Recent developments on

FACTORSYNTH

Machine learning for sound deconstruction

Juan José Burred

Independent researcher and developer, Paris, France

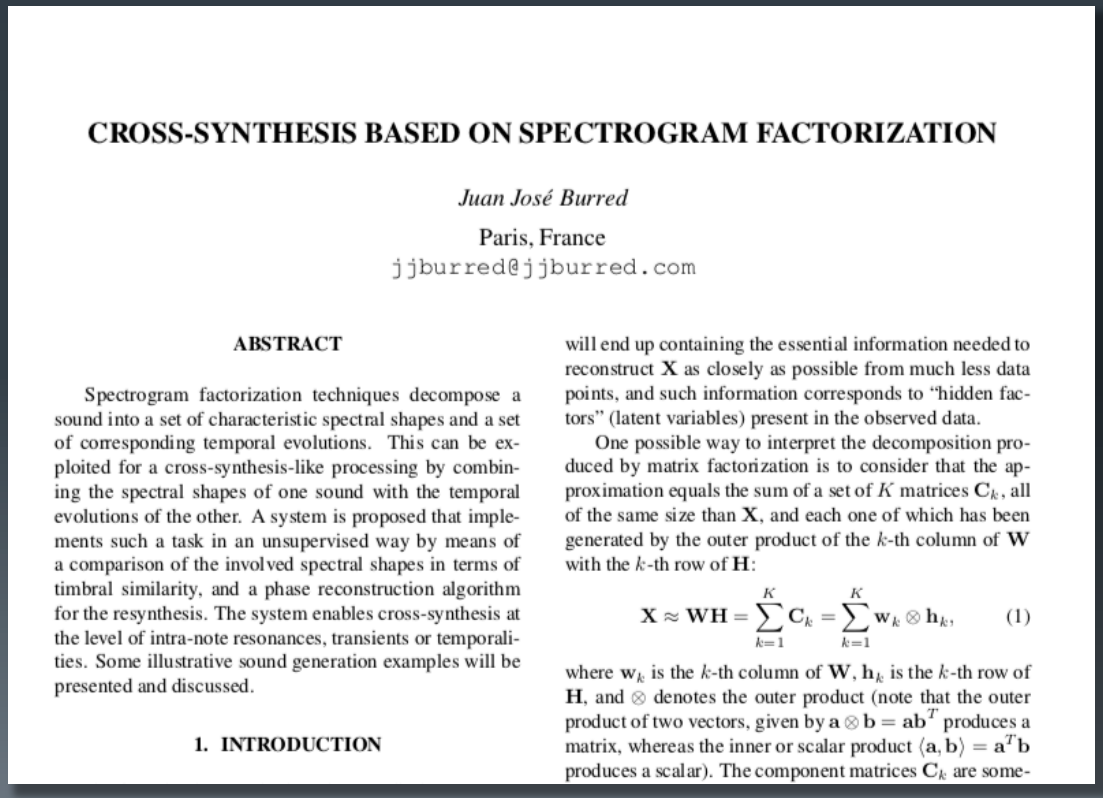
IRCAM Forum Workshops, Paris

March 28th 2019

Factorsynth timeline

2013 research —— 2014 command-line —— 2016 Max —— 2018 Max For Live

Original idea: use of a data analysis / source separation technique (matrix factorization) as a new tool for composition and sound design



CROSS-SYNTHESIS BASED ON SPECTROGRAM FACTORIZATION

Juan José Burred
Paris, France
jjburred@jjburred.com

ABSTRACT

Spectrogram factorization techniques decompose a sound into a set of characteristic spectral shapes and a set of corresponding temporal evolutions. This can be exploited for a cross-synthesis-like processing by combining the spectral shapes of one sound with the temporal evolutions of the other. A system is proposed that implements such a task in an unsupervised way by means of a comparison of the involved spectral shapes in terms of timbral similarity, and a phase reconstruction algorithm for the resynthesis. The system enables cross-synthesis at the level of intra-note resonances, transients or temporalities. Some illustrative sound generation examples will be presented and discussed.

1. INTRODUCTION

will end up containing the essential information needed to reconstruct $\mathbf{X}$ as closely as possible from much less data points, and such information corresponds to "hidden factors" (latent variables) present in the observed data.

One possible way to interpret the decomposition produced by matrix factorization is to consider that the approximation equals the sum of a set of K matrices $\mathbf{C}_k$, all of the same size than $\mathbf{X}$, and each one of which has been generated by the outer product of the k-th column of $\mathbf{W}$ with the k-th row of $\mathbf{H}$:

$$\mathbf{X} \approx \mathbf{WH} = \sum_{k=1}^{K} \mathbf{C}_k = \sum_{k=1}^{K} \mathbf{w}_k \otimes \mathbf{h}_k, \qquad (1)$$

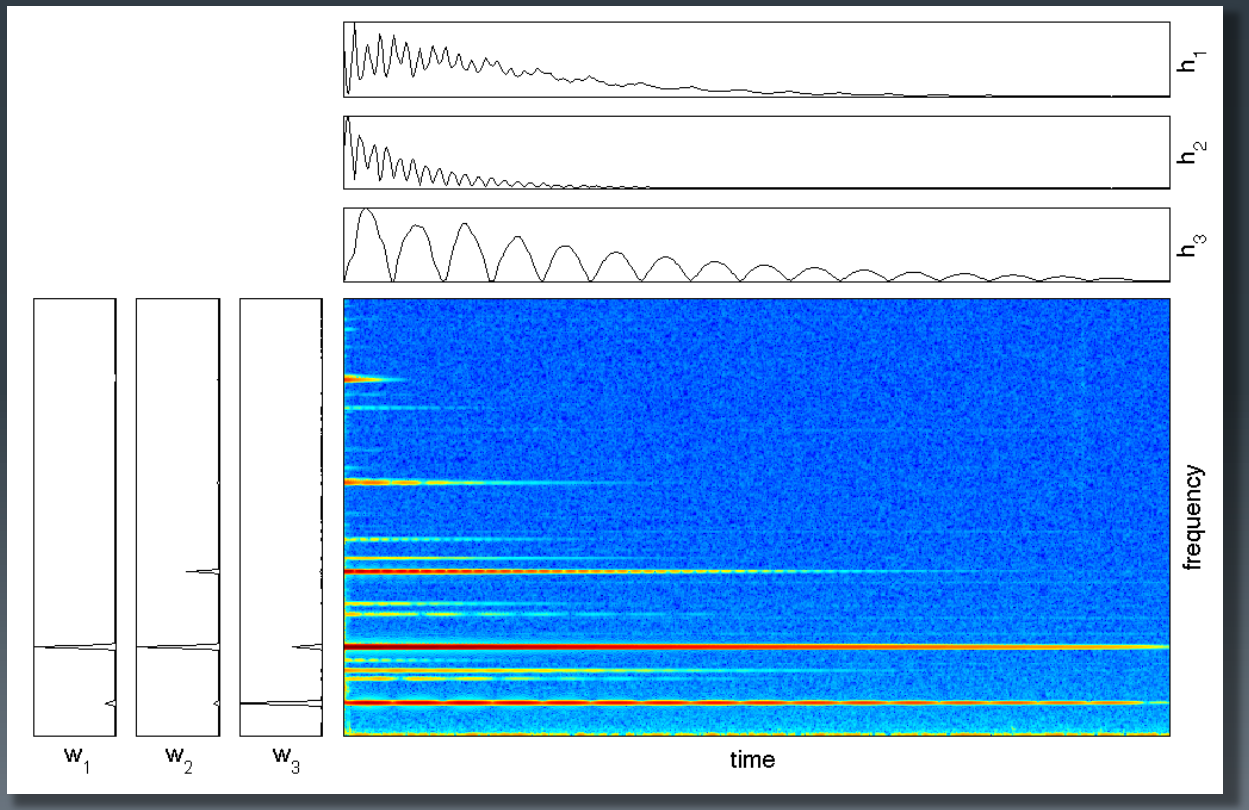
where $\mathbf{w}_k$ is the k-th column of $\mathbf{W}$, $\mathbf{h}_k$ is the k-th row of $\mathbf{H}$, and $\otimes$ denotes the outer product (note that the outer product of two vectors, given by $\mathbf{a} \otimes \mathbf{b} = \mathbf{a}\mathbf{b}^T$ produces a matrix, whereas the inner or scalar product $\langle \mathbf{a}, \mathbf{b} \rangle = \mathbf{a}^T\mathbf{b}$ produces a scalar). The component matrices $\mathbf{C}_k$ are some-

ICMC 2013

Factorsynth timeline

2013 research —— 2014 command-line —— 2016 Max —— 2018 Max For Live

Non-negative Matrix Factorization (NMF) applied to a spectrogram

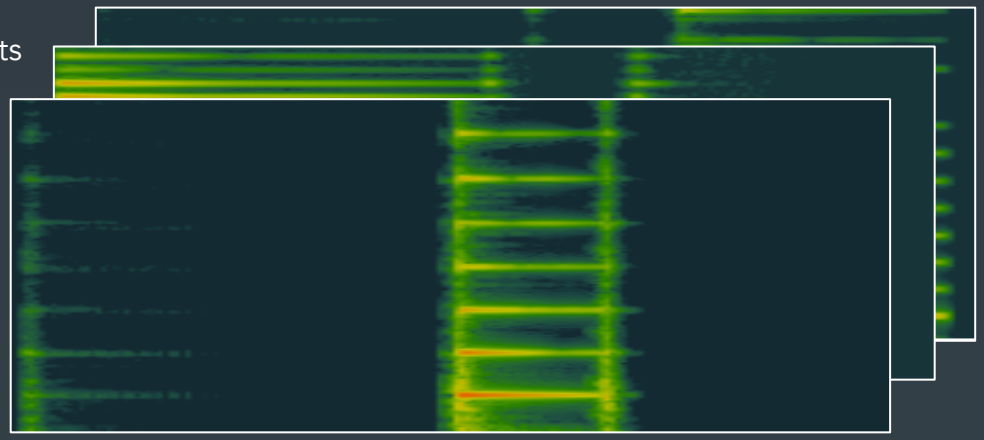


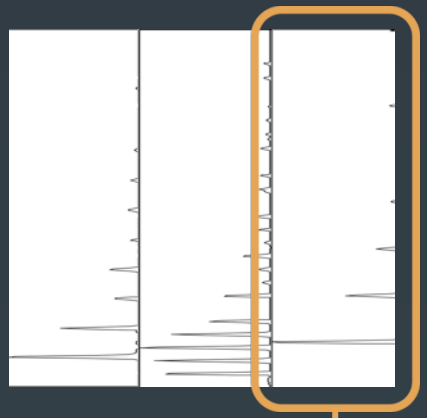
Factorsynth timeline

2013 research —— 2014 command-line —— 2016 Max —— 2018 Max For Live

Non-negative Matrix Factorization (NMF) applied to a spectrogram

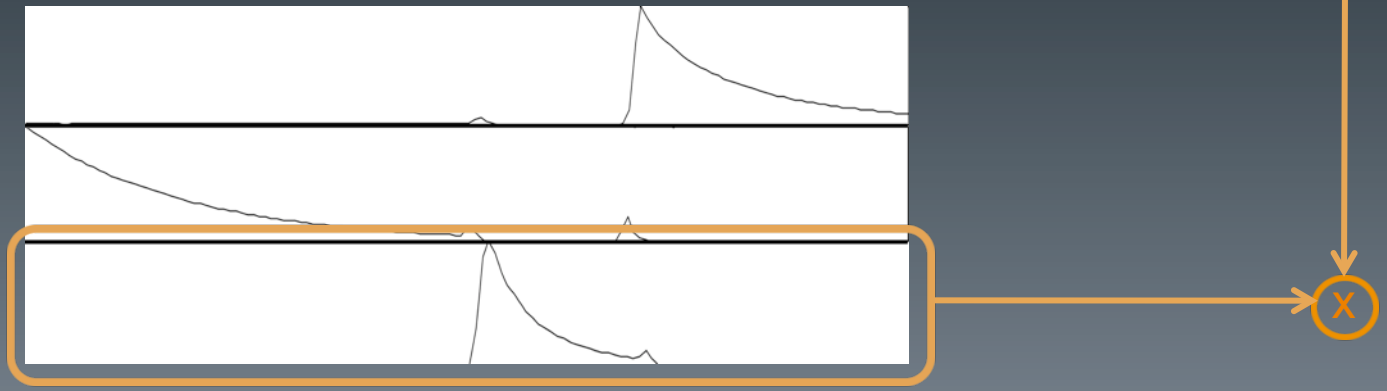
components





spectral shapes

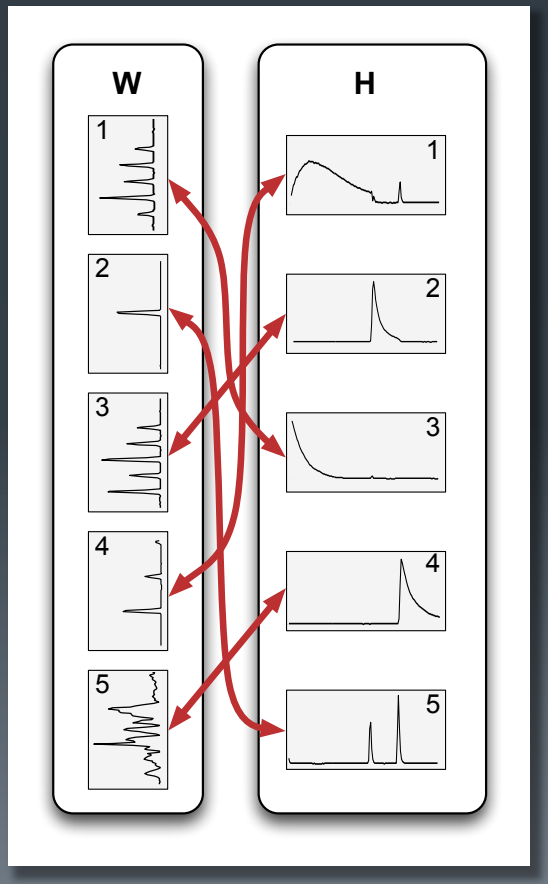
temporal shapes



Factorsynth timeline

2013 research —— 2014 command-line —— 2016 Max —— 2018 Max For Live

Free recombination of the temporal and spectral shapes



Factorsynth timeline

2013 research —— 2014 command-line —— 2016 Max —— 2018 Max For Live

Command-line prototype tool for Mac OS (still available at jjburred.com)

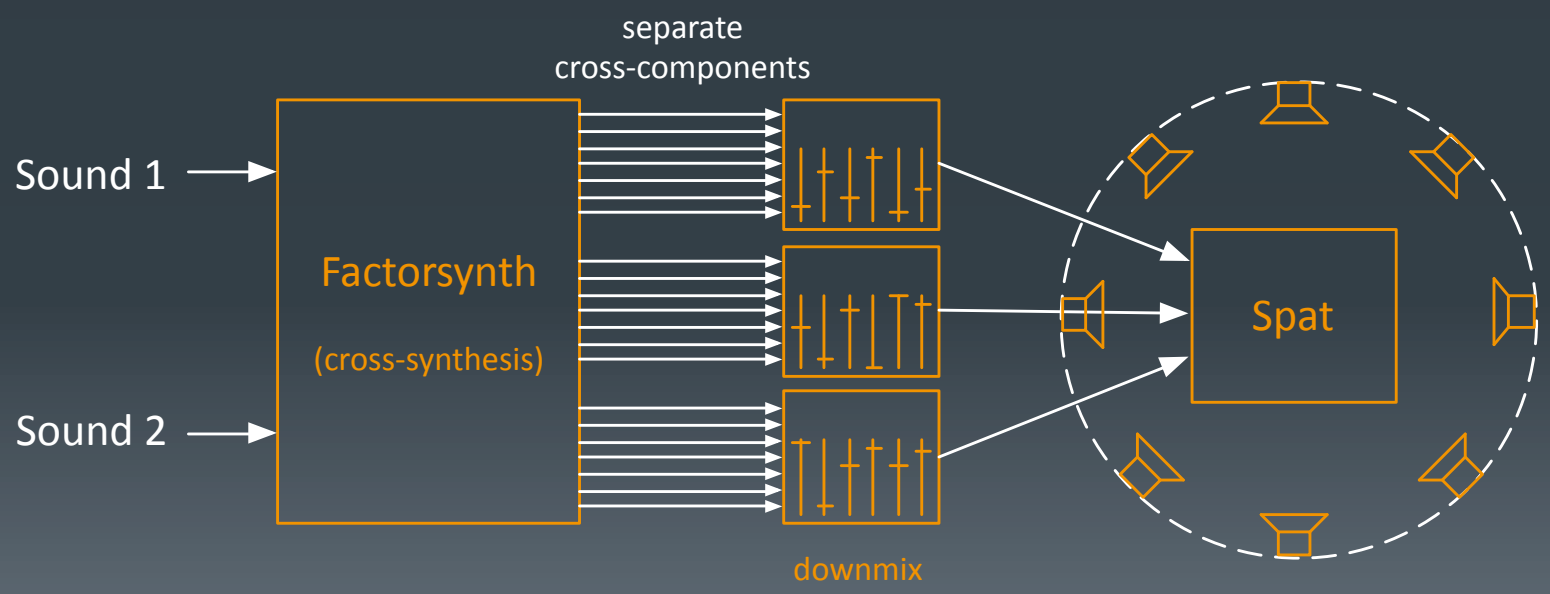
```
jjmac:factorsynth-v0.21 jjb$ ./factorsynth
factorsynth v0.21, 31/5/2016
J.J.Burred, http://jjburred.com
---
Usage:
        factorsynth -i input_file [-o output_file] [-t target_file] [-p operation] [-k numSourceCompone
nts] [-c numTargetComponents] [-f constrainedFactor] [-h]

Options:
        -i      Path to input sound file.
        -o      Path to output sound file (default: ./out.wav).
        -t      Path to target sound file for cross-synthesis (only when using option -p xsynth).
        -p      Operation to perform. Possible values:
                        scr             scramble (default)
                        cscr            constrained scramble
                        rank            inverse rank
                        xsynth          cross-synthesis
                        xsynth_inj      cross-synthesis with injective mapping
        -k      Number of components (default: 30).
        -c      Number of components for target (only for cross-synthesis) (default: 30).
        -f      Factor for constrained scramble (default: 0.8).
        -s      Output separated input and output components in folder './component_out'.

Example:
        factorsynth -i source.wav -t target.wav -p xsynth -k 30 -c 20
jjmac:factorsynth-v0.21 jjb$
```

Factorsynth timeline

2013 research —— 2014 command-line —— 2016 Max —— 2018 Max For Live

Example of use: generation of components for spatialization

Emanuele Palumbo: "Artaud Overdrive", IRCAM Manifeste 2016

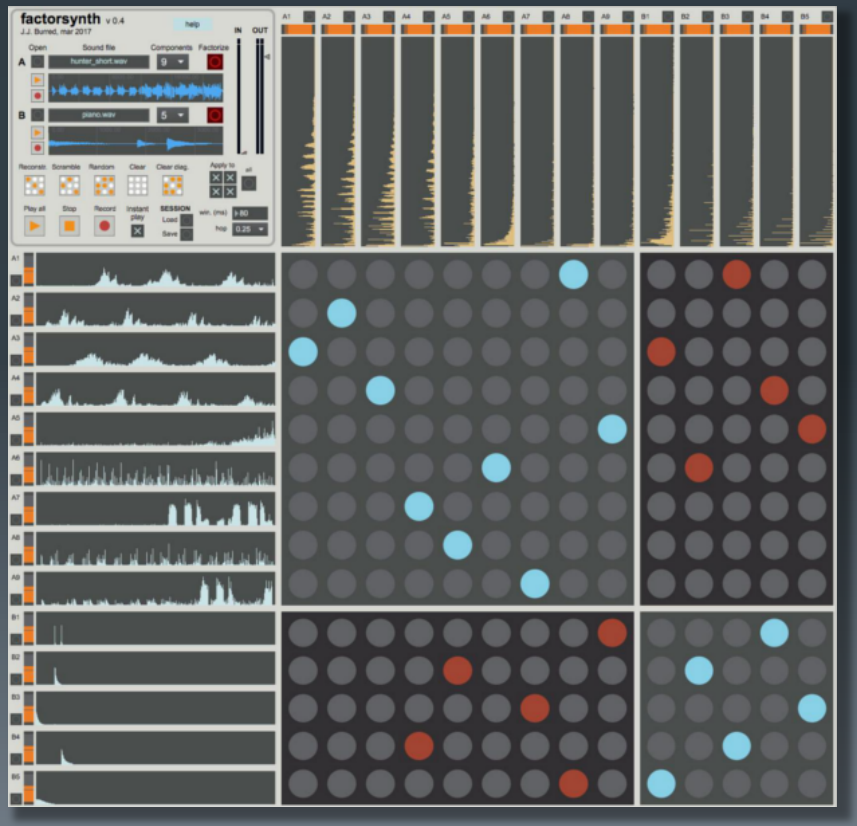


Factorsynth timeline

2013 research —— 2014 command-line —— 2016 Max —— 2018 Max For Live

Max prototype (Mac OS) (still available at jjburred.com)

Graphical interface for offline processing



Factorsynth timeline

2013 research —— 2014 command-line —— 2016 Max —— 2018 Max For Live

Examples of use:

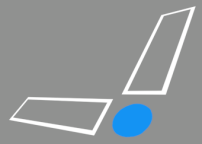
- component-level cross-synthesis of extended violin techniques
  Maurizio Azzan: “Each Mirror Infects Itself”, Le Dôme festival, Montbazon, July 2017

- component-level processing of extended cello techniques
  Maurizio Azzan: “Where the here and now of nowhere is”, IRCAM Manifeste 2018

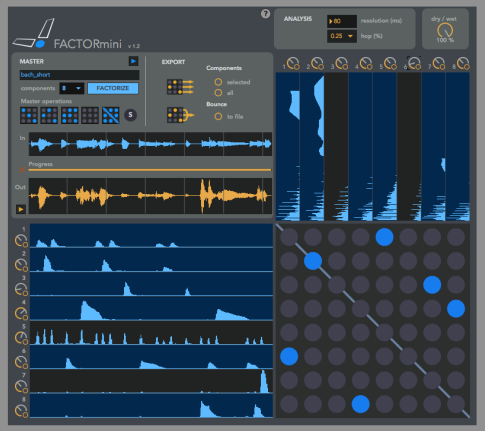
Factorsynth timeline

2013 research —— 2014 command-line —— 2016 Max —— 2018 Max For Live

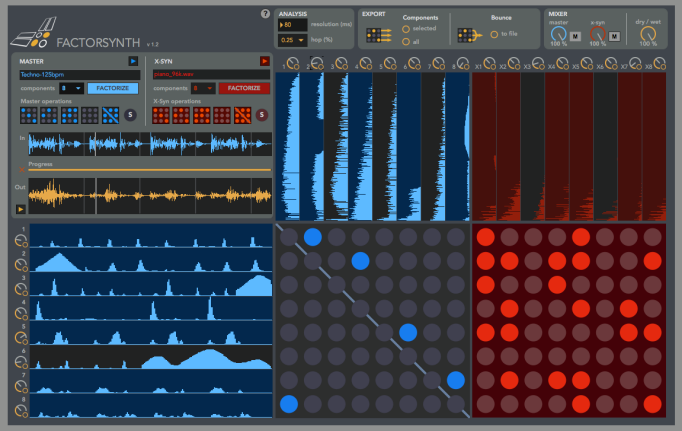
Max For Live commercial devices (available at jjburred.com)



FACTORmini



FACTORSYNTH



Factorsynth timeline

2013 research —— 2014 command-line —— 2016 Max —— 2018 Max For Live

New features of the Max For Live version:

- Real-time resynthesis engine
  - Components can be modified and recombined in real time
  - Aimed to use in performance
- Integration with Ableton Live
  - Synchronization with clip position and looping in arrangement and session views
  - Possible to process warped and transposed clips
  - Parameter automation and MIDI-mapping
- Support of control surfaces
- Mac and Windows versions

Factorsynth timeline

2013 research —— 2014 command-line —— 2016 Max —— 2018 Max For Live

Timeline of the Max For Live version:

- July 2018: first Mac OS release
- July 2018: member of Isotonik Collective
  (biggest reseller of commercial Max For Live devices)

- Sep. 2018: v.1.2: first cross-platform version (Mac+Windows)
- Oct. 2018: Release of FACTORmini
- Jan. 2019: Partner product of the IRCAM Forum
  - 25% off for premium members of the Forum (37 €)

- Feb. 2019: v1.4: support for control surfaces

Future developments

- Processing of real-time audio input
  - Application of spectral or temporal shapes to input stream
- Better algorithms for factorization and resynthesis
  - More control of algorithm parameters by user
- Online factorization
  - Extraction of the spectral and temporal shapes on the fly
- Multichannel output
- Custom Max versions for composers and sound installation artists
- General research scope: morphological sound analysis/synthesis
  - Other methods besides factorization (neural methods)
  - Constraint of interpretability and user-based component manipulation

Thanks for your attention!

jjburred.com

jjburred@jjburred.com