Audio Inpainting

Ondřej Mokrý, Pavel Rajmic

Brno University of Technology
Signal Processing Laboratory

March 28, 2023

Table of Contents

Table of Contents

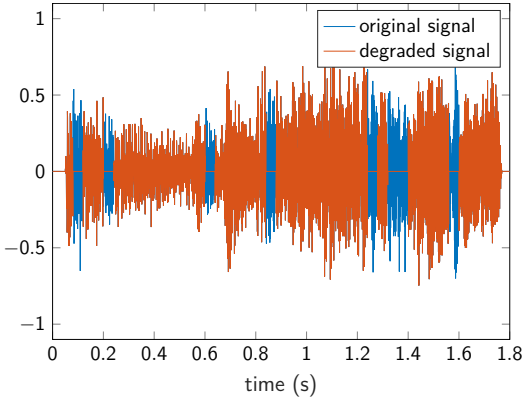
GitHub repository:

`https://github.com/ondrejmokry/InpaintingLecture`

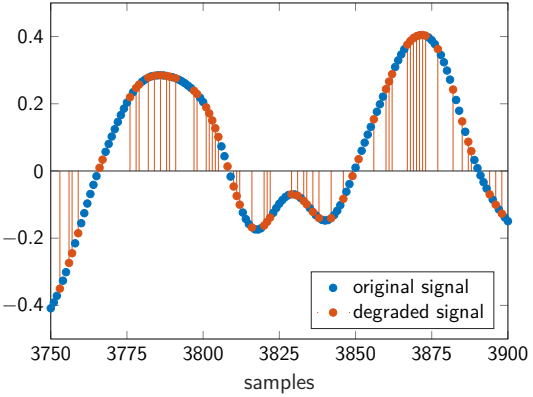
The problem of audio inpainting

The problem of audio inpainting

Time domain



(a) missing signal blocks



(b) missing random signal samples

Figure: Signal degraded by lost samples.

In case (a), 18 % of all samples are lost in blocks of length 40 ms. In case (b), 60 % of randomly selected samples are lost.

The problem of audio inpainting

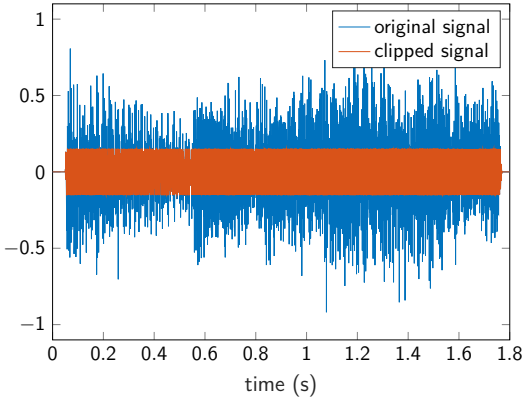
Typical situations

- Damaged medium (LP, CD, wax phonograph cylinders and any other historical media)
- Transmission error $\rightarrow$ *packet loss concealment*
- Impulsive noise $\rightarrow$ *declicking*
- Other problems (clipping) when treated naively
- Special case: upsampling / interpolation

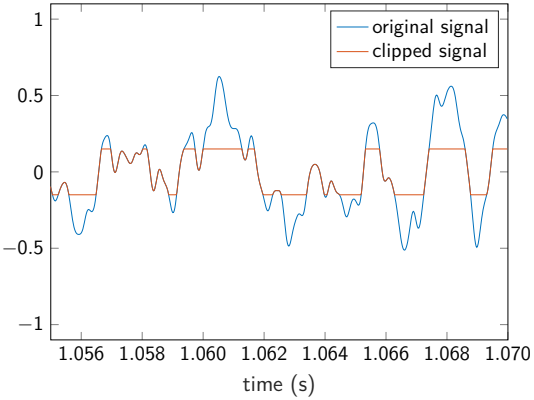
- Detection + restoration

- Image restoration $\rightarrow$ *inpainting*

Clipping



(a) the whole signal



(b) selection

Figure: Signal degraded by clipping.

55 % of all samples are clipped

- translation (τ) + modulation (ω) of the window function ($\mathbf{g}$)

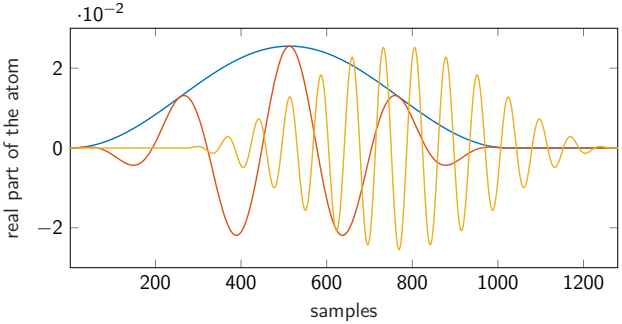


Figure: Atoms of the STFT (in discrete setting).

- Fourier transform in translated "snip-outs" of the signal ($\mathbf{x}$)

$$\mathbf{x}(t) \mapsto \mathbf{C}(\tau, \omega) = \langle \mathbf{x}, \mathbf{g}_{\tau,\omega} \rangle, \quad \mathbf{g}_{\tau,\omega}(t) = \mathbf{w}(t-\tau)\mathbf{e}^{\mathrm{i}\omega t}$$

Missing signal blocks

Time-frequency domain

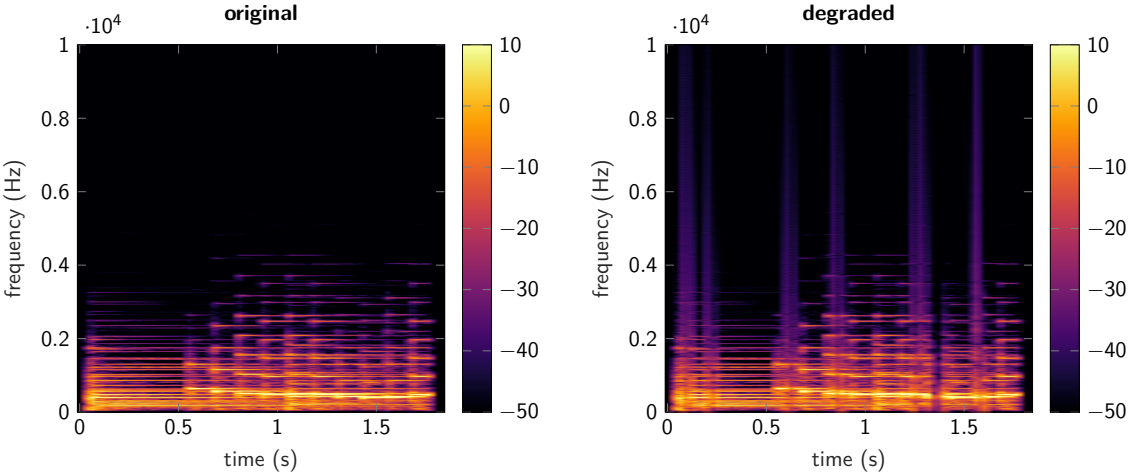


Figure: 18 % of all samples are lost in blocks of length 40 ms

Audio links: original, degraded

Missing signal blocks

Time-frequency domain, hopefully better visibility

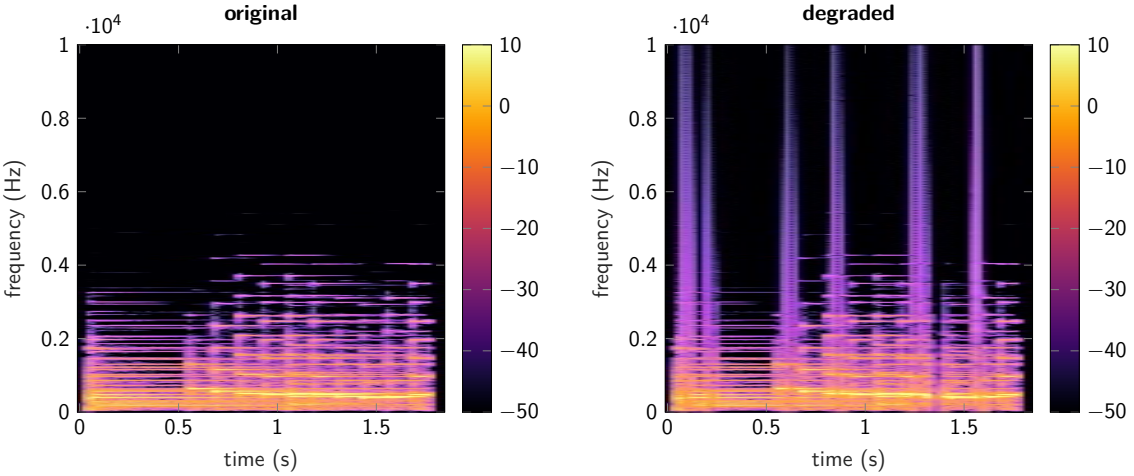


Figure: 18 % of all samples are lost in blocks of length 40 ms

Audio links: original, degraded

Missing random samples

Time-frequency domain

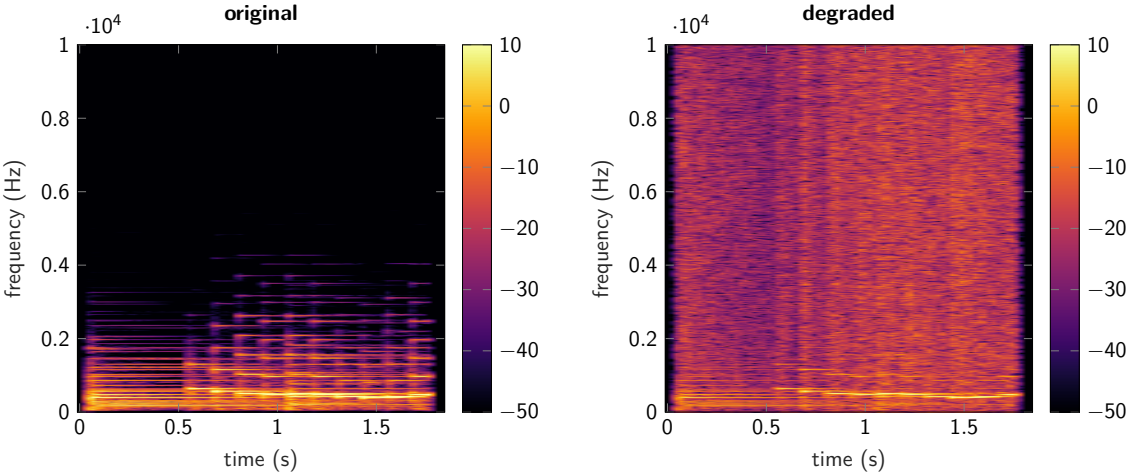


Figure: 60 % of randomly selected samples are lost

Audio links: original, degraded

Clipping

Time-frequency domain

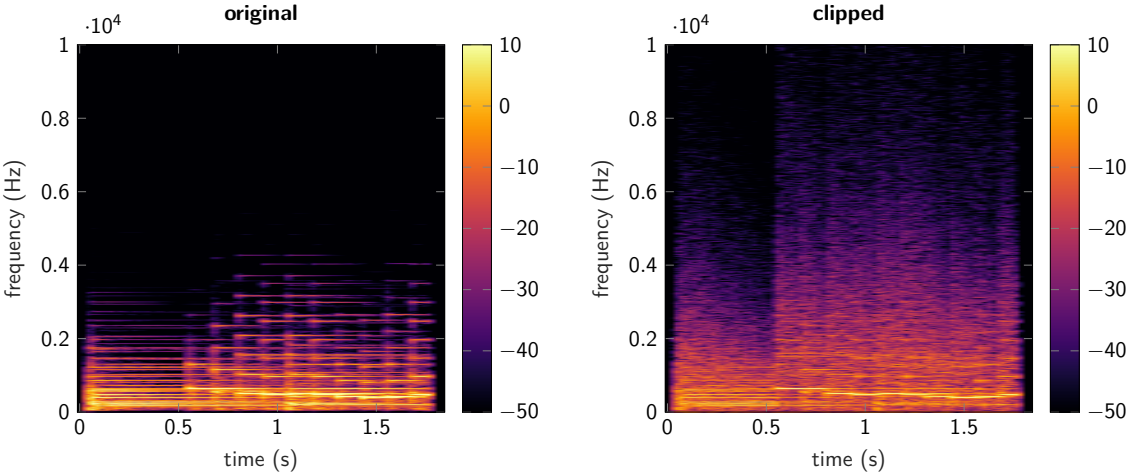


Figure: 55 % of all samples are clipped

Audio links: original, clipped

Audio inpainting

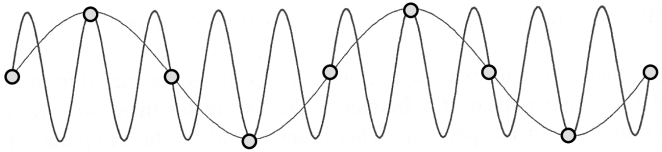
Audio examples

- clean signal
- different solutions to different problems

clipping	**long gaps**	**random loss**
55% clipped	40 ms per gap	60% lost
method 1	method 1	method 1
method 2	method 2	method 2
method 3	method 3	method 3
method 4	method 4	method 4

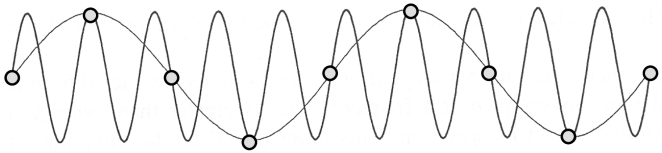
Audio inpainting

- Inpainting is an ill-posed problem, plenty of possible solutions exist
- Non-uniqueness similar to reconstruction (interpolation) from signal samples:



Audio inpainting

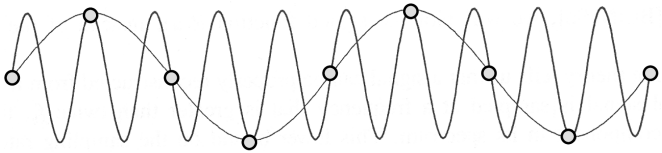
- Inpainting is an ill-posed problem, plenty of possible solutions exist
- Non-uniqueness similar to reconstruction (interpolation) from signal samples:



- It is **necessary to involve some assumption about the signal** (bandlimitedness in the above example)

Audio inpainting

- Inpainting is an ill-posed problem, plenty of possible solutions exist
- Non-uniqueness similar to reconstruction (interpolation) from signal samples:



- It is **necessary to involve some assumption about the signal** (bandlimitedness in the above example)
- **Regularization** can be formulated mathematically, as a function whose value grows when a possible solution deviates from the assumption
- Leading to optimization problems whose solutions are inpainted waveforms (for example, we search for 44 100 unknowns per second of audio)

Inverse problems more formally

- We want to invert the degradation/observation process

- Clean signal $\xrightarrow{\text{deg}}$ Degraded signal
- Clean signal $\xleftarrow{\text{deg}^{-1}}$ Degraded signal

- Since that is not possible, we seek a signal that fits the observation and is regularized by a function R

Inverse problems more formally

- We want to invert the degradation/observation process

- Clean signal $\xrightarrow{\text{deg}}$ Degraded signal
- Clean signal $\xleftarrow{\text{deg}^{-1}}$ Degraded signal

- Since that is not possible, we seek a signal that fits the observation and is regularized by a function R

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \{E(\mathbf{x}, \mathbf{y}) + R(\mathbf{x})\} \qquad \text{(regularized problem)}$$

- $\mathbf{y}$ is the (degraded) observation
- $\mathbf{x}$ is the variable of the problem
- $\hat{\mathbf{x}}$ is the solution
- $E(\mathbf{x}, \mathbf{y})$ ensures that the solution $\hat{\mathbf{x}}$ fits the observed signal $\mathbf{y}$, i.e. the relation $\hat{\mathbf{x}} \xrightarrow{\text{deg}} \mathbf{y}$ holds

Inverse problems more formally

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \{E(\mathbf{x}, \mathbf{y}) + R(\mathbf{x})\} \qquad \text{(regularized problem)}$$

- Set of signals consistent with the observation:

$$\mathbf{x} \in \Gamma = \{\mathbf{u} \mid M_R \mathbf{u} = M_R \mathbf{y}\} \qquad \text{(consistency in time domain)}$$

  - $M_R \mathbf{y}$ is the observed signal
  - M_R selects the reliable samples
- Examples of the the error term:
  - $E(\mathbf{x}, \mathbf{y}) = 0$ if $\mathbf{x} \in \Gamma$, ∞ if $\mathbf{x} \notin \Gamma$ (noiseless case)
  - Distance of $\mathbf{x}$ from the set Γ

Overview of approaches to inpainting

and related audio restoration problems

- Abel (1991) utilizes **limited bandwidth** – possible non-smooth solutions contain high frequencies; therefore he forces bandlimitedness; but oversampling is needed, which is not practical
- Janssen (1986) assumes that the signal is governed by an **AR (autoregressive) model**
- Fong (2001) also **autoregressive** assumption, but the waveform is found by Monte-Carlo particle filtering
- Selesnick (2013) proposes to penalize **high values of signal derivatives** and finds his solution via least squares, explicitly
- Bilen (2015) uses **non-negative matrix factorization** to decompose audio to "notes" and their activation patterns (analog to MIDI)
- Takahashi (2015) uses autoregressive assumption, but through **low-rank properties** of certain matrices generated from the signal
- Rencker (2018) lets the **transform be learned** from training clipped data

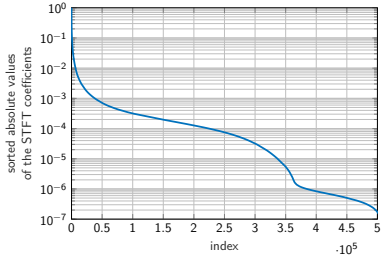
  ⋮
- most of current methods are based on **signal sparsity** . . .

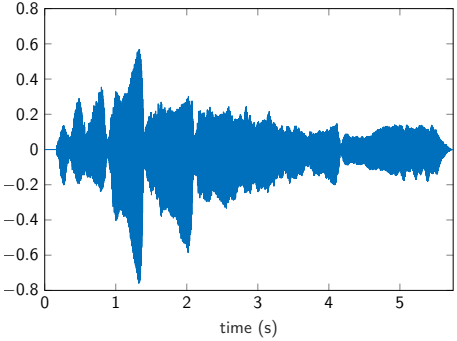
Sparse representations

Sparse representations

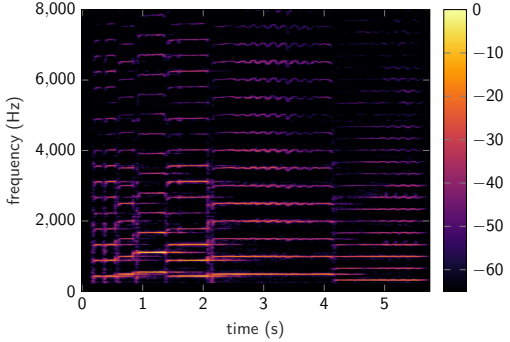
General information

- A particularly successful signal model is “sparse representation”
- Not only audio, even more in image processing
- What is actually meant by *sparse*?





STFT



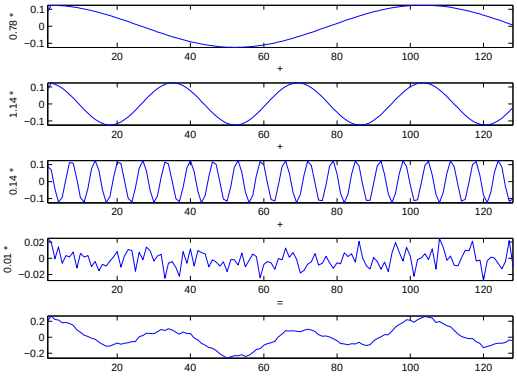
- Transform plays a crucial role!
- JPEG relies on the DCT (Discrete Cosine Transform) + leaving out small coefficients
- Audio field relies on the Short-Time Fourier Transform (STFT), Constant-Q etc.

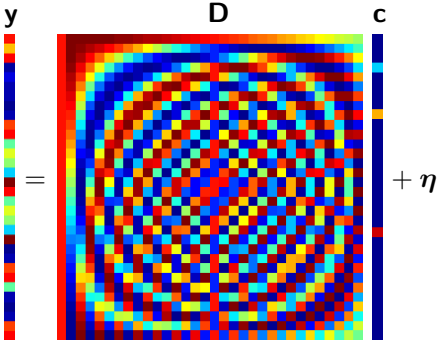
- Time-domain signal $\mathbf{y}$ (i.e. a vector)
  is modelled as a sum of basis signals $\mathbf{d}_n$ with proper weights c_n
  (linear combination, coefficients):
  $$\mathbf{y} = \sum_n c_n \mathbf{d}_n$$
- We **synthesize** the signal
- For example, the inverse DCT is a synthesis, with cosines as basis functions

Sparse representations

DCT example

- $\mathbf{y} = \sum_n c_n \mathbf{d}_n + \boldsymbol{\eta}$ (with noise)
- Vectors $\{\mathbf{d}_n\}$ below are three cosine waves, each of length 128 (taken from the DCT basis)
- In matrix form, $\mathbf{y} = \mathbf{D}\mathbf{c} + \boldsymbol{\eta}$





- **Synthesis** model $\mathbf{y} = \mathbf{D}\mathbf{c}$
- For transforms like DCT and DFT, there is a *unique* representation $\mathbf{c} = \mathbf{D}^{-1}\mathbf{y}$
  $\mathbf{D}^{-1}$ is called **analysis**, also denoted $\mathbf{A}$
- Redundant transforms do not offer unique representation, but might promote sparsity

- **Synthesis** model $\mathbf{y} = \mathbf{D}\mathbf{c}$
- For transforms like DCT and DFT, there is a *unique* representation $\mathbf{c} = \mathbf{D}^{-1}\mathbf{y}$
  $\mathbf{D}^{-1}$ is called **analysis**, also denoted $\mathbf{A}$
- Redundant transforms do not offer unique representation, but might promote sparsity

- Sparsity (in suitable domain) can be the regularizer for an ill-posed inverse problem (i.e. non-sparse solutions will not be preferred during the search)

- **Synthesis** model $\mathbf{y} = \mathbf{D}\mathbf{c}$
- For transforms like DCT and DFT, there is a *unique* representation $\mathbf{c} = \mathbf{D}^{-1}\mathbf{y}$
  $\mathbf{D}^{-1}$ is called **analysis**, also denoted $\mathbf{A}$
- Redundant transforms do not offer unique representation, but might promote sparsity

- Sparsity (in suitable domain) can be the regularizer for an ill-posed inverse problem (i.e. non-sparse solutions will not be preferred during the search)

- Notation: $\|\mathbf{c}\|_0 =$ number of non-zero elements of $\mathbf{c}$

- Truly sparse solutions not achievable (and not necessary) in practical problems
- Moreover, optimizing true sparsity has combinatorial complexity, it is *NP-hard*
- Two ways to approximate available:
  - ℓ_1-norm-based: $\|\mathbf{c}\|_1 = \sum_n |c_n|$
  - greedy

- Truly sparse solutions not achievable (and not necessary) in practical problems
- Moreover, optimizing true sparsity has combinatorial complexity, it is *NP-hard*
- Two ways to approximate available:
  - ℓ_1-norm-based: $\|\mathbf{c}\|_1 = \sum_n |c_n|$
  - greedy

Algorithms

Example: convex method

ℓ_1-norm-based

$$\arg\min_{\mathbf{c}} \|\mathbf{c}\|_1 \quad \text{subject to} \quad M_\mathrm{R}\mathbf{D}\mathbf{c} = M_\mathrm{R}\mathbf{y}$$

$\mathbf{c}$	time-frequency coefficients of the restored signal
$\mathbf{y}$	observed time-domain signal
M_R	selection of the observed (reliable) samples
$\mathbf{D}$	synthesis operator: coefficients $\mapsto$ signal

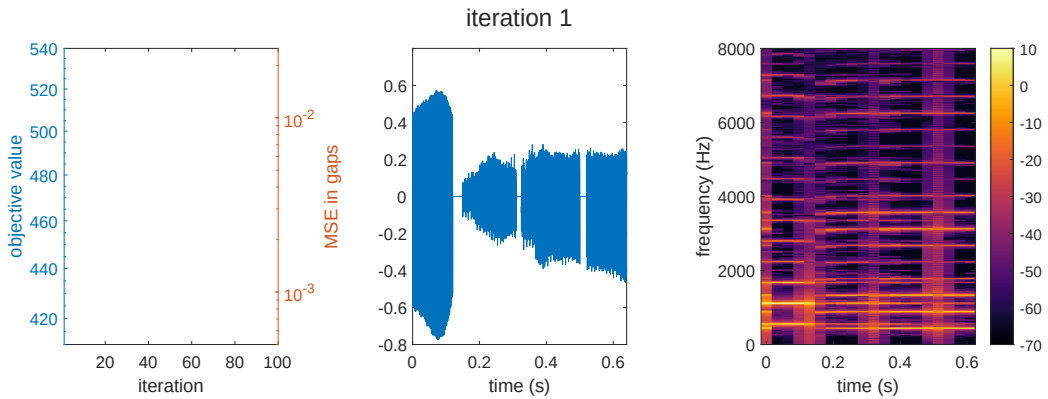
- objective is convex
- constraints are linear

- practically solvable using splitting algorithms

Example: convex method

ℓ_1-norm-based

$$\arg\min_{\mathbf{c}} \|\mathbf{c}\|_1 \quad \text{subject to} \quad M_R \mathbf{D}\mathbf{c} = M_R \mathbf{y}$$



left – sparsity $\|\mathbf{c}\|_1$, MSE in time domain, **center** – synthesized solution $\mathbf{Dc}$, **right** – iterates (coefficients) $\mathbf{c}$

Example: non-convex method

Sparse audio inpainter (SPAIN)

$$\arg\min_{\mathbf{x},\mathbf{c}} \|\mathbf{c}\|_0 \quad \text{subject to} \quad M_R\mathbf{x} = M_R\mathbf{y},\ \|\mathbf{A}\mathbf{x} - \mathbf{c}\| < \varepsilon$$

$\mathbf{c}$	time-frequency coefficients of the restored signal
$\mathbf{x}$	the restored signal
$\mathbf{y}$	observed time-domain signal
M_R	selection of the observed (reliable) samples
$\mathbf{A}$	analysis operator: signal $\mapsto$ coefficients
ε	chosen tolerance

- constraints are linear and convex
- objective is non-convex

- theoretically not solvable in polynomial time
- need for a heuristic algorithm

Example: non-convex method

Sparse audio inpainter (SPAIN)

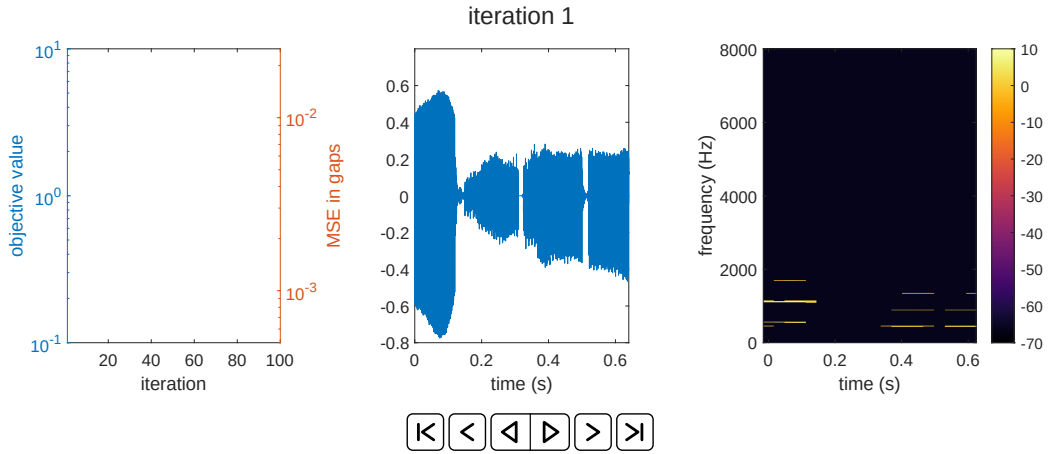
$$\arg\min_{\mathbf{x},\mathbf{c}} \|\mathbf{c}\|_0 \quad \text{subject to} \quad M_R\mathbf{x} = M_R\mathbf{y},\ \|\mathbf{A}\mathbf{x} - \mathbf{c}\| < \varepsilon$$

- it is almost solvable for known sparsity k of the solution
- first idea:
  - start with low k
  - search for the best approximation of $\mathbf{A}\mathbf{x}$ with k-sparse $\mathbf{c}$
  - increase k and repeat
- second idea:
  - merge it with a convex method (add some intermediate steps) such that it works in practice

Example: non-convex method

Sparse audio inpainter (SPAIN)

$$\arg\min_{\mathbf{x},\mathbf{c}} \|\mathbf{c}\|_0 \quad \text{subject to} \quad M_R\mathbf{x} = M_R\mathbf{y},\ \|\mathbf{A}\mathbf{x} - \mathbf{c}\| < \varepsilon$$

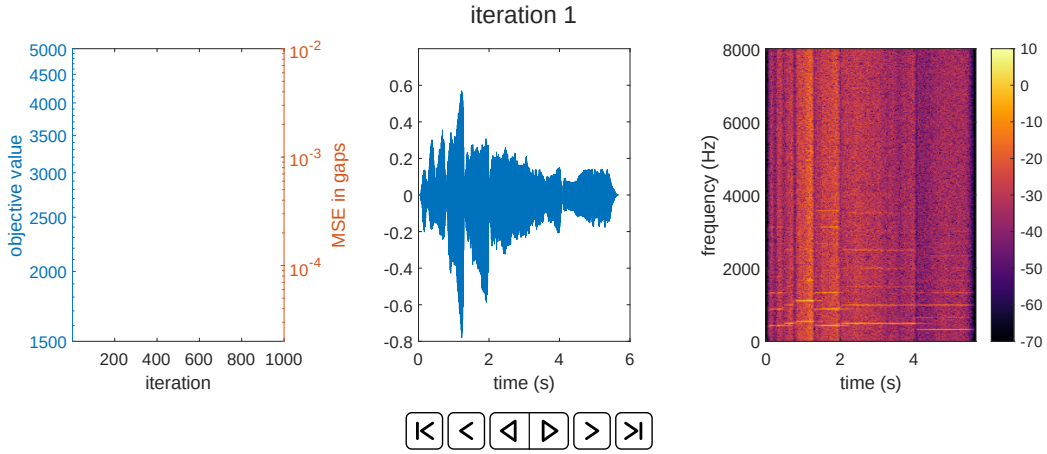


left – error $\|\mathbf{A}\mathbf{x} - \mathbf{c}\|$, MSE in time domain, **center** – the solution $\mathbf{x}$, **right** – the sparse coefficients $\mathbf{c}$

2nd example: convex method

ℓ_1-norm-based

$$\arg\min_{\mathbf{c}} \|\mathbf{c}\|_1 \quad \text{subject to} \quad M_R \mathbf{D}\mathbf{c} = M_R \mathbf{y}$$

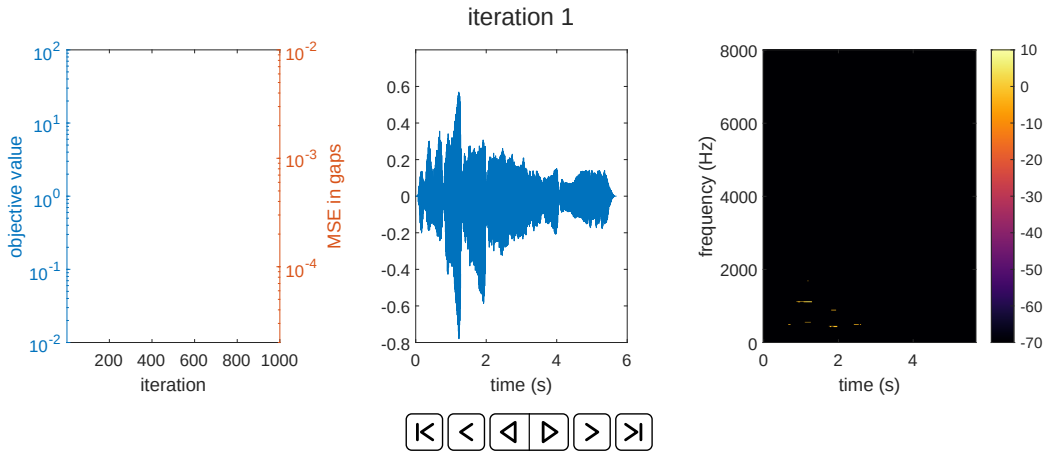


left – sparsity $\|\mathbf{c}\|_1$, MSE in time domain, **center** – synthesized solution $\mathbf{D}\mathbf{c}$, **right** – iterates (coefficients) $\mathbf{c}$, **audio link**

2nd example: non-convex method

Sparse audio inpainter (SPAIN)

$$\arg\min_{\mathbf{x},\mathbf{c}} \|\mathbf{c}\|_0 \quad \text{subject to} \quad M_R\mathbf{x} = M_R\mathbf{y},\ \|\mathbf{A}\mathbf{x} - \mathbf{c}\| < \varepsilon$$

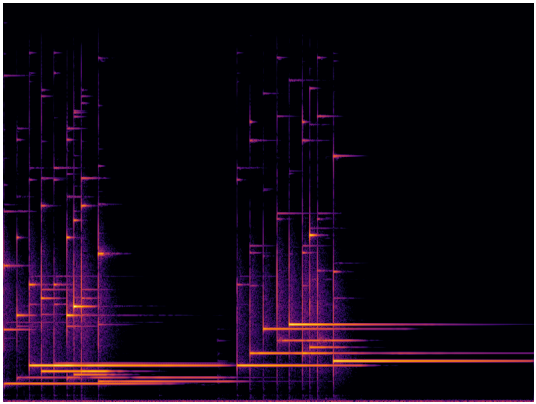


left – error $\|\mathbf{A}\mathbf{x} - \mathbf{c}\|$, MSE in time domain, **center** – the solution $\mathbf{x}$, **right** – the sparse coefficients $\mathbf{c}$, **audio link**

Other approaches

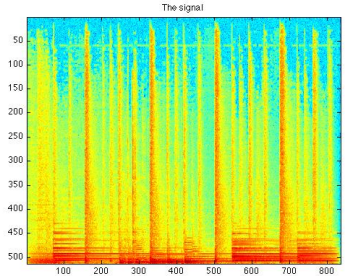
Social sparsity

- Music contains harmonic and transient components
- Horizontal and vertical structures in spectrogram
- Creating groups of coefficients (*group* or *social* sparsity)
- Using this as regularizers instead of plain sparsity

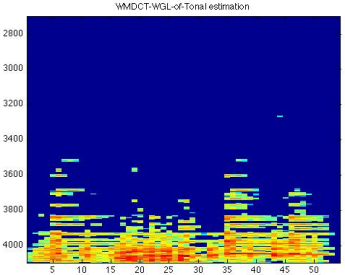


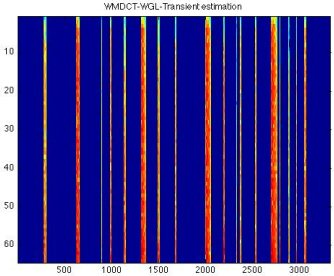
Social sparsity

Audio decomposition into harmonic and transient components



Audio links:
original
tonal
transient





Autoregressive modeling

Autoregressive (AR) model

Signal samples are linear combinations of previous samples + white noise.

Autoregressive modeling

Autoregressive (AR) model

Signal samples are linear combinations of previous samples + white noise.

. . . from other perspective

AR process is an output of an all-pole IIR filter whose input is white noise.

Autoregressive modeling

Autoregressive (AR) model

Signal samples are linear combinations of previous samples + white noise.

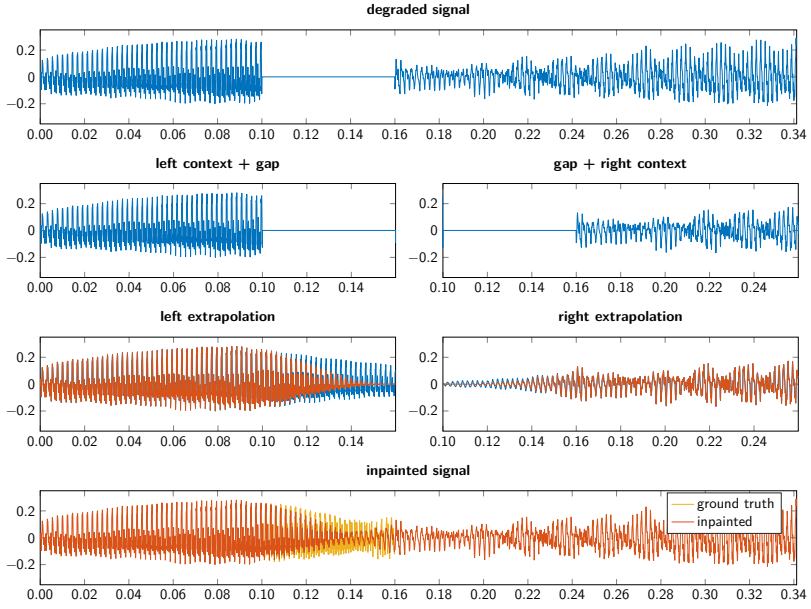
. . . from other perspective

AR process is an output of an all-pole IIR filter whose input is white noise.

- Mostly in line with the source-filter model of speech
- Applicable also on music in practice

- Useful model for analysis (and extrapolation!) of time series, including audio signals

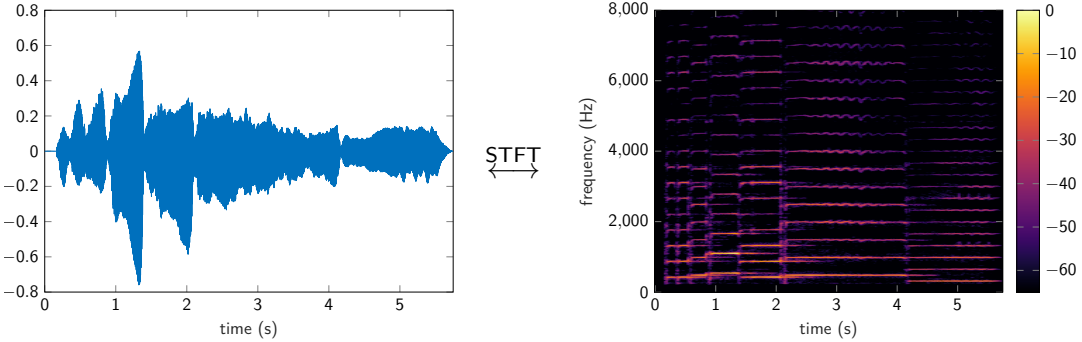
- Simple approach for long gaps – extrapolation of the reliable sections
- More demanding approach – Janssen's iteration



time axis in seconds, note the time-stretching of the right context

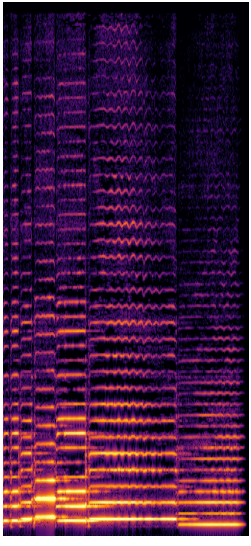
Matrix factorization

- Musical and speech signals consist of repeating spectral patterns (notes, syllables)
- The these patterns are (simultaneously) activated
- The mix differs in time, but the set of the patterns stays constant

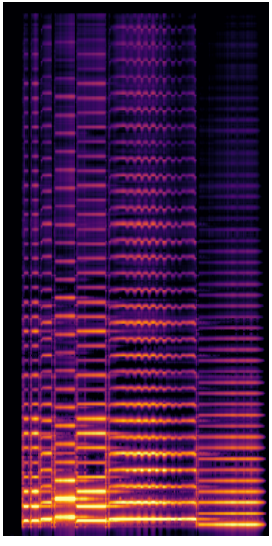


- This can be modeled as a non-negative matrix factorization (NMF) of the power spectrogram $\mathbf{P}$

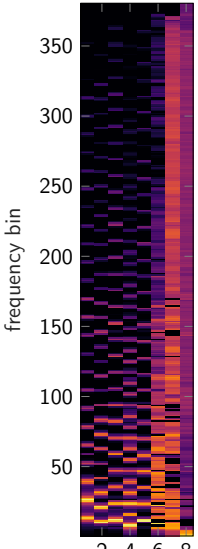
P



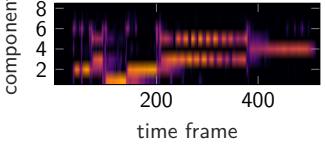
V = WH ≈ P



W



H



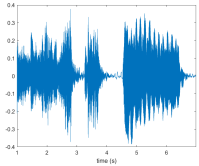
Audio links:

whole signal
component 1
component 2
component 3
component 4
component 5
component 6
component 7
component 8

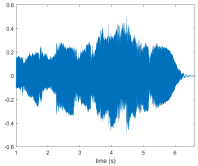
Evaluation of the reconstruction quality

Evaluation of the reconstruction quality – Audio Database

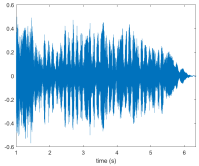
- 10 musical excerpts with approximate length 7 seconds
- Different degrees of signal sparsity w.r.t. STFT
- 44.1 kHz sampling rate, 16 bps



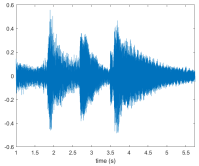
(a) violin



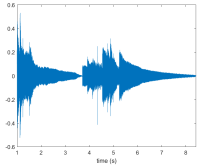
(b) clarinet



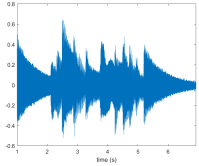
(c) bassoon



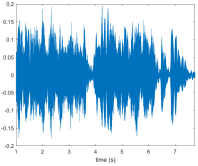
(d) harp



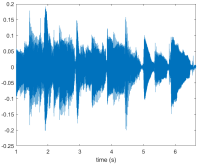
(e) glockenspiel



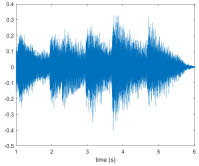
(f) celesta



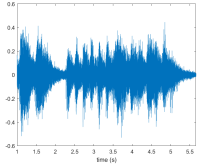
(g) accordion



(h) guitar



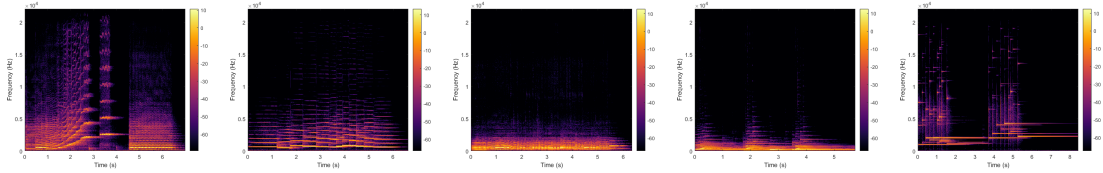
(i) piano



(j) wind ensemble

Evaluation of the reconstruction quality – Audio Database

- 10 musical excerpts with approximate length 7 seconds
- Different degrees of signal sparsity w.r.t. STFT
- 44.1 kHz sampling rate, 16 bps



(a) violin (b) clarinet (c) bassoon (d) harp (e) glockenspiel

(f) celesta (g) accordion (h) guitar (i) piano (j) wind ensemble

Evaluation of the reconstruction quality – evaluation methods

- SDR (signal-to-distortion ratio)
  - easiest and simplest method
  - compares only the physical similarity of waveforms
  - does not correspond to human hearing
- PEAQ
  - ITU-R standard for evaluating audio quality
  - used very often
  - free unofficial Matlab implementation available
- PEMO-Q
  - may produce significantly different results compared to PEAQ (especially for audio declipping)
  - free Matlab implementation for research
- ViSQOLAudio
  - relatively new method
  - C++ implementation on GitHub (open source)
  - good results in general
  - not tested for audio inpainting

ODG	Impairment description
0.0	Imperceptible
−1.0	Perceptible, but not annoying
−2.0	Slightly annoying
−3.0	Annoying
−4.0	Very annoying

On the computational complexity

On the computational complexity

- Work focused on restoration quality, **far from real-time** processing
- Implementations in **Matlab** (LTFAT has backend in C), recently **PyTorch**
- Sparsity-based methods
  - dominated by signal synthesis and analysis
- AR-based methods
  - inversion of very large matrices
  - extrapolation-based method is much faster but with limited usability (only long gaps)
- NMF-based methods
  - even worse matrix manipulations
- Computational complexity depends on the number of missing samples for some algorithms
- Usually there is a trade-off between time and restoration quality

Future research and possible cooperation

Future research and possible cooperation

- Plenty of ideas how to improve state-of-the-art methods or develop new
  - model-based deep learning
  - making older ideas computationally feasible
- Bachelor & diploma & dissertation theses
- Collaboration
  - Jiří Schimmel
  - IRIT (Toulouse), ARI (Vienna)
- Research funding
  - MŠMT projects (2013–2016)
  - FWF–GAČR project (2017–2019)
  - GAČR projects (2020–2022, 2023–2025)

This is the end. . .

This is the end...

Thank you for your attention!