BLASTER

An off-grid method for blind and regularized acoustic echoes retrieval

---

Diego Di Carlo, Clément Elvira, Antoine Deleforge, Nancy Bertin, Rémi gribonval
April 17, 2020

IEEE ICASSP 2020

Introduction

Introduction

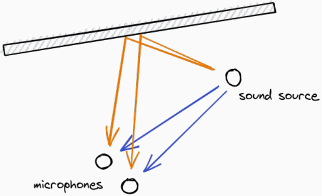
Proposed Approach

Results

Echoes help acoustic processing

Audio Speech Signal Processing

- **suffers** in real non-anechoic environments
- **early reflections** and **reverberation**
  - … breaks the *free-field* assumption
  - … are considered as *foes*



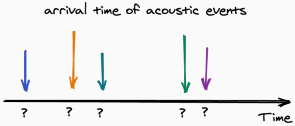
Echo-aware Audio Processing turns them into friends

- for **speech enhancement**
  [Ribeiro et al., 2010, Dokmanić et al., 2015, Scheibler et al., 2018]
- for 3D **room geometry estimation** from sound
  [Antonacci et al., 2012, Dokmanić et al., 2015, Crocco et al., 2017]

The acoustic echoes retrieval (AER) problem

Estimating early (strong) acoustic reflections:

- their time of arrivals → TOAs Estimation
- their amplitude



We consider the scenario

1. BLIND: Source signal is unknown
2. SIMO: Single input and multiple outputs (here only stereophonic recordings)

Room Impulse Response, h_i

The linear filtering effect due to the propagation of sound from a source to a microphone in a indoor space

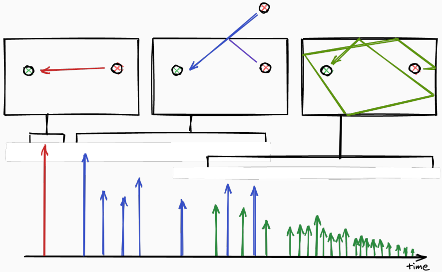
$$x_i(t) = (h_i * s)(t) + n_i(t)$$

with Image Source Model:

as stream of Diracs:

$$h_i(t) = \sum_{r=0}^{R} \alpha_{i,r} \delta(t - \tau_{i,r})$$

$$\Longleftrightarrow$$



Key ingredient – *Cross relation identity*

$$x_i = h_i * s$$

$$h_2 * x_1 = h_2 * h_1 * s = h_1 * h_2 * s = h_1 * x_2$$

Ideas

1. Sampled version of x_1, x_2 are available $(\mathbf{x}_1, \mathbf{x}_2)$
2. Assume echoes belong to multiples of the sampling frequency
3. Identify echoes $\rightarrow$ find sparse vectors $\mathbf{h}_1, \mathbf{h}_2$
4. Lasso-like problem

$$\widehat{\mathbf{h}}_1, \widehat{\mathbf{h}}_2 \in \underset{\mathbf{h}_1, \mathbf{h}_2 \in \mathbb{R}^n}{\arg\min} \|\mathbf{x}_1 * \mathbf{h}_2 - \mathbf{x}_2 * \mathbf{h}_1\|_2^2 + \lambda Reg(\mathbf{h}_1, \mathbf{h}_2)$$

$Reg(\mathbf{h}_1, \mathbf{h}_2) \longrightarrow$ sparse promoting regularizer

✓ [Lin et al., 2007] ✓ [Aïssa-El-Bey and Abed-Meraim, 2008]
✓ [Kowalczyk et al., 2013] ✓ [Crocco and Del Bue, 2015]

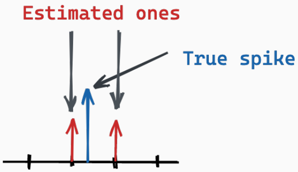
Limitations / bottleneck

Limitations

- Echoes are not necessarily "on grid"
- *Body guard* effect [Duval and Peyré, 2017]

  $\longrightarrow$ low recall $\Longrightarrow$ low accuracy

  $\longrightarrow$ slow convergence



Increase the sampling frequency, F_s

$\longrightarrow$ Increase Precision

Computational bottleneck

- Bigger vectors and matrices

  $\longrightarrow$ memory usage

- Computational complexity: at best $\mathcal{O}(F_s^2)$ per iteration
- the higher the sampling frequency, the more ill-conditioned

  $\longrightarrow$ slow convergence

State of the Art approach

State Of The Art

1. discrete (sparse)
   Blind Channel Estimation
   (BCE)
2. Peak-picking

State Of The Art

1. discrete (sparse)
   Blind Channel Estimation
   (BCE)
2. Peak-picking

$\Longrightarrow$ however

- Full channel
  so lot of memory
- Echoes are “off-grid”

State of the Art approach

State Of The Art

1. discrete (sparse) Blind Channel Estimation (BCE)
2. Peak-picking

$\Longrightarrow$ **we propose**

1. BCE + Continuous Dictionary
2. Greedy-like approach
3. Inputs:
   - mic recordings
   - # echoes

$\Longrightarrow$ **however**

- Full channel so lot of memory
- Echoes are “off-grid”

Acoustic Echoes Retrieval as off-grid Spike Retrieval Problem

Introduction

Proposed Approach

Results

Off-the-grid acoustic echo retrieval

Observation 1: the cross relation remains true in the frequency domain

$$\mathcal{F}x_1 \cdot \mathcal{F}h_2(^n/_{F_s}) = \mathcal{F}x_2 \cdot \mathcal{F}h_1(^n/_{F_s}) \qquad n = 0 \dots N-1$$

Observation 2: $\mathcal{F}\delta_{\mathrm{echo}}$ is known in closed-form

Observation 3: $\mathcal{F}\mathrm{x_i}$ can be (well) approximated by DFT

$$\mathsf{X}_i = DFT(\mathsf{x}_i) \simeq \mathcal{F}\mathsf{x}_i(nF_s) \qquad n = 0 \dots N-1$$

Idea: Recover echoes by matching a finite number of frequencies

$$\underset{h_1,h_2 \in \substack{\text{measure}\\ \text{space}}}{\text{arg min}} \ \frac{1}{2}\|\mathsf{X}_1 \cdot \mathcal{F}h_2(f) - \mathsf{X}_2 \cdot \mathcal{F}h_1(f)\|_2^2 + \lambda\|h_1 + h_2\|_{\mathrm{TV}} \quad \text{s.t.} \begin{cases} h_1(\{0\}) = 1 \\ h_l \geq 0 \end{cases}$$

Instance of a **BLasso** problem [Bredies and Pikkarainen, 2013]

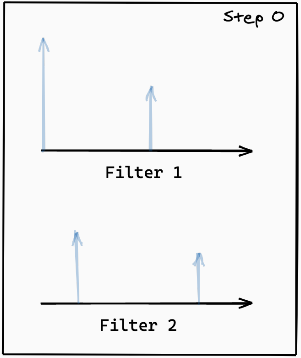
✓ no Toeplitz matrix

✓ Solutions is a train of Dirac

✓ anchor prevents trivial solution

Algorithm

Problem is **convex** with respect to the filters h_1 and h_2
$\longrightarrow$ Sliding Frank-Wolfe algorithm [Denoyelle et al., 2019]

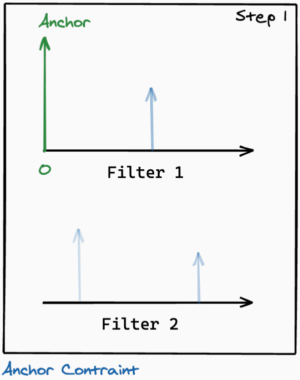


Initialization

Problem is **convex** with respect to the filters h_1 and h_2

$\longrightarrow$ Sliding Frank-Wolfe algorithm [Denoyelle et al., 2019]

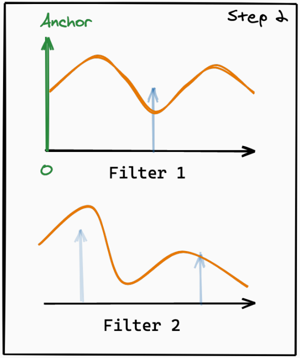
1. Start from the anchor



Algorithm

Problem is **convex** with respect to the filters h_1 and h_2
$\longrightarrow$ Sliding Frank-Wolfe algorithm [Denoyelle et al., 2019]

1. Start from the anchor
2. Compute the *local* cost based on Cross-relation

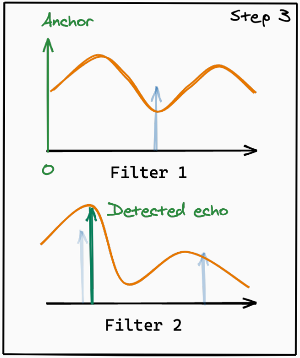


Local Const

Algorithm

Problem is **convex** with respect to the filters h_1 and h_2
$\longrightarrow$ Sliding Frank-Wolfe algorithm [Denoyelle et al., 2019]

1. Start from the anchor
2. Compute the *local* cost based on Cross-relation
3. Find the maximizer

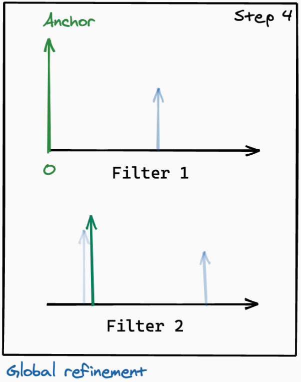


Detected echo

Algorithm

Problem is **convex** with respect to the filters h_1 and h_2
$\longrightarrow$ Sliding Frank-Wolfe algorithm [Denoyelle et al., 2019]

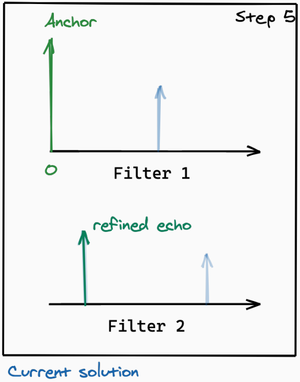
1. Start from the anchor
2. Compute the *local* cost based on Cross-relation
3. Find the maximizer
4. Update weight (Lasso-like)



Algorithm

Problem is **convex** with respect to the filters h_1 and h_2
$\longrightarrow$ Sliding Frank-Wolfe algorithm [Denoyelle et al., 2019]

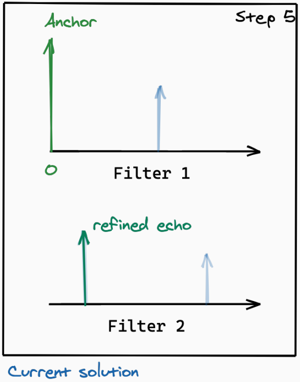
1. Start from the anchor
2. Compute the *local* cost based on Cross-relation
3. Find the maximizer
4. Update weight (Lasso-like)
5. Joint refinement



Problem is **convex** with respect to the filters h_1 and h_2
$\longrightarrow$ Sliding Frank-Wolfe algorithm [Denoyelle et al., 2019]

1. Start from the anchor
2. Compute the *local* cost based on Cross-relation
3. Find the maximizer
4. Update weight (Lasso-like)
5. Joint refinement

Repeat until optimality conditions are met



Numerical Results

Introduction

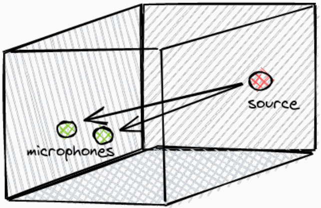
Proposed Approach

Results

Experimental Setup

Condition

- 2 microphones, 1 sound source
- Shoebox with random dimension
- 2 signals: broadband and speech
- 2 dataset: $\mathcal{D}^{\text{SNR}}$, $\mathcal{D}^{\text{RT60}}$
  - $\mathcal{D}^{\text{SNR}}$: $SNR \in [0, 20]$ dB, $\text{RT}_{60} = 400$ ms
  - $\mathcal{D}^{\text{RT60}}$: $\text{RT}_{60} = [100, 1000]$ ms, $SNR = 20$ dB



Considered Methods

- BSN: Blind Sparse and Non-negative BCE [Lin et al., 2007]

$$\underset{\mathbf{h}=[\mathbf{h}_1,\mathbf{h}_2]}{\arg\min} \ \|\mathcal{T}(\mathbf{x}_1)\mathbf{h}_2 - \mathcal{T}(\mathbf{x}_2)\mathbf{h}_1\|_2^2 + \lambda\|\mathbf{h}\|_1 \quad \text{s.t.} \quad \mathbf{h}[0] = 1, \mathbf{h} \geq 0$$

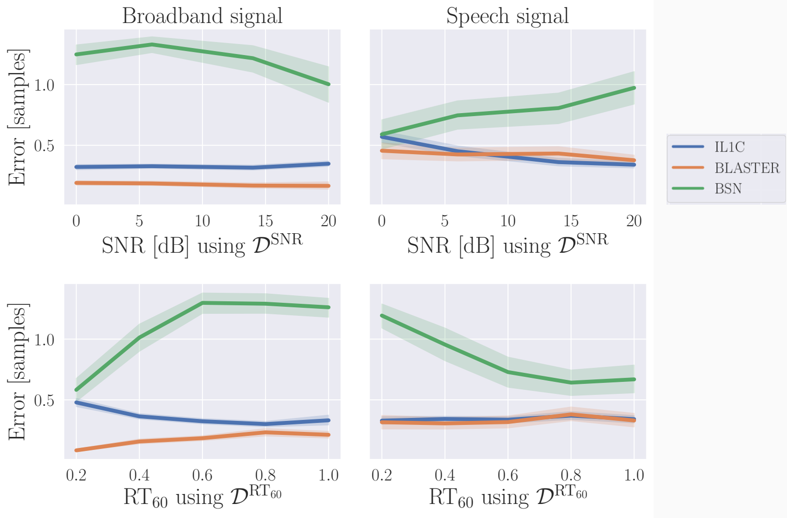
- IL1C: Iterative ℓ_1 Constraint BCE [Crocco and Del Bue, 2015]

$$\underset{\mathbf{h}=[\mathbf{h}_1,\mathbf{h}_2]}{\arg\min} \ \|\mathcal{T}(\mathbf{x}_1)\mathbf{h}_2 - \mathcal{T}(\mathbf{x}_2)\mathbf{h}_1\|_2^2 + \|\mathbf{h}\|_1 \quad \text{s.t.} \quad \mathbf{h}^\top \mathbf{p}^{(z)} = 1, \mathbf{h} \geq 0$$

- BLASTER: Off-grid BCE

$$\underset{h_1,h_2 \in \text{measure}}{\arg\min} \ \|X_1 \cdot \mathcal{F}h_2(f) - X_2 \cdot \mathcal{F}h_1(f)\|_2^2 + \lambda\|h_1 + h_2\|_{\mathrm{TV}} \quad \text{s.t.} \quad h_1(\{0\}) = 1, h_l \geq 0$$

Error per Dataset/Signal while recovering 7 echoes



✓Lower RMSE

✓ Robustness to SNR and RT_{60}

✗ Source signal dependent

Precision per threshold in typical scenario

	Precision [%]									
	R = 2 echoes					R = 7 echoes				
τ_{thr} [samples]	0.5	1	2	3	10	0.5	1	2	3	10
BSN	8	9	27	46	62	5	8	38	54	73
IL1C	51	55	55	56	58	42	53	55	56	58
BLASTER	**68**	**73**	**74**	**75**	**75**	46	53	56	57	61

Table 1: $RT_{60} = 200$ ms and SNR = 20 dB.

✓ Invariant
to threshold

✗ Sensitive
to # echoes

Performance per # of echoes

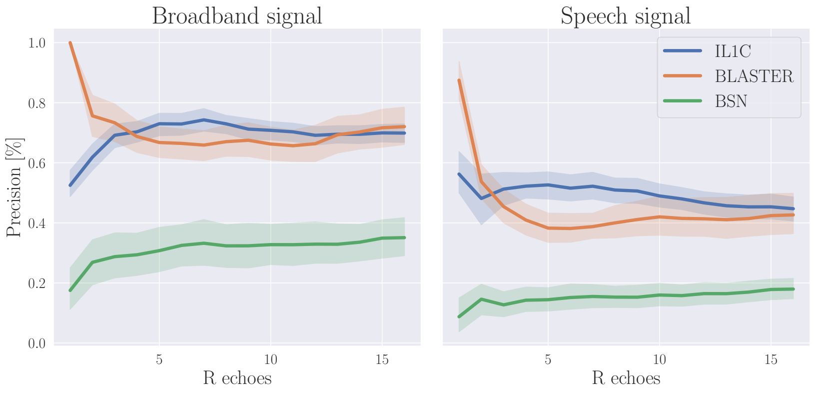


Figure 1: $RT_{60} = 400$ ms and SNR = 20 dB.

✗ Sensitive to # echoes

✗ Sensitive source signal

✓ Good for 2 echoes

Performance per # of echoes

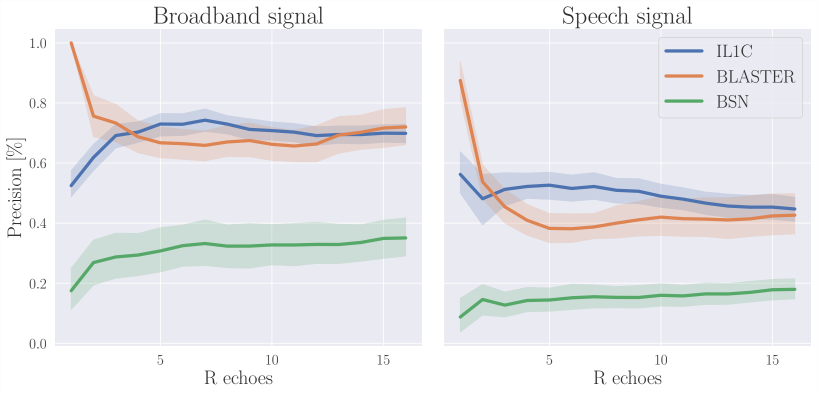


Figure 1: $RT_{60} = 400$ ms and SNR = 20 dB.

✗ Sensitive to # echoes

✗ Sensitive source signal

✓ Good for 2 echoes [Di Carlo et al., 2019, Scheibler et al., 2018]

Conclusion

1. Introduction
    - Echoes helps indoor processing
    - On-grid method suffer of pathological problem when off-grid problem
2. BLASTER
    - Super resolution can be applied to SIMO BCE
    - Dirac modeled in closed-from
3. Experiments
    - Smaller RMSE due to super-resolution
    - Better performances for smaller # echoes
    - Performances are source-dependent

Future Work

- Extension to multichannel recording
- Test on real data recordings

Thank you!

https://gitlab.inria.fr/panama-team/blaster

Aïssa-El-Bey, A. and Abed-Meraim, K. (2008).
Blind SIMO channel identification using a sparsity criterion.
IEEE Workshop on Signal Processing Advances in Wireless Communications, SPAWC, pages 271–275.

Antonacci, F., Filos, J., Thomas, M. R., Habets, E. A., Sarti, A., Naylor, P. A., and Tubaro, S. (2012).
Inference of room geometry from acoustic impulse responses.
IEEE Transactions on Audio, Speech and Language Processing, 20(10):2683–2695.

Bredies, K. and Pikkarainen, H. K. (2013).
Inverse problems in spaces of measures.
ESAIM: Control, Optimisation and Calculus of Variations, 19(1):190–218.

Crocco, M. and Del Bue, A. (2015).
Room impulse response estimation by iterative weighted L1-norm.
In *Proc. Europ. Sig. Proces. Conf.*, pages 1895–1899.

Crocco, M., Trucco, A., and Del Bue, A. (2017).
Uncalibrated 3D room geometry estimation from sound impulse responses.
Journal of the Franklin Institute, 354(18):8678–8709.

Denoyelle, Q., Duval, V., Peyré, G., and Soubies, E. (2019).
The Sliding Frank-Wolfe Algorithm and its Application to Super-Resolution Microscopy.
Inverse Problems.

Di Carlo, D., Deleforge, A., and Bertin, N. (2019).
Mirage: 2d source localization using microphone pair augmentation with echoes.
In *Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proces.*, pages 775–779.

Dokmanić, I., Scheibler, R., and Vetterli, M. (2015).
Raking the Cocktail Party.
IEEE Journal on Selected Topics in Signal Processing, 9(5):825–836.

Duval, V. and Peyré, G. (2017).
Sparse Regularization on Thin Grids I: the LASSO.
Inverse Problems, 33(5).

Kowalczyk, K., Habets, E., Kellermann, W., and Naylor, P. (2013).
Blind system identification using sparse learning for TDOA estimation of room reflections.
IEEE Signal Processing Letters, 20(7):653–656.

Lin, Y., Chen, J., Kim, Y., and Lee, D. (2007).
Blind sparse-nonnegative (BSN) channel identification for acoustic time-difference-of-arrival estimation.
IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, pages 106–109.

Ribeiro, F., Ba, D., Zhang, C., and Florêncio, D. (2010).
Turning enemies into friends: Using reflections to improve sound source localization.
In *Proc. IEEE Int. Conf. on Multimedia and Expo*, pages 731–736.

Scheibler, R., Bezzam, E., and Dokmanic, I. (2018).
Pyroomacoustics: A python package for audio room simulation and array processing algorithms.
Proc. IEEE Int. Conf. Acoust., Speech, and Signal Proces.