CASCADE: Channel-Aware Structured Cosparse Audio DEclipper

Clément Gaultier, Nancy Bertin, Rémi Gribonval

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 cosparse 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 *cosparse* 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].

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.

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

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 cosparse 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 manyfold. 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 $\ell_{1,2}$-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. $\mathbf{Y}_\mathsf{n} \in \mathbb{R}^{\mathsf{J}\times\mathsf{K}}$ the n^{th} windowed frame that signal and its clean version $\mathbf{X}_\mathsf{n}$. We define $\mathbf{Z}_\mathsf{n} \simeq \mathbf{A}\mathbf{X}_\mathsf{n}$ a frequency representation of $\mathbf{X}_\mathsf{n}$ such that $\mathbf{Z}_\mathsf{n} \in \mathbb{C}^{\mathsf{P}\times\mathsf{K}}$ and $\mathbf{A} \in \mathbb{C}^{\mathsf{P}\times\mathsf{J}}$, where $\mathbf{A}$ performs an analysis frequency transform (possibly redundant when

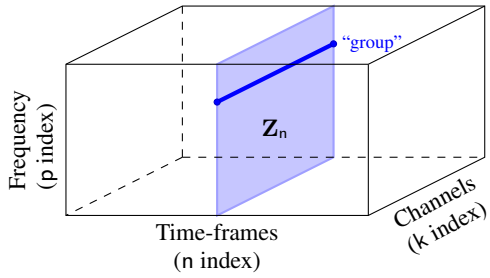


Fig. 1. Schematic representation of the data. In a given channel k, the extracted layer is a time-frequency representation of this channel. At a given time frame n, the channel-frequency layer $\mathbf{Z}_n$ is *group* sparse in the channel dimension.

$P > J$), P is the number of frequency bins used for the frequency representation, J is the number of time-domain samples in a frame. We consider $P = RJ$ with R the redundancy factor of $\mathbf{A}$. Lastly, K is the number of channels. Fig. 1 displays a schematic representation of the involved quantities.

2.2. Clipping model

We use the forward hard-clipping degradation model for $\mathbf{Y}_n$:

$$\mathbf{y}_{jk} = \begin{cases} \mathbf{x}_{jk} & \text{for } |\mathbf{x}_{jk}| \leq \tau_k; \\ \operatorname{sgn}(\mathbf{x}_{jk})\tau_k & \text{otherwise;} \end{cases} \tag{1}$$

with $\mathbf{y}_{jk}$ (resp. $\mathbf{x}_{jk}$) the j^{th} sample recorded on the k^{th} channel from $\mathbf{Y}_n$ (resp. $\mathbf{X}_n$) and τ_k 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 $\mathbf{Z}_n$ and $\mathbf{X}_n$ as well as properties of $\mathbf{Z}_n$ which are: i) $\mathbf{A} \in \mathbb{C}^{P\times J}, P \geq J$; ii) $\mathbf{Z}_n \simeq \mathbf{A}\mathbf{X}_n, \mathbf{Z}_n \in \mathbb{C}^{P\times K}$; iii) $\|\mathbf{Z}_n\|_0 \ll P \times K$; iv) $\mathbf{Z}_n$ is "structured across channels". The underlying hypothesis behind the structure (group sparsity) in the frequency representation $\mathbf{Z}_n$ 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 $\mathbf{Z}_n$ and a cosparse prior on $\mathbf{X}_n$ 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 $\mathbf{Y}_n$ to output an estimate $\hat{\mathbf{X}}_n$ which satisfies: i) the channel-aware structured cosparsity modeling constraint, ii) the data fidelity constraint regarding the clipped $\mathbf{Y}_n$. 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 $\hat{\mathbf{X}}, \mathbf{Y}, \mathbf{Z}$, 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 $\mathbf{Z} \in \mathbb{C}^{P\times K}$ be a local multichannel frequency representation to sparsify. Let pk be coordinates of such a point and $\mathbf{z}_p \in \mathbb{C}^{1\times K}$ the p-th row from matrix $\mathbf{Z}$ (corresponding to a group, as illustrated on Fig. 1). GEW is defined as:

$$\mathcal{S}_\mu(\mathbf{Z})_{pk} = \mathbf{Z}_{pk} \cdot \left(1 - \frac{\mu^2}{\|\mathbf{z}_p\|_2^2}\right)_+, \tag{2}$$

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

3.2. Projection on the declipping constraint

For the data fidelity constraint, we define Ω_r the set of reliable samples indices jk in $\mathbf{Y}_n$. We also define respectively Ω_+ and Ω_- 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 $\mathbf{V}_\Omega$ denotes the matrix formed by considering only those indices of $\mathbf{V}$ indexed by Ω while $\preccurlyeq, \succcurlyeq, \prec, \succ$ are used for entry-wise comparisons between matrices. The data fidelity projection is the solution of the following optimization problem:

$$\underset{\mathbf{X}\in\Theta}{\text{minimize}} \ \|\mathbf{A}\mathbf{X} - \mathbf{Z}\|_F^2, \tag{3}$$

with Θ the magnitude constraint convex set expressed as:

$$\Theta = \left\{ \mathbf{X} \mid \begin{array}{l} \mathbf{X}_{\Omega_r} = \mathbf{Y}_{\Omega_r}; \\ \mathbf{X}_{\Omega_+} \succcurlyeq \mathbf{Y}_{\Omega_+}; \\ \mathbf{X}_{\Omega_-} \preccurlyeq \mathbf{Y}_{\Omega_-}. \end{array} \right\}. \tag{4}$$

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

$$\hat{\mathbf{X}}_{jk} = \begin{cases} \mathbf{Y}_{jk} & \text{if } jk \in \Omega_r; \\ (\mathbf{A}^H\mathbf{Z})_{jk} & \text{if } \begin{cases} jk \in \Omega_+, (\mathbf{A}^H\mathbf{Z})_{jk} \geq \tau_k; \\ \text{or} \\ jk \in \Omega_-, (\mathbf{A}^H\mathbf{Z})_{jk} \leq -\tau_k; \end{cases} \\ \operatorname{sgn}(\mathbf{Y}_{jk})\tau_k & \text{otherwise.} \end{cases} \tag{5}$$

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 $\mathbf{Y} \in \mathbb{R}^{J\times K}$ with $\mathbf{U} \in \mathbb{C}^{J\times K}$ the ADMM dual variable. In Algorithm 1 the μ 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 α $(0 < \alpha < 1)$. This way the algorithm relaxes the sparsity constraint while it progresses. Typical values for $\mu^{(0)}$ and α are given in section 4.1. The procedure is applied in a frame-based manner and outputs the $\hat{\mathbf{X}}_n$ declipped estimates, which are used to rebuild the full length estimated signal by means of overlap-and-add synthesis.

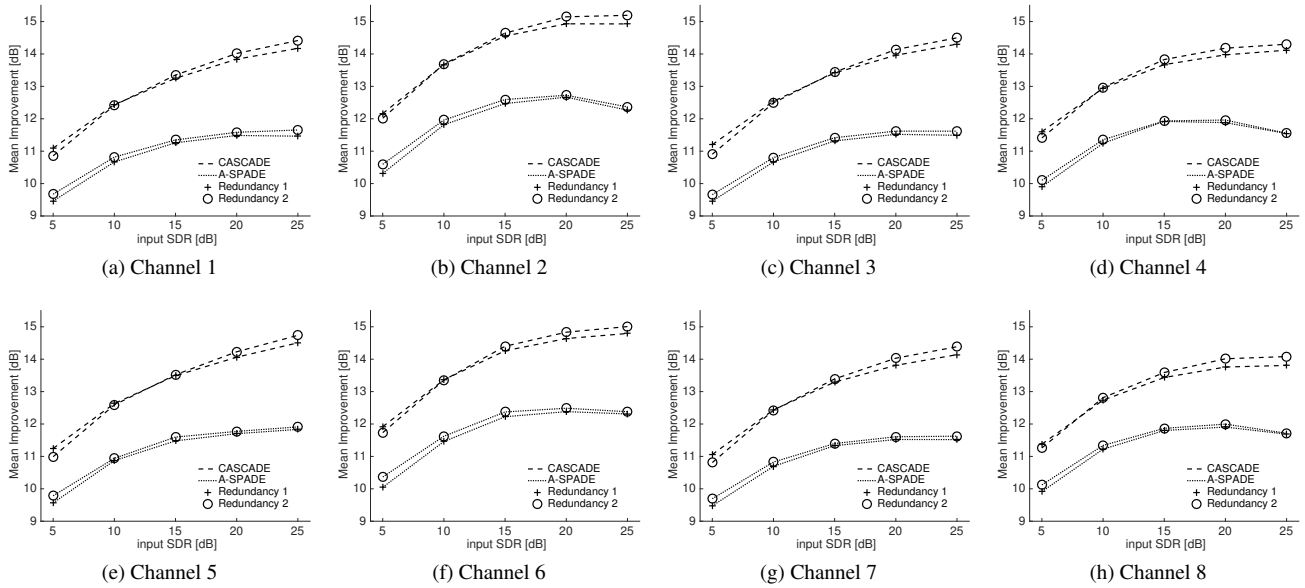
4. EXPERIMENTS

We perform experiments on 8 channels recordings excerpts from the VoiceHome2 Corpus [14][2]. We use the 359 clean speech available

[2] http://voice-home.gforge.inria.fr/voiceHome-2_corpus.html

Table 1. CASCADE parameters

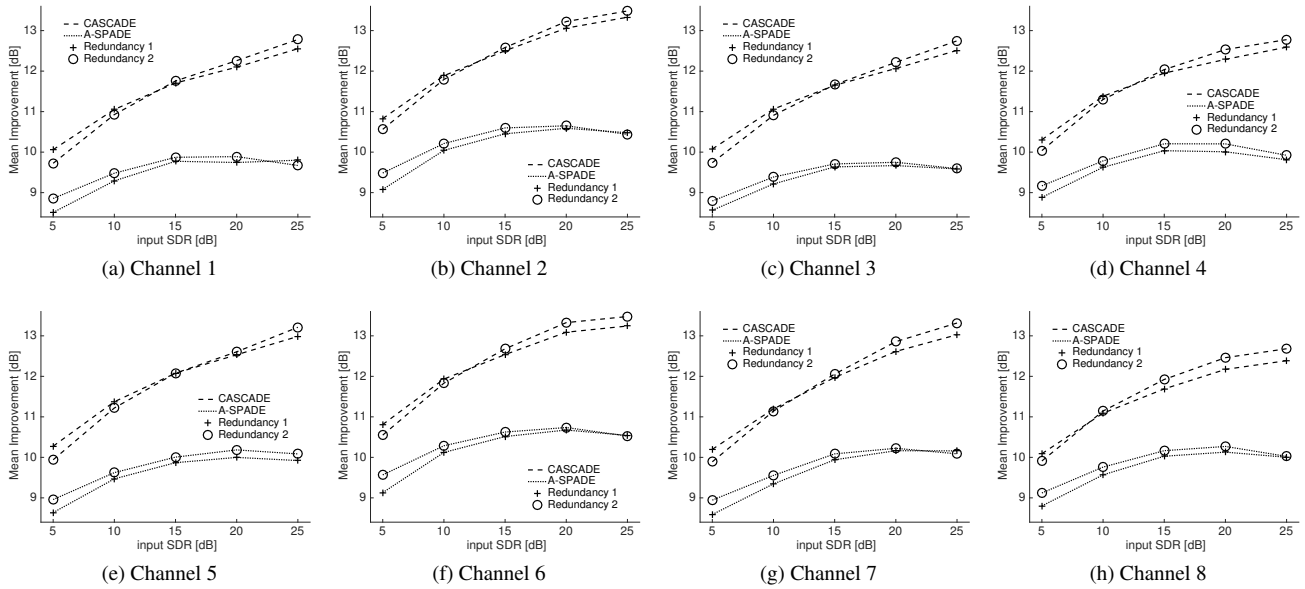
Parameters	Window size [samples]	Overlap [%]	Window Type	Channel number	Maximum iterations	Accuracy	Analysis operator
Value	$\mathsf{J} = 1024$	75	Hamming	$\mathsf{K} = 8$	$\mathsf{i}_{\max} = 10^6$	$\beta = 10^{-3}$	$\mathbf{A} = \mathrm{DFT}$



(a) Channel 1 (b) Channel 2 (c) Channel 3 (d) Channel 4

(e) Channel 5 (f) Channel 6 (g) Channel 7 (h) Channel 8

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



(a) Channel 1 (b) Channel 2 (c) Channel 3 (d) Channel 4

(e) Channel 5 (f) Channel 6 (g) Channel 7 (h) Channel 8

Fig. 3. Speech&Music declipping numerical results: SDR improvement [dB]

Algorithm 1 CASCADE algorithm

Require: $\mathbf{A}, \mathbf{Y}, \Gamma, \alpha, \mu^{(0)}, \Theta, \beta, \mathsf{i}_{\max}$

Initialization step:

$\hat{\mathbf{X}}^{(0)} = \mathbf{Y}, \mathbf{U}^{(0)} = 0, \mathsf{i} = 1$

Projection step on the modeling constraint:

$\mathbf{Z}^{(\mathsf{i})} = \mathcal{S}_{\mu^{(\mathsf{i}-1)}}(\mathbf{A}\hat{\mathbf{X}}^{(\mathsf{i}-1)} + \mathbf{U}^{(\mathsf{i}-1)})$

Projection step on declipping constraint:

$$\hat{\mathbf{X}}^{(\mathsf{i})} = \underset{\mathbf{X}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{X} - \mathbf{Z}^{(\mathsf{i})} + \mathbf{U}^{(\mathsf{i}-1)}\|_{\mathrm{F}}^2 \quad \text{subject to } \mathbf{X} \in \Theta$$

Update step:

$\mu^{(\mathsf{i})} = \alpha\mu^{(\mathsf{i}-1)}$

if $\frac{\|\mathbf{A}\mathbf{X}^{(\mathsf{i})} - \mathbf{Z}^{(\mathsf{i})}\|_{\mathrm{F}}}{\|\mathbf{A}\mathbf{X}^{(\mathsf{i})}\|_{\mathrm{F}}} \leq \beta$ or $\mathsf{i} \geq \mathsf{i}_{\max}$ **then**
  terminate
else
  $\mathbf{U}^{(\mathsf{i})} = \mathbf{U}^{(\mathsf{i}-1)} + \mathbf{A}\hat{\mathbf{X}}^{(\mathsf{i})} - \mathbf{Z}^{(\mathsf{i})}$
  $\mathsf{i} \leftarrow \mathsf{i} + 1$
end if
return $\hat{\mathbf{X}}^{(\mathsf{i})}$

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 $\mathbf{A} \in \mathbb{C}^{\mathsf{P}\times\mathsf{J}}$ 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 $\mathsf{P} = \mathsf{RJ}$). A first pilot study (data not shown) allowed us to choose the best parameters α 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 declipper (which uses a simple cosparse 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 presentend 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

Algorithm		CASCADE		A-SPADE	
Redundancy		R=1	R=2	R=1	R=2
Input SDR	5	167	398	73	190
	10	120	265	59	148
	15	80	177	42	103
	20	54	119	29	72
	25	37	78	20	50

(a) Runtime performance (ratio to realtime processing)

Algorithm		CASCADE		A-SPADE	
Redundancy		R=1	R=2	R=1	R=2
Input SDR	5	11.11	10.76	9.31	9.63
	10	12.39	12.45	10.57	10.79
	15	13.31	13.39	11.20	11.37
	20	14.01	14.32	11.67	11.79
	25	14.40	14.44	11.73	11.68

(b) Corresponding improvements (ΔSDR)

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 3a shows runtime performances and Table 3b the corresponding SDR improvements (ΔSDR) 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 cosparse prior with structure in the frequency-channel domain to address the multichannel audio declipping problem. We showed that adding across-channel structure on top of cosparse modeling was bringing considerable reconstruction improvements compared to a 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

[1] A. Janssen, R. Veldhuis, and L. Vries, "Adaptive interpolation of discrete-time signals that can be modeled as autoregressive processes," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 34, no. 2, pp. 317–330, 1986.

[2] A. Adler, V. Emiya, M. G. Jafari, M. Elad, R. Gribonval, and M. D. Plumbley, "Audio inpainting," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 3, pp. 922–932, 2012.

[3] S. Kitić, L. Jacques, N. Madhu, M. P. Hopwood, A. Spriet, and C. De Vleeschouwer, "Consistent Iterative Hard Thresholding for signal declipping," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013*. IEEE, 2013, pp. 5939–5943.

[4] S. Kitić, N. Bertin, and R. Gribonval, "Sparsity and cosparsity for audio declipping: a flexible non-convex approach," in *Latent Variable Analysis and Signal Separation (LVA/ICA)*, pp. 243–250. Springer, Liberec, Czech Republic, 2015.

[5] S. Kitić, N. Bertin, and R. Gribonval, "Audio declipping by cosparse hard thresholding," in *iTwist-2nd international-Traveling Workshop on Interactions between Sparse models and Technology*, Namur, Belgium, 2014.

[6] K. Siedenburg, M. Kowalski, and M. Dorfler, "Audio declipping with social sparsity," in *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 1577–1581.

[7] A. Ozerov, C. Bilen, and P. Perez, "Multichannel audio declipping," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'16)*, Shanghai, China, Mar. 2016.

[8] J. A. Tropp, A. C. Gilbert, and M. J. Strauss, "Algorithms for simultaneous sparse approximation: Part i: Greedy pursuit," *Signal Process.*, vol. 86, no. 3, pp. 572–588, Mar. 2006.

[9] R. Gribonval, H. Rauhut, K. Schnass, and P. Vandergheynst, "Atoms of all channels, unite! average case analysis ofmulti-channel sparse recovery using greedy algorithms," *Journal of Fourier Analysis and Applications*, vol. 14, no. 5, pp. 655–687, Dec 2008.

[10] R. Jenatton, J. Audibert, and F. Bach, "Structured variable selection with sparsity-inducing norms," *The Journal of Machine Learning Research*, vol. 12, pp. 2777–2824, 2011.

[11] M. Kowalski, K. Siedenburg, and M. Dorfler, "Social sparsity! neighborhood systems enrich structured shrinkage operators," *IEEE Transactions on Signal Processing*, vol. 61, no. 10, pp. 2498–2511, 2013.

[12] C. Févotte and M. Kowalski, "Hybrid sparse and low-rank time-frequency signal decomposition," in *23rd European Signal Processing Conference (EUSIPCO)*. IEEE, 2015, pp. 464–468.

[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[14] N. Bertin, E. Camberlein, R. Lebarbenchon, E. Vincent, S. Sivasankaran, I. Illina, and F. Bimbot, "VoiceHome-2, an extended corpus for multichannel speech processing in real homes," *To appear in Speech Communication*, 2017.