the use of a simple algorithm and speeding up the convergence by a factor of 2–3. We show the benefit of our novel projection on a simple declipping model, being aware that more sophisticated models cited above lead to a better quality of reconstruction.

4.1. Problem formulation

It is assumed that a discrete-time signal $\mathbf{y}_\mathrm{c} \in \mathbb{R}^M$ is given that has been hard-clipped, i.e. any original time-domain sample above the clipping level θ_H has been substituted by θ_H and analogously for the samples below the level θ_L. The rest of the samples in $\mathbf{y}_\mathrm{c}$ have not been altered by clipping, and they will be called reliable.

It is furthermore assumed that the unknown original signal $\mathbf{y}$ is sparse or compressible in the SFTF domain, formally written as $\mathbf{y} \approx G\mathbf{c}$ with $\|\mathbf{c}\|_0 \ll N$, where G is the linear Gabor synthesis operator [33], and where the pseudo-norm $\|\mathbf{c}\|_0$ counts the non-zero elements of $\mathbf{c}$. This is a natural model, since it is known that human perception is based on a kind of time-frequency analysis [13,16]. In this paper we utilize the ℓ_1-norm as a convex surrogate of the sparsity pseudo-norm [34] and we formulate the optimization problem as

$$\arg\min_{\mathbf{c}} \|\mathbf{c}\|_1 \quad \text{subject to} \quad \begin{cases} M_\mathrm{R}G\mathbf{c} = M_\mathrm{R}\mathbf{y}_\mathrm{c} & \text{(reliable samples)} \\ M_\mathrm{H}G\mathbf{c} \geq \theta_\mathrm{H} & \text{(samples clipped from above)} \\ M_\mathrm{L}G\mathbf{c} \leq \theta_\mathrm{L} & \text{(samples clipped from below),} \end{cases} \tag{16}$$

where $M_\mathrm{R}, M_\mathrm{H}, M_\mathrm{L}$ are simple projection operators (masks) selecting only samples related to the reliable and the clipped parts, respectively, from the complete set of indexes $\{1, \ldots, M\}$. Formulation (16) thus looks for coefficients that are approximately sparse and at the same time generate a signal that is consistent with the time-domain constraints.

4.2. The Gabor operators

As a prelude to Gabor operators, return to the ordinary DFT. The DFT analysis, $F^* : \mathbb{C}^M \to \mathbb{C}^Q$, computes scalar products of the signal with complex exponentials

$$\frac{1}{\sqrt{Q}} \left[\exp\left(\mathrm{i}2\pi 0 \frac{q}{Q}\right), \exp\left(\mathrm{i}2\pi 1 \frac{q}{Q}\right), \ldots, \exp\left(\mathrm{i}2\pi (M-1) \frac{q}{Q}\right) \right]^\top \tag{17}$$

for each frequency $q = 0, \ldots, Q-1$. Following Remark 4, we now recall that the DFT operator takes the real signal into the complex domain of dimension Q such that the image has a complex-conjugated structure. To make this statement precise, $F^* : \mathbb{R}^M \to \mathbb{K}^Q$, where we define

$$\mathbb{K}^Q = \left\{ \mathbf{c} \in \mathbb{C}^Q \,|\, c_0, c_{\frac{Q}{2}} \in \mathbb{R};\, c_q = \overline{c_{Q-q}},\ q = 1, \ldots, \frac{Q}{2} - 1 \right\}. \tag{18}$$

Such a structure follows immediately from (17). In the common DFT case, $Q = M$, which implies that (17) is an orthonormal basis of $\mathbb{C}^M$, and thus $F^* = F^{-1}$. Nevertheless the relation between coefficients holds true also in the case $Q > M$ (with Q even) in (17), which is referred to as the *overcompleteness in frequency* and sometimes the *redundant DFT*. In such a situation, the complex exponentials form a Parseval tight frame of $\mathbb{C}^M$ [33]. Both cases can be computed using the FFT.

The Gabor transform is a time-frequency transform with an extra structure compared to the DFT. Gabor analysis G^* of a signal consists of windowing the time-domain signal and then taking scalar products of such windowed signal with complex harmonics (17); effectively, G^* amounts in computing the FFT of the windowed signal. Generally, $G^* : \mathbb{C}^M \to \mathbb{C}^N$, $N > M$ and the other way round for the synthesis G. The window moves along the signal.

Conditions are known which guarantee that the frame operator GG^* is invertible. Under additional conditions, the related synthesis operator G moreover bears the same time-frequency structure as G^* does [13]. From the practical point of view, systems for which $GG^* = \nu\, Id_M$ are preferably used in applications, see for example [16,29,35–39], however let us note that the assumptions of our lemma, in addition, cover an important class of the so-called *painless* Gabor transforms [40], for which the operator GG^* is diagonal and not constant. Gabor Parseval tight frames thus fall into our setup as a special case.

For the Gabor transform, parameters have to be specified such as the window shape and length, time-shift a of the window, and the number of frequency channels Q [13]. The number of time samples M must be a multiple of a [41]; then the analysis operator is $G^* : \mathbb{R}^M \to \mathbb{C}^{MQ/a}$. Thanks to the above-described structure of the DFT analysis, it is clear that we can even understand the operator as $G^* : \mathbb{R}^M \to \mathbb{K}^Q \times \ldots \times \mathbb{K}^Q$, where each of the vectors from $\mathbb{K}^Q$ originates in one of the M/a possible window positions. The synthesis $G : \mathbb{K}^Q \times \ldots \times \mathbb{K}^Q \to \mathbb{R}^M$ produces a real signal again. It will be clear later that computations in the algorithms do not tackle the coefficients in such a way that they would fall out of the space $\mathbb{K}^Q \times \ldots \times \mathbb{K}^Q$, making it possible to omit the real-part operator in (8).

4.3. Problem solution

We will use the so-called proximal splitting algorithms to solve (16). Proximal splitting algorithms were introduced in Sec. 2. First of all, it will be convenient to rewrite (16) to an unconstrained form using indicator functions (defined in Sec. 2) as

$$\operatorname*{arg\,min}_{\mathbf{c}\in\mathbb{C}^N} \|\mathbf{c}\|_1 + \iota_R(\mathbf{c}) + \iota_H(\mathbf{c}) + \iota_L(\mathbf{c}) \tag{19}$$

where the sets R, H, L coincide with the three conditions in (16), i.e. they are

$$R = \{\mathbf{c} \,|\, M_\mathrm{R} G\mathbf{c} = M_\mathrm{R}\mathbf{y}_\mathrm{c}\}, \quad H = \{\mathbf{c} \,|\, M_\mathrm{H} G\mathbf{c} \geq \theta_\mathrm{H}\}, \quad L = \{\mathbf{c} \,|\, M_\mathrm{L} G\mathbf{c} \leq \theta_\mathrm{L}\}. \tag{20}$$

Once optimal coefficients are found, signal recovery is simply obtained by their synthesis through G.

Eq. (19) presents a sum of convex functions, considering the fact that the sets R, H, L are convex. Therefore, (19) is suitable to be solved by a proximal algorithm.

Recall that proximal algorithms rely on evaluating the so-called proximal operators linked to the functions present in the sum. It is known that the proximal operator of $\|\cdot\|_1$ is the soft thresholding [42–44] and that the proximal operator of ι_K is the projection onto the convex set K [3,10]. Performing soft thresholding is a simple operation, but regarding projections onto the sets R, H, L, it becomes more complicated. Explicit formulas for projecting onto these sets are known only for G being an unitary operator or a tight frame ($GG^* = \nu \, Id_M$) defined in real domain, see Lemma 1. Our G, however, maps $\mathbb{C}^N \to \mathbb{R}^M$ and moreover there are practical Gabor systems for which GG^* is only diagonal [13,40].

In this experiment, we compare two approaches to overcome this complication. One is to adapt the general Condat algorithm (CA) which allows us to simplify the projections at the cost of an additional application of G and G^* in each iteration. In the second approach, we utilize the novel projection lemma and show its superiority over the first choice.

4.4. Condat algorithm

Problem (19) is equivalent to

$$\operatorname*{arg\,min}_{\mathbf{c}} \|\mathbf{c}\|_1 + \iota_R(\mathbf{c}) + \iota_{H'}(G\mathbf{c}) + \iota_{L'}(G\mathbf{c}) \tag{21}$$

where $H' = \{\mathbf{z} \,|\, M_\mathrm{H}\mathbf{z} \geq \theta_\mathrm{H}\}$ and $L' = \{\mathbf{z} \,|\, M_\mathrm{L}\mathbf{z} \leq \theta_\mathrm{L}\}$. Note that projection onto such sets is trivial compared to projections onto H and L. Regarding the reliable part of the signal, we could also resort to replacing $\iota_R(\mathbf{c})$ by $\iota_{R'}(G\mathbf{c})$, where projecting onto $R' = \{\mathbf{z} \,|\, M_\mathrm{R}\mathbf{z} = M_\mathrm{R}\mathbf{y}_\mathrm{c}\}$ amounts to simple substitution of $\mathbf{z}$ with $\mathbf{y}_\mathrm{c}$ at the reliable positions. Note, however, that R is an affine set and there exists an explicit formula for the respective projection, see Lemma 2. In our setup, this lemma renders the following corollary.

Algorithm 1: Condat algorithm adapted to solving (21)

```
Input: Set starting points $\mathbf{c}^{(0)}$, $\mathbf{u}_\mathrm{R}^{(0)}$, $\mathbf{u}_\mathrm{H}^{(0)}$, $\mathbf{u}_\mathrm{L}^{(0)}$.
Set parameters $\rho \in (0;2)$, $\sigma, \tau > 0$.
for $i = 0, 1, \ldots$ do
    Combine coefficients and sparsify them:
    $\tilde{\mathbf{c}}^{(i+1)} = \mathrm{soft}_\tau \left( \mathbf{c}^{(i)} - \tau \left( \mathbf{u}_\mathrm{R}^{(i)} + G^* \mathbf{u}_\mathrm{H}^{(i)} + G^* \mathbf{u}_\mathrm{L}^{(i)} \right) \right)$
    $\mathbf{c}^{(i+1)} = \rho\, \tilde{\mathbf{c}}^{(i+1)} + (1-\rho)\mathbf{c}^{(i)}$
    Project reliable:
    $\mathbf{v}_\mathrm{R} = \mathbf{u}_\mathrm{R}^{(i)} + \sigma \left( 2\tilde{\mathbf{c}}^{(i+1)} - \mathbf{c}^{(i)} \right)$      % auxiliary variable
    $\tilde{\mathbf{u}}_\mathrm{R}^{(i+1)} = \mathbf{v}_\mathrm{R} - \sigma\, \mathrm{proj}_R\, (\mathbf{v}_\mathrm{R}/\sigma)$      % using (22)
    $\mathbf{u}_\mathrm{R}^{(i+1)} = \rho\, \tilde{\mathbf{u}}_\mathrm{R}^{(i+1)} + (1-\rho)\mathbf{u}_\mathrm{R}^{(i)}$
    Project clipped from above:
    $\mathbf{v}_\mathrm{H} = \mathbf{u}_\mathrm{H}^{(i)} + \sigma\, G \left( 2\tilde{\mathbf{c}}^{(i+1)} - \mathbf{c}^{(i)} \right)$      % auxiliary variable
    $\tilde{\mathbf{u}}_\mathrm{H}^{(i+1)} = \mathbf{v}_\mathrm{H} - \sigma\, \mathrm{proj}_{H'}\, (\mathbf{v}_\mathrm{H}/\sigma)$      % elementwise projection
    $\mathbf{u}_\mathrm{H}^{(i+1)} = \rho\, \tilde{\mathbf{u}}_\mathrm{H}^{(i+1)} + (1-\rho)\mathbf{u}_\mathrm{H}^{(i)}$
    Project clipped from below:
    $\mathbf{v}_\mathrm{L} = \mathbf{u}_\mathrm{L}^{(i)} + \sigma\, G \left( 2\tilde{\mathbf{c}}^{(i+1)} - \mathbf{c}^{(i)} \right)$      % auxiliary variable
    $\tilde{\mathbf{u}}_\mathrm{L}^{(i+1)} = \mathbf{v}_\mathrm{L} - \sigma\, \mathrm{proj}_{L'}\, (\mathbf{v}_\mathrm{L}/\sigma)$      % elementwise projection
    $\mathbf{u}_\mathrm{L}^{(i+1)} = \rho\, \tilde{\mathbf{u}}_\mathrm{L}^{(i+1)} + (1-\rho)\mathbf{u}_\mathrm{L}^{(i)}$
return $\mathbf{c}^{(i+1)}$
```

Corollary 1. *Let G be a linear synthesis operator, $G : \mathbb{C}^N \to \mathbb{C}^M$ such that GG^* is diagonal. Let M_R be the "reliable" mask operator. Then for the projection onto R we have*

$$\mathrm{proj}_R(\mathbf{z}) = \mathbf{z} + G^* M_R^\top (M_R G G^* M_R^\top)^{-1} (M_R \mathbf{y}_\mathrm{c} - M_R G \mathbf{z}). \tag{22}$$

Proof. We use Lemma 2 in the particular case of the diagonal product. The role of L is played by $M_\mathrm{R} G$. The only step to comment on is the determination of the pseudoinverse operator. Recall that G is assumed full-rank; therefore, $M_\mathrm{R} G$ spans the subspace onto which it projects, since M_R is applied elementwise. Therefore we can use the formula mentioned in Lemma 3 and obtain

$$(M_\mathrm{R} G)^+ = (M_\mathrm{R} G)^* (M_\mathrm{R} G G^* M_\mathrm{R}^*)^{-1} = G^* M_\mathrm{R}^\top (M_\mathrm{R} G G^* M_\mathrm{R}^\top)^{-1}. \tag{23}$$

Formula (22) follows immediately. □

Formula (22) employs application of one synthesis and one analysis. Note that $(M_\mathrm{R} G G^* M_\mathrm{R}^\top)$ is a square diagonal matrix whose size equals the number of reliable samples, and as such, its inversion is simple and can be precomputed.

The complete Condat algorithm for declipping is described in Alg. 1. The convergence is guaranteed when the step sizes τ and σ satisfy $\tau\sigma \| Id + G^* G + G^* G \| \leq 1$, where $\| \cdot \|$ is the operator norm [6]. With the help of the triangle inequality we find a (suboptimal) bound for the step sizes. Since GG^* is diagonal and it holds $\|L^* L\| = \|L L^*\|$ for any operator, we have $\|G^* G\| = \max \mathrm{diag} |GG^*| =: \mu$. Thus, $\tau\sigma \leq 1/(1+2\mu)$ ensures convergence; for Parseval frames, it simplifies to $\tau\sigma \leq 1/3$. Still, the choice of particular σ and τ influences the convergence speed.

In practice, we terminate the main loop when a proper convergence criterion is fulfilled (will be discussed later). Usually, when the algorithm stops, the solution might happen to lie slightly out of the

Algorithm 2: Douglas-Rachford algorithm solving (24)

```
Input: Set starting point $\mathbf{c}^{(0)}$.
Set parameters $\lambda = 1, \gamma > 0$.
for $i = 0, 1, \ldots$ do
    $\tilde{\mathbf{c}}^{(i)} = \mathrm{proj}_K\, \mathbf{c}^{(i)}$    % using (26) and (27)
    $\mathbf{c}^{(i+1)} = \mathbf{c}^{(i)} + \lambda \left( \mathrm{soft}_\gamma(2\tilde{\mathbf{c}}^{(i)} - \mathbf{c}^{(i)}) - \tilde{\mathbf{c}}^{(i)} \right)$
return $\mathbf{c}^{(i+1)}$
```

feasible set. Therefore, we normally perform a final projection of the current solution onto the feasible set. (This applies also to the second algorithm below.)

The advantage of the Condat algorithm is its universality; it would work for audio declipping with arbitrary (meaningful) linear operator G. In the following, however, we exploit the special properties of G which are assumed.

4.5. Douglas-Rachford algorithm

Now we approach the problem (19) directly. Notice that it could be rewritten as a sum of two convex functions

$$\operatorname*{arg\,min}_{\mathbf{c}} \|\mathbf{c}\|_1 + \iota_K(\mathbf{c}) \tag{24}$$

with $K = \{\mathbf{z} \,|\, M_\mathrm{R} G\mathbf{z} = M_\mathrm{R}\mathbf{y}_\mathrm{c},\ M_\mathrm{H} G\mathbf{z} \geq \theta_\mathrm{H},\ M_\mathrm{L} G\mathbf{z} \leq \theta_\mathrm{L}\}$. Due to such a form, the simple Douglas-Rachford (DR) algorithm [1,3] can be applied to solve (24), if the projector onto K is available. We use our new lemma to develop such a projector.

First, define the "lower" and "upper" bounding vectors $\mathbf{b}_\mathrm{L}, \mathbf{b}_\mathrm{H} \in \tilde{\mathbb{R}}^M$ such that for $m = 1, \ldots, M$,

$$(\mathbf{b}_\mathrm{L})_m = \begin{cases} (\mathbf{y}_\mathrm{c})_m & \text{for } \theta_\mathrm{L} < (\mathbf{y}_\mathrm{c})_m < \theta_\mathrm{H}, \\ \theta_\mathrm{H} & \text{for } (\mathbf{y}_\mathrm{c})_m = \theta_\mathrm{H}, \\ -\infty & \text{for } (\mathbf{y}_\mathrm{c})_m = \theta_\mathrm{L}, \end{cases} \qquad (\mathbf{b}_\mathrm{H})_m = \begin{cases} (\mathbf{y}_\mathrm{c})_m & \text{for } \theta_\mathrm{L} < (\mathbf{y}_\mathrm{c})_m < \theta_\mathrm{H}, \\ \infty & \text{for } (\mathbf{y}_\mathrm{c})_m = \theta_\mathrm{H}, \\ \theta_\mathrm{L} & \text{for } (\mathbf{y}_\mathrm{c})_m = \theta_\mathrm{L}. \end{cases} \tag{25}$$

Note that defining the multidimensional box $[\mathbf{b}_\mathrm{L}, \mathbf{b}_\mathrm{H}]$ this way matches the set of feasible solutions K. Specifically, we have $K = \{\mathbf{c} \,|\, \mathbf{b}_\mathrm{L} \leq G\mathbf{c} \leq \mathbf{b}_\mathrm{H}\}$. Due to Lemma 4, the projection onto K is realized via

$$\mathrm{proj}_K(\mathbf{c}) = \mathbf{c} + G^+ \left(\mathrm{proj}_{[\mathbf{b}_\mathrm{L}, \mathbf{b}_\mathrm{H}]}(G\mathbf{c}) - G\mathbf{c} \right), \tag{26}$$

where

$$\mathrm{proj}_{[\mathbf{b}_\mathrm{L}, \mathbf{b}_\mathrm{H}]}(G\mathbf{c}) = \min(\max(\mathbf{b}_\mathrm{L}, \Re(G\mathbf{c})), \mathbf{b}_\mathrm{H}). \tag{27}$$

The Douglas-Rachford algorithm is presented in Alg. 2. It always converges, nevertheless the parameter γ is responsible for the convergence speed.

4.6. Comparison of the algorithms

If converged, both the presented algorithms produce the same solution when the objective function is strictly convex. In our case, (19) is not strictly convex, and therefore we cannot hope for the same solution. The objective function, i.e. the ℓ_1-norm of coefficients, should coincide, however.

Regarding the computational cost, the CA requires, per iteration:

- Sparsifying step: one soft thresholding, which is performed elementwise, and thus it is $\mathcal{O}(N)$, and one analysis G^*, which is $\mathcal{O}(N \log N)$
- Reliable part: one synthesis G and one analysis G^*, both $\mathcal{O}(N \log N)$
- Each of the clipped parts: one synthesis, $\mathcal{O}(N \log N)$, and one elementwise projection, $\mathcal{O}(N)$.

We neglect simple addition of vectors and multiplication of a vector with a scalar, and it is assumed that, where applicable, parts of the formulas are precomputed. It is clear that the overall cost is dominated by the transforms, namely every iteration requires three applications of G and two of G^*.

On the other hand, the DR requires:

- Sparsifying step: one soft thresholding, which is $\mathcal{O}(N)$
- Projection onto K: one synthesis G and one pseudoinverse G^+, which is in the order of G^*, i.e. $\mathcal{O}(N \log N)$ in our particular setup; projection that is performed elementwise, $\mathcal{O}(N)$.

Again, the overall cost is dominated by the transforms, but we can see that the DR is much less demanding per iteration. Notice that this does not automatically mean that it will converge faster. We provide an experiment on this issue later.

The CA is a primal-dual method and needs more memory than the DR does. The exact factor depends on the redundancy of the transform G^*, i.e. on the ratio of M and N.

The practical advantage of the DR lies also in the fact that only one step size has to be hand-tuned instead of two in the CA.

4.7. Redundancy of the real-part operator

The operator $\Re(\cdot)$ can be omitted in (8) and (27) if its argument is real. As mentioned several times before, if the coefficient vector $\mathbf{c} \in (\mathbb{K}^Q \times \ldots \times \mathbb{K}^Q) \subset (\mathbb{C}^Q \times \ldots \times \mathbb{C}^Q)$, then the synthesis $G\mathbf{c}$ is real. It remains to check whether there exists an operation in any of the above algorithms that would make the coefficients fall out of this subspace: The soft thresholding is harmless in this regard. In the consolidated projection (26), $G^+ = G^*(GG^*)^{-1}$ has the same complex-conjugate structure as G^* since GG^* is a real diagonal matrix, according to the assumption. In the projection (22) onto the reliable set, it is straightforward to see that if the input coefficients belong to $\mathbb{K}^Q \times \ldots \times \mathbb{K}^Q$, then the same holds for the output, since $(M_\mathrm{R} GG^* M_\mathrm{R}^\top)$ is a real diagonal matrix.

4.8. Results

For the following experiments, five audio excerpts sampled at 16 kHz with length of approximately 5 seconds were chosen. Because the goal of the experiments is to demonstrate the convenience of the proposed projector in a practical example, rather than developing a novel method for audio declipping outperforming the current state-of-the-art, these audio excerpts are fully sufficient for this purpose. Used signals differ in tonal content and sparsity with respect to the used time-frequency representation to cover a wide spectrum of audio signals; they consists of recordings of acoustic guitar, double bass, artificial signature tone, speech and string orchestra.

As a preprocessing step, the signals were peak-normalized and then artificially clipped to multiple thresholds $\theta_\mathrm{c} \in \{0.3, 0.5, 0.7\}$. The clipping is considered to be symmetric, i.e. $\theta_\mathrm{c} = \theta_\mathrm{H} = -\theta_\mathrm{L}$.

Regarding the used transform, the common setting for audio declipping on 16 kHz sampled signals was adopted, i.e. the Gabor transform (STFT) with 1024 samples long Hann window (corresponding to 64 ms) and a 75% overlap. In all the cases, the number of frequency channels is the same as the window length, i.e. $Q = 1024$, except for the experiment shown in Fig. 5, where the number of frequency channels was twice as many as the window length, specifically 2048 channels. In both the settings it holds $GG^* = Id$.

It has been noticed that the convergence performance of Alg. 1 and Alg. 2 is influenced by the setting of parameters. These parameters were manually tuned such that both declipping algorithms performed well for all testing audio excerpts and all clipping thresholds. Specifically, the parameters of the CA were set to $\tau = 0.5$ and $\sigma = 0.666$ and the DR parameter γ was set to 1.

The results are evaluated using ΔSDR, the signal-to-distortion ratio improvement, which is the difference between SDR of the restored signal and SDR of the clipped signal. Formally:

$$\Delta\mathrm{SDR} = \mathrm{SDR}(\mathbf{y}, \hat{\mathbf{y}}) - \mathrm{SDR}(\mathbf{y}, \mathbf{y}_\mathrm{c}), \tag{28}$$

where $\mathbf{y}$ denotes the original (undistorted, ground-truth) signal, $\hat{\mathbf{y}}$ represents the restored signal and $\mathbf{y}_\mathrm{c}$ the clipped signal, while the SDR is the standard ratio computed as

$$\mathrm{SDR}(\mathbf{u},\mathbf{v}) = 10\log_{10}\frac{\|\mathbf{u}\|_2^2}{\|\mathbf{u}-\mathbf{v}\|_2^2}\ [\mathrm{dB}]. \tag{29}$$

Figures 2 and 3 are designed to demonstrate how the Condat algorithm and the Douglas-Rachford algorithm act over time, therefore the development of the ΔSDR and the objective function (ℓ_1-norm of the coefficients $\mathbf{c}$) for both the algorithms are presented in these figures. Whereas Fig. 2 displays only data obtained from reconstructing the "acoustic guitar" excerpt clipped at the level of 0.3, Fig. 3 uses the average of all five audio signals and three tested clipping thresholds. It can be concluded from the figures that the objective function somewhat correlates with the ΔSDR curve and both cases show that both algorithms converge to the solution with the same final objective function value, but the DR algorithm converges faster, i.e. the curves level up faster reaching slightly higher ΔSDR value. This is caused most likely by the projection step, where in the CA, there are three individual projections (onto R, H and L) combined in a sum, thus it is not as accurate as in the DR case, where the projection step (26) is exploited.

Since the horizontal axis was converted to time in seconds in the figures, they also offer an overview of the number of iterations, which is handy since the iterations of CA and DR do not consume the same amount of time. For this reason, apart from the ΔSDR and objective function graphs, the (+ and ×) markers emphasizing every 100th iteration are also displayed in the plots. Note that although both algorithms run for exactly 1000 iterations, since the time axis is limited to 26 seconds, only 570 iterations of the Condat algorithm are shown in Fig. 2, and 581 iterations in Fig. 3.

The average computational times for all sound excerpts and all computed clipping thresholds with a fixed number of 1000 iterations were 44.73 seconds for the CA and 23.82 seconds for the DR.

More detailed time comparison of both algorithms can be seen in Figs. 4 and 5 where the time ratio (time of DR divided by the time of CA) is plotted for every sound excerpt and clipping threshold θ_c combination. Here both algorithms ran until the objective function differed by 0.1 % from the state of full convergence, which assessment was determined to be reached after exactly 3000 iterations. With the average value of the plotted time ratios being ca 0.53, it is possible to claim that the DR algorithm is about twice as fast as the CA.

Fig. 5 presents the same time ratios as Fig. 4 with only one difference—the number of frequency channels of the Gabor transform is set to 2048 instead of 1024. The average value of the time ratio is now around 0.42. It can be assumed that the more DGT coefficients we will work with (either longer windows, bigger overlaps or more frequency channels), the more significant time difference between the two algorithms there will be.

The algorithms were implemented and tested in MATLAB R2017a using the LTFAT toolbox [45] and supporting the idea of reproducible research, the codes are available at

http://www.utko.feec.vutbr.cz/~rajmic/software/project_accel_declip.zip

All the experiments ran on a PC with Intel i7-3770 3.4 GHz, 24 GB RAM and OS Windows 10 Pro (version 1809).

5. Conclusion

A new projector onto a box-shaped convex set in the transformed domain has been presented in this article. The projector forms a generalization of previous results and its use was demonstrated on a simple audio signal declipping task.

Author Contributions: Conceptualization and methodology, P.R.; formal analysis, V.V. and O.M.; software, experiments, visualization of results, P.Z.; draft preparation, review and editing, P.R., V.V., P.Z. and O.M.

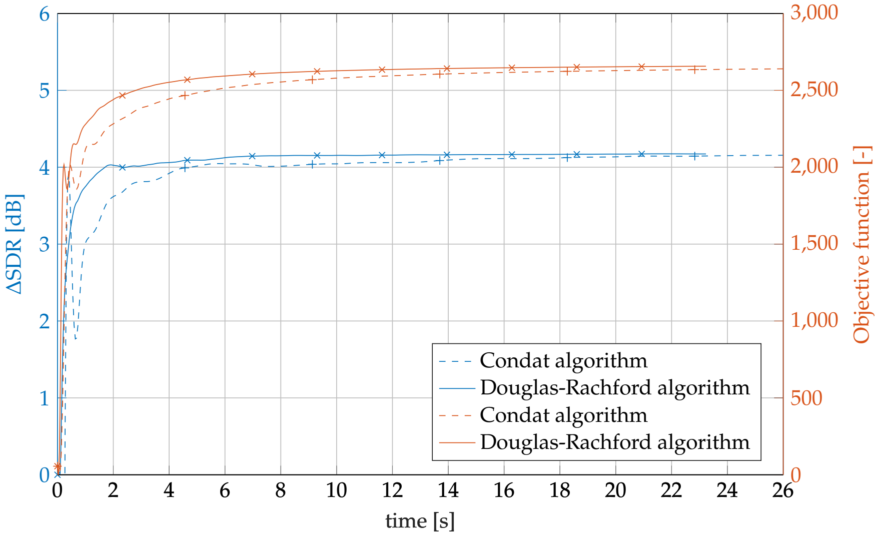

Figure 2. Development of the ΔSDR (blue) and objective function (orange) through iterations for the particular "acoustic guitar" excerpt and the clipping threshold $\theta_c = 0.3$. Both algorithms run for a fixed number of 1000 iterations here. To demonstrate the time differences between the algorithms, every 100th iteration is emphasized with a marker (+ for Condat and × for Douglas-Rachford).

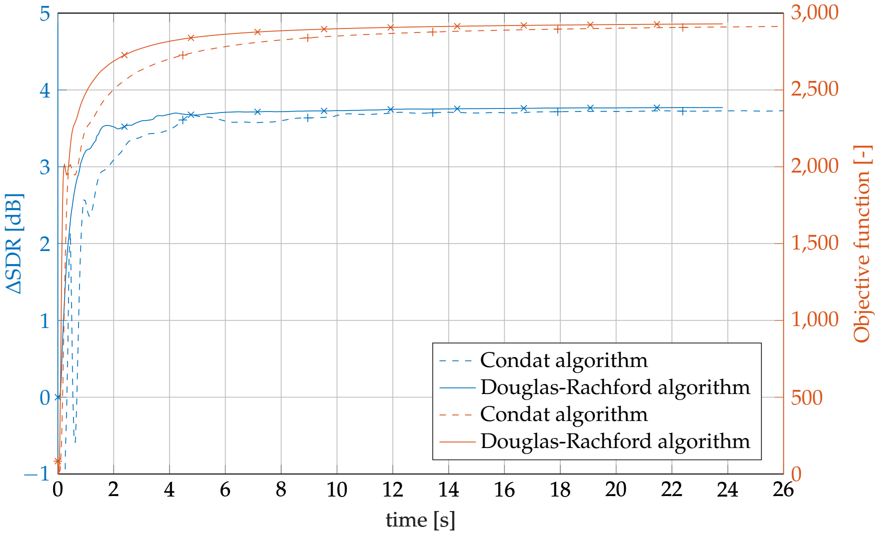

Figure 3. Analog to Fig. 2, development of the ΔSDR (blue) and objective function (orange) through iterations is shown, but now averaged over the testing sounds and the clipping thresholds. The number of iterations was 1000.

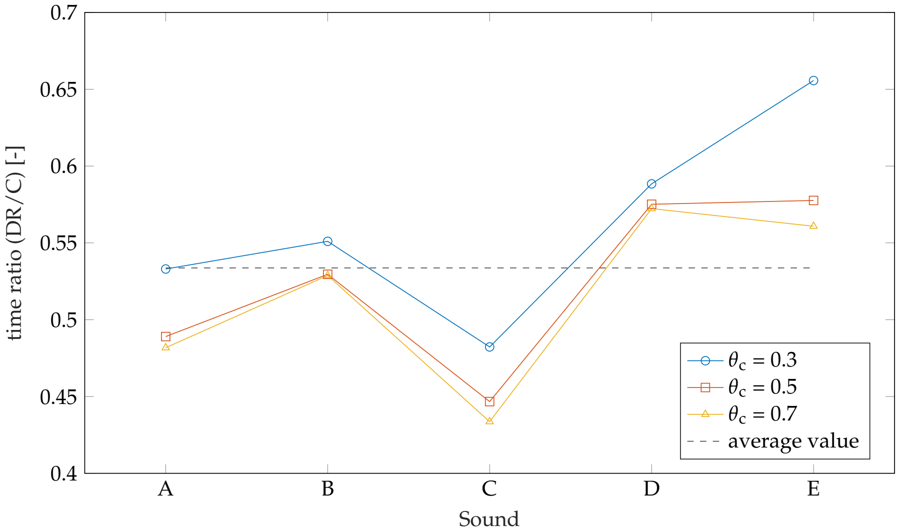

Figure 4. Relative elapsed time, Douglas-Rachford versus Condat algorithm, for all testing sounds and clipping thresholds. Both the algorithms were first let to fully converge, which was in practice observed after 3000 iterations. Then, both algorithms ran again until the objective function differed by 0.1 % from the respective solutions. The Gabor transform with 1024 samples long Hann window and 1024 frequency channels with 75 % overlap was used.

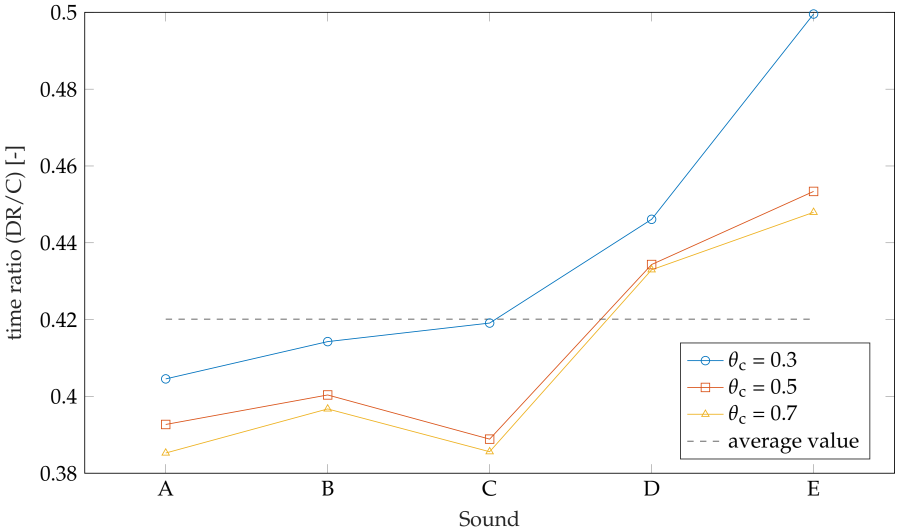

Figure 5. Analog to Fig. 4, this picture presents relative running times of the Douglas-Rachford versus Condat algorithm. The only difference is that the Gabor transform now used 2048 frequency channels.

Funding: 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. Research described in this paper was financed by the National Sustainability Program under grant LO1401. Infrastructure of the SIX Center was used.

Acknowledgments: The authors thank Petr Sysel, Nicki Holighaus and Zdeněk Průša for discussing parts of the draft.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

CA	Condat Algorithm
DCT	Discrete Cosine Transform
DGT	Discrete Gabor Transform
DR	Douglas-Rachford (algorithm)
MDCT	Modified Discrete Cosine Transform
PA	Proximal Algorithm
SDR	Signal-to-Distortion Ratio
STFT	Short-Time Fourier Transform

References

1. Combettes, P.; Pesquet, J. A Douglas–Rachford splitting approach to nonsmooth convex variational signal recovery. *IEEE Journal of Selected Topics in Signal Processing* **2007**, *1*, 564–574. doi:10.1109/JSTSP.2007.910264.
2. Combettes, P.; Wajs, V. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation* **2005**, *4*, 1168–1200.
3. Combettes, P.; Pesquet, J. Proximal splitting methods in signal processing. *Fixed-Point Algorithms for Inverse Problems in Science and Engineering* **2011**, pp. 185–212. doi:10.1007/978-1-4419-9569-8_10.
4. Chambolle, A.; Pock, T. A First-Order Primal-Dual Algorithm for Convex Problems with Applications to Imaging. *Journal of Mathematical Imaging and Vision* **2011**, *40*, 120–145. doi:10.1007/s10851-010-0251-1.
5. Komodakis, N.; Pesquet, J. Playing with Duality: An Overview of Recent Primal-Dual Approaches for Solving Large-Scale Optimization Problems. *IEEE Signal Processing Magazine* **2015**, *32*, 31–54. doi:10.1109/MSP.2014.2377273.
6. Condat, L. A Generic Proximal Algorithm for Convex Optimization—Application to Total Variation Minimization. *Signal Processing Letters, IEEE* **2014**, *21*, 985–989. doi:10.1109/LSP.2014.2322123.
7. Condat, L. A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms. *Journal of Optimization Theory and Applications* **2013**, *158*, 460–479. doi:10.1007/s10957-012-0245-9.
8. Moreau, J.J. Proximité et dualité dans un espace hilbertien. *Bulletin de la société mathématique de France* **1965**, *93*, 273–299.
9. Bauschke, H.H.; Combettes, P.L. *Convex analysis and monotone operator theory in Hilbert spaces*, 2 ed.; Springer: Cham, Switzerland, 2011.
10. Boyd, S.P.; Vandenberghe, L. *Convex Optimization*; Cambridge University Press, 2004.
11. Šorel, M.; Bartoš, M. Efficient JPEG decompression by the alternating direction method of multipliers. 2016 23rd International Conference on Pattern Recognition (ICPR), 2016, pp. 271–276. doi:10.1109/ICPR.2016.7899645.
12. Fadili, M.; Starck, J.L. Monotone operator splitting for optimization problems in sparse recovery. IEEE Publishing, 2009, pp. 1461–1464.
13. Gröchenig, K. *Foundations of time-frequency analysis*; Birkhäuser, 2001.
14. Christensen, O. *An Introduction to Frames nad Riesz Bases*; Birkhäuser: Boston-Basel-Berlin, 2003.
15. Balazs, P.; Dörfler, M.; Jaillet, F.; Holighaus, N.; Velasco, G. Theory, implementation and applications of nonstationary Gabor frames. *Journal of computational and applied mathematics* **2011**, *236*, 1481–1496. doi:10.1016/j.cam.2011.09.011.

16. Necciari, T.; Balazs, P.; Holighaus, N.; Sondergaard, P. The ERBlet transform: An auditory-based time-frequency representation with perfect reconstruction. Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, 2013, pp. 498–502. doi:10.1109/ICASSP.2013.6637697.
17. Meyer, C.D. *Matrix Analysis and Aplied Linear Algebra*; SIAM, 2000.
18. Etter, W. Restoration of a discrete-time signal segment by interpolation based on the left-sided and right-sided autoregressive parameters. *IEEE Transactions on Signal Processing* **1996**, *44*, 1124–1135. doi:10.1109/78.502326.
19. Abel, J.; Smith, J. Restoring a clipped signal. Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on, 1991, pp. 1745–1748 vol.3. doi:10.1109/ICASSP.1991.150655.
20. Dahimene, A.; Noureddine, M.; Azrar, A. A simple algorithm for the restorationof clipped speech signal. *Informatica* **2008**, *32*, 183–188.
21. Selesnick, I. Least Squares with Examples in Signal Processing, 2013.
22. Bilen, C.; Ozerov, A.; Perez, P. Audio declipping via nonnegative matrix factorization. Applications of Signal Processing to Audio and Acoustics (WASPAA), 2015 IEEE Workshop on, 2015, pp. 1–5. doi:10.1109/WASPAA.2015.7336948.
23. Adler, A.; Emiya, V.; Jafari, M.; Elad, M.; Gribonval, R.; Plumbley, M. Audio Inpainting. *IEEE Transactions on Audio, Speech, and Language Processing* **2012**, *20*, 922–932. doi:10.1109/TASL.2011.2168211.
24. Adler, A.; Emiya, V.; Jafari, M.; Elad, M.; Gribonval, R.; Plumbley, M. A constrained matching pursuit approach to audio declipping. Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, 2011, pp. 329–332. doi:10.1109/ICASSP.2011.5946407.
25. Defraene, B.; Mansour, N.; Hertogh, S.D.; van Waterschoot, T.; Diehl, M.; Moonen, M. Declipping of Audio Signals Using Perceptual Compressed Sensing. *IEEE Transactions on Audio, Speech, and Language Processing* **2013**, *21*, 2627–2637. doi:10.1109/TASL.2013.2281570.
26. Siedenburg, K.; Kowalski, M.; Dorfler, M. Audio declipping with social sparsity. Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on. IEEE, 2014, pp. 1577–1581.
27. Kitić, S.; Jacques, L.; Madhu, N.; Hopwood, M.; Spriet, A.; De Vleeschouwer, C. Consistent iterative hard thresholding for signal declipping. Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, 2013, pp. 5939–5943. doi:10.1109/ICASSP.2013.6638804.
28. Kitić, S.; Bertin, N.; Gribonval, R. Audio Declipping by Cosparse Hard Thresholding. 2nd Traveling Workshop on Interactions between Sparse models and Technology, 2014.
29. Kitić, S.; Bertin, N.; Gribonval, R. Sparsity and cosparsity for audio declipping: a flexible non-convex approach. LVA/ICA 2015 – The 12th International Conference on Latent Variable Analysis and Signal Separation; , 2015.
30. Záviška, P.; Rajmic, P.; Průša, Z.; Veselý, V. Revisiting Synthesis Model in Sparse Audio Declipper. Latent Variable Analysis and Signal Separation; Deville, Y.; Gannot, S.; Mason, R.; Plumbley, M.D.; Ward, D., Eds.; Springer International Publishing: Cham, 2018; pp. 429–445.
31. Záviška, P.; Mokrý, O.; Rajmic, P. A Proper Version of Synthesis-based Sparse Audio Declipper. 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP); , 2019.
32. Weinstein, A.J.; Wakin, M.B. Recovering a Clipped Signal in Sparseland. *CoRR* **2011**, *abs/1110.5063*.
33. Christensen, O. *Frames and Bases, An Introductory Course*; Birkhäuser: Boston, 2008.
34. Donoho, D.L.; Elad, M. Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization. *Proceedings of The National Academy of Sciences* **2003**, *100*, 2197–2202.
35. Bayram, I.; Kamasak, M. A Simple Prior for Audio Signals. *IEEE Transactions on Acoustics Speech and Signal Processing* **2013**, *21*, 1190–1200. doi:10.1109/TASL.2013.2245652.
36. Bayram, I.; Akykıldız, D. Primal-dual algorithms for audio decomposition using mixed norms. *Signal, Image and Video Processing* **2014**, *8*, 95–110. doi:10.1007/s11760-013-0472-z.
37. Rajmic, P.; Bartlová, H.; Průša, Z.; Holighaus, N. Acceleration of Audio Inpainting by Support Restriction. 7th International Congress on Ultra Modern Telecommunications and Control Systems, 2015.
38. Průša, Z.; Søndergaard, P.; Balazs, P.; Holighaus, N. LTFAT: A Matlab/Octave toolbox for sound processing. Proceedings of the 10th International Symposium on Computer Music Multidisciplinary Research (CMMR 2013); Laboratoire de Mécanique et d'Acoustique, Publications of L.M.A.: Marseille, France, 2013; pp. 299–314.

39. Holighaus, N.; Wiesmeyr, C. Construction of warped time-frequency representations on nonuniform frequency scales, Part I: Frames. *submitted, preprint available: arXiv:1409.7203* **2016**.

40. Daubechies, I.; Grossmann, A.; Meyer, Y. Painless nonorthogonal expansions. *J. Math. Phys.* **1986**, *27*, 1271–1283. doi:10.1063/1.527388.

41. Søndergaard, P.L.; Torrésani, B.; Balazs, P. The Linear Time Frequency Analysis Toolbox. *International Journal of Wavelets, Multiresolution Analysis and Information Processing* **2012**, *10*. doi:10.1142/S0219691312500324.

42. Donoho, D. De-noising by soft-thresholding. *IEEE Transactions on Information Theory* **1995**, *41*, 613 –627.

43. Bruckstein, A.M.; Donoho, D.L.; Elad, M. From Sparse Solutions of Systems of Equations to Sparse Modeling of Signals and Images. *SIAM Review* **2009**, *51*, 34–81.

44. Rajmic, P. Exact risk analysis of wavelet spectrum thresholding rules. Electronics, Circuits and Systems, 2003. ICECS 2003. Proceedings of the 2003 10th IEEE International Conference on, 2003, Vol. 2, pp. 455–458 Vol.2. doi:10.1109/ICECS.2003.1301820.

45. Průša, Z.; Søndergaard, P.L.; Holighaus, N.; Wiesmeyr, C.; Balazs, P. The Large Time-Frequency Analysis Toolbox 2.0. In *Sound, Music, and Motion*; Aramaki, M.; Derrien, O.; Kronland-Martinet, R.; Ystad, S., Eds.; Lecture Notes in Computer Science, Springer International Publishing, 2014; pp. 419–442. doi:{10.1007/978-3-319-12976-1_25}.