CASCADE: Channel-Aware Structured Cosparse Audio DEclipper
Clément Gaultier, Nancy Bertin, Rémi Gribonval

To cite this version:

HAL Id: hal-01714667
https://inria.hal.science/hal-01714667
Submitted on 21 Feb 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
CASCADE: CHANNEL-AWARE STRUCTURED COSPARSE AUDIO DECLIPPER

Clément Gaultier*, Nancy Bertin*, Rémi Gribonval*

* Univ Rennes, Inria, CNRS, IRISA, France

ABSTRACT
This work features a new algorithm, CASCADE, which leverages a structured cosparsity prior across channels to address the multichannel audio declipping problem. CASCADE technique outperforms the state-of-the-art method A-SPADE applied on each channel separately in all tested settings, while retaining similar runtime.

Index Terms— declipping, multichannel, structured sparsity, cosparsity

1. INTRODUCTION

Clipping, also known as saturation, is a common phenomenon that can arise from hardware or software limitations in any audio acquisition pipeline. It results in severely distorted audio recordings. Declipping consists in performing the inverse process, in order to restore saturated audio signals and improve their quality.

1.1. State-of-the-art in single-channel declipping

While we can trace back some attempts to address this issue, e.g. with autoregressive models [1], to several decades, significant progress towards efficient desaturation was recently made in several directions. First, the declipping problem was recast as an undetermined, linear inverse problem, akin to inpainting, which could be addressed by means of a sparse regularization [2]. On this basis, algorithmic frameworks evolved from usual greedy algorithms to thresholding [3] then to non-convex approaches [4]. In parallel, a switch from a (now) traditional sparse synthesis approach, to a sparse analysis (also known as cosparsely model [5]) was proposed, as well as some model refinements exploiting notions of structured sparsity, especially that of social sparsity in the time-frequency domain [6]. These layers, in line of which the current paper is written, led to significant improvements in reconstruction accuracy and computational efficiency.

1.2. Multichannel declipping

However, it must be noted that all these methods were developed and tested for mono signals, while multichannel data now represent a large part of available audio content, from stereo to more and more channels. Intuitively, we expect that a joint processing of all channels could be more efficient than declipping independently each channel with the previous single-channel algorithms. To date, the multichannel joint declipping problem has only been addressed by [7] through a modeling of the signals as mixtures of sound sources, in order to encompass inter-channel correlations. This approach requires prior knowledge or estimation of the mixing process.

In this work, we propose a blinder approach to joint declipping of multichannel audio from compact antennas, which operates purely at the signal level and does not require any kind of spatial information (including the microphone positions). Our method is based on the aforementioned ideas, namely a cosparsity model of data, with the original addition of a structured sparsity prior across channels which allows to take implicitly into account the spatial correlation.

1.3. Structured sparsity

In the field of sparse representations and techniques, the notion of structure which is basically the idea that the nonzero coefficients of expectedly sparse quantities may not be “indifferently” distributed, is manifold. It has given rise to various definitions and developments, all of which were initially defined in the context of sparse synthesis, but can all be straightforwardly extended to the sparse analysis point of view.

- Joint or simultaneous sparsity. Several vectors are gathered and assumed to admit a sparse decomposition on the same dictionary, and the sparse decomposition can be jointly performed [8][9].
- Group sparsity. The index set of the considered sparse vector is partitioned into non-overlapping groups, and the signal is assumed to be sparse at the group level but not within an active group. This prior is typically enforced by mixed-norms such as the \( f_{1,2} \)-norm [10].
- Social sparsity. The previous structure is extended to the case of possibly overlapping groups or neighborhoods enforced by the use of a Persistent Empirical Wiener shrinkage [11].

1.4. Contributions and outline

In this paper, we propose an original multichannel declipping method based on a twofold prior: simultaneous cosparsity of all channels, and group sparsity across channels. Section 2 introduces our notations and models. The resulting algorithm, which outperforms the naive channel-by-channel declipping strategy while keeping computation time in the same order of magnitude, is presented in Section 3 and experimentally validated on real eight-channel audio data in Section 4. We conclude in Section 5.

2. NOTATIONS AND MODEL

2.1. Notations

We observe a time-domain multichannel clipped audio signal composed of \( K \) channels. \( \mathbf{Y}_n \in \mathbb{R}^{J \times K} \) the \( n \)-th windowed frame that signal and its clean version \( \mathbf{X}_n \). We define \( \mathbf{Z}_n \simeq \mathbf{A X}_n \) a frequency representation of \( \mathbf{X}_n \) such that \( \mathbf{Z}_n \in \mathbb{C}^{P \times K} \) and \( \mathbf{A} \in \mathbb{C}^{P \times J} \), where \( \mathbf{A} \) performs an analysis frequency transform (possibly redundant when

State-of-the-art results can be appreciated for instance from the SPADE software webpage: [https://spade.inria.fr/]

1
2.2. Clipping model

We use the forward hard-clipping degradation model for \( Y_n \):

\[
Y_{jk} = \begin{cases} 
  x_{jk} & \text{for } |x_{jk}| \leq \tau; \\
  \text{sgn}(x_{jk})\tau & \text{otherwise;}
\end{cases}
\]

with \( y_{jk} \) (resp. \( x_{jk} \)) the \( j \)-th sample recorded on the \( k \)-th channel from \( Y_n \) (resp. \( X_n \)) and \( \tau \) the hard-clipping level in the \( k \)-th channel.

2.3. Channel-aware structured cosparse modeling

The main model characteristics for this work derive from the relation between \( Z_n \) and \( X_n \) as well as properties of \( Z_n \) which are: i) \( A \in \mathbb{C}^{P \times J} \), \( P \geq J \); ii) \( Z_n \cong AX_n, Z_n \in \mathbb{C}^{P \times K} \); iii) \( \|Z_n\|_0 \ll P \times K \); iv) \( Z_n \) is “structured across channels”. The underlying hypothesis behind the structure (group sparsity) in the frequency representation \( Z_n \) is that non zero coefficients are roughly distributed equivalently from one channel to another. This way we encompass a channel-aware structured sparse prior on \( Z_n \) and a cosparse prior on \( X_n \) so the name of the proposed algorithm: “Channel-Aware Structured Cosparse Audio DEclipper (CASCADE)”.

3. MULTICHANNEL DECLIPPING ALGORITHM

The goal of the algorithm is to simultaneously declip each channel in the observation \( Y_n \) to output an estimate \( \hat{X}_n \) which satisfies: i) the channel-aware structured cosparse modeling constraint, ii) the data fidelity constraint regarding the clipped \( Y_n \). For that we use an iterative algorithm which alternatively projects the solution on the modeling and the declipping constraints. Projection on the modeling constraint is achieved using the Group Empirical Wiener (GEW) operator presented below.

3.1. Sparsifying operator

For clarity, in the following we remove the time-frame index \( n \) subscript and consider \( X, Y, Z \) matrices of size \((J \times K)\) or \((P \times K)\) corresponding to the \( n \)-th time-frame. To handle structured cosparse constraints, we use the GEW operator as in [12] as a sparsifying step in the declipping procedure. Let \( Z \in \mathbb{C}^{P \times K} \) be a local multichannel frequency representation to sparsify. Let \( pk \) be coordinates of such a point and \( z_{pk} \in \mathbb{C}^{1 \times K} \) the \( p \)-th row from matrix \( Z \) (corresponding to a group, as illustrated on Fig. 1). GEW is defined as:

\[
S_p(Z)_{pk} = Z_{pk} \cdot \left( 1 - \frac{\mu^2}{\|Z_{pk}\|^2} \right),
\]

with \((\cdot)_+ = \max(\cdot, 0)\) the positive part and \( \mu \) the parameter controlling the amount of shrinkage to apply. This shrinkage explicitly promotes group sparsity of \( Z \) along the channel dimension.

3.2. Projection on the declipping constraint

For the data fidelity constraint, we define \( \Omega \), the set of reliable samples indices \( jk \) in \( Y_n \). We also define respectively \( \Omega_+ \) and \( \Omega_- \) the sets of clipped positive and clipped negative samples indices. We note that in the case of hard-clipping, the sets are easily retrieved comparing each sample to the clipping level. The notation \( V_\Omega \) denotes the matrix formed by considering only those indices of \( V \) indexed by \( \Omega \) while \( \prec, \succ, \leq, \geq, \bowtie \) are used for entry-wise comparisons between matrices. The data fidelity projection is the solution of the following optimization problem:

\[
\minimize_{X \in \Theta} \|AX - Z\|_F^2, \tag{3}
\]

with \( \Theta \) the magnitude constraint convex set expressed as:

\[
\Theta = \left\{ X \mid \begin{array}{l}
X_{\Omega_+} = Y_{\Omega_+};
X_{\Omega_-} \succ Y_{\Omega_-};
X_{\Omega_-} \prec Y_{\Omega_-};
\end{array} \right\}. \tag{4}
\]

From [4], we can rewrite the solution of [3] as the following component-wise magnitude constraints:

\[
\hat{X}_{jk} = \begin{cases} 
  Y_{jk} & \text{if } jk \in \Omega_+; \\
  \mu (A^H Z)_{jk} & \text{if } jk \in \Omega_-, \mu (A^H Z)_{jk} \geq \tau; \\
  \text{sgn}(Y_{jk})\tau & \text{or } jk \in \Omega_-, \mu (A^H Z)_{jk} \leq -\tau; \\
  \text{otherwise.}
\end{cases}
\]

(5)

3.3. Overall functioning for the algorithm

As for A-SPADE [4], the algorithm is built on the Alternating Direction Method of Multipliers (ADMM) numerical scheme [13]. Pseudo code in Algorithm 1 presents the functioning of the CASCADE algorithm for a given frame \( Y \in \mathbb{R}^{J \times K} \) with \( U \in \mathbb{C}^{J \times K} \) the ADMM dual variable. In Algorithm 1 the \( \mu \) parameter is of great importance as it tells the procedure how aggressively to perform the sparsification step. This value is updated along with the iterations following a geometric progression of common ratio \( \alpha \). This way the algorithm relaxes the sparsity constraint while it progresses. Typical values for \( \mu(0) \) and \( \alpha \) are given in section 4.1.

The procedure is applied in a frame-based manner and outputs the \( X_n \) declipped estimates, which are used to rebuild the full length estimated signal by means of overlap-and-add synthesis.

4. EXPERIMENTS

We perform experiments on 8 channels recordings excerpts from the VoiceHome2 Corpus [14]. We use the 359 clean speech available.
Table 1. CASCADE parameters

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Window size [samples]</th>
<th>Overlap [%]</th>
<th>Window Type</th>
<th>Channel number</th>
<th>Maximum iterations</th>
<th>Accuracy</th>
<th>Analysis operator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Value</td>
<td>J = 1024</td>
<td>75</td>
<td>Hamming</td>
<td>K = 8</td>
<td>(i_{\text{max}} = 10^6)</td>
<td>(\beta = 10^{-1})</td>
<td>A = DFT</td>
</tr>
</tbody>
</table>

Fig. 2. Speech declipping numerical results: SDR improvement [dB]

Fig. 3. Speech&Music declipping numerical results: SDR improvement [dB]
Algorithm 1 CASCADE algorithm

Require: A, Y, Γ, α, μ(0), Θ, β, i_{max}

Initialization step:
\[ X^{(0)} = Y, U^{(0)} = 0, i = 1 \]

Projection step on the modeling constraint:
\[ Z^{(i)} = S_{R}(i-1) (AX^{(i-1)} + U^{(i-1)}) \]

Projection step on declipping constraint:
\[ X^{(i)} = \text{argmin}_{X} \|AX - Z^{(i)} + U^{(i-1)}\|_F^2 \times X \]
subject to \( X \in \Theta \)

Update step:
\[ \mu^{(i)} = \alpha \mu^{(i-1)} \]
if \( \frac{\|AX^{(i)} - Z^{(i)}\|_F}{\|AX^{(0)}\|_F} \leq \beta \) or \( i \geq i_{\text{max}} \) then
terminate
else
\[ U^{(i)} = U^{(i-1)} + A \hat{X}^{(i)} - Z^{(i)} \]
\[ i \leftarrow i + 1 \]
end if
return \( \hat{X}^{(i)} \)

examples (total duration: about one hour) and the 118 mixed music and speech examples (total duration: 20 minutes). We artificially saturated all the excerpts at five signal to distortion ratios (SDR) levels in dB: 5, 10, 15, 20, 25. The analysis operator \( A \in \mathbb{C}^{p \times J} \) is a possibly redundant Discrete Fourier Transform (DFT); indeed, we study the effect of the frequency transform redundancy by comparing two redundancy factors: \( R = 1 \), \( R = 2 \) (we recall that \( P = RJ \)).

A first pilot study (data not shown) allowed us to choose the best parameters \( \alpha \) and \( \mu^{(0)} \). So far, the best results are obtained with \( \mu^{(0)} = 1024 \) and \( \alpha = 0.99 \). Other parameters of the algorithm are listed in Table 1. We confront the CASCADE algorithm with the A-SPADE state-of-the-art decliper (which uses a simple cosparsity prior and operates on each channel separately) and compare results channel-by-channel. Performance is assessed by SDR improvement and runtime.

4.1. Quality improvements

SDR improvement results are presented in Fig. 2 (for speech only subset) and Fig. 3 (for mixed music and speech). We observe that the CASCADE method outperforms the A-SPADE algorithm by 1 dB to more than 3 dB in all settings. The improvement brought by CASCADE over A-SPADE is even more salient on mixed speech and music data (which is the most difficult subset, with a globally lower performance for both algorithms, compared to that obtained on speech only data.)

The effect of a redundant DFT transform (\( R=2 \)) appears to be slightly different for each method. We note that except for Fig. 3a twice redundant DFT provides at least as good results than non redundant DFT for A-SPADE. For the CASCADE method, redundant DFT is profitable for mild to high input SDR (15 dB to 25 dB), but detrimental at low input SDR (high saturation).

4.2. Computational Aspects

As DFT can be efficiently implemented with a fast transform, the computational cost of the declipping procedure mainly stems from the sparsifying step and the projection on the declipping constraint.

Table 2. Runtime tests numerical results

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CASCADE</th>
<th>A-SPADE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redundancy</td>
<td>R=1</td>
<td>R=2</td>
</tr>
<tr>
<td>Input SDR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>167</td>
<td>398</td>
</tr>
<tr>
<td>10</td>
<td>120</td>
<td>265</td>
</tr>
<tr>
<td>15</td>
<td>80</td>
<td>177</td>
</tr>
<tr>
<td>20</td>
<td>54</td>
<td>119</td>
</tr>
<tr>
<td>25</td>
<td>37</td>
<td>78</td>
</tr>
</tbody>
</table>

(a) Runtime performance (ratio to realtime processing)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>CASCADE</th>
<th>A-SPADE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Redundancy</td>
<td>R=1</td>
<td>R=2</td>
</tr>
<tr>
<td>Input SDR</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>11.11</td>
<td>10.76</td>
</tr>
<tr>
<td>10</td>
<td>12.39</td>
<td>12.45</td>
</tr>
<tr>
<td>15</td>
<td>13.31</td>
<td>13.39</td>
</tr>
<tr>
<td>20</td>
<td>14.01</td>
<td>14.32</td>
</tr>
<tr>
<td>25</td>
<td>14.40</td>
<td>14.44</td>
</tr>
</tbody>
</table>

(b) Corresponding improvements (\( \Delta \)SDR)

For this runtime comparisons, we choose a subset of 25 excerpts (totalizing 3 minutes of audio) from the dataset and compare the computing time of the CASCADE and the A-SPADE algorithms. The runtime tests are performed on workstations running the Matlab® associated code in single-thread mode. The computers are equipped with Intel® Xeon® CPU 5140 @ 2.33 GHz with 2 GB available RAM memory. Table 2a shows runtime performances and Table 2b the corresponding SDR improvements (\( \Delta \)SDR) averaged on the eight channels and the 25 excerpts. We clearly note higher computing times for both methods with twice redundant DFT. The A-SPADE method is 2.4 to 2.6 times faster in the non redundant case while CASCADE is between 2.1 to 2.4 faster with this setting. We observe that substantial improvements given by the CASCADE algorithm are achieved at the cost of only slightly lower computational efficiency (1.5 to 2.2 times slower than A-SPADE). These different computation time characteristics might come from the properties of the sparsifying operator when used inside the ADMM framework and the total number of iterations needed to finish or converge.

5. CONCLUSION

We introduced a new algorithm combining a cosparsity prior with structure in the frequency-channel domain to address the multi-channel audio declipping problem. We showed that adding across-channel structure on top of cosparsity-based state-of-the-art method applied channel-wise. In addition, we showed that performance can be improved by the use of a redundant frequency transform when the clipping level is moderate. Finally, we demonstrated that the method implies a very limited runtime overcost. Future studies could include perceptual assessments, and model integration of time-frequency structures on top of structured cosparsity across channels.

6. ACKNOWLEDGMENTS

This work was supported in part by the ERC, PLEASE project (ERC-StG-2011-277906) and Région Bretagne. The authors thank Matthieu Kowalski and Srđan Kitić for precious advice.
7. REFERENCES


