On the use of quasi-random sets for the decimation of continuous wavelet transforms

Nicki Holighaus

joint work with Günther Koliander, Clara Hollomey, and Friedrich Pillichshammer

Acoustics Research Institute, Vienna, Austria

ASCETE Workshop 2023, Grenoble

09.11.2023

Overview

Preliminaries

Discretization of continuous time-frequency representations

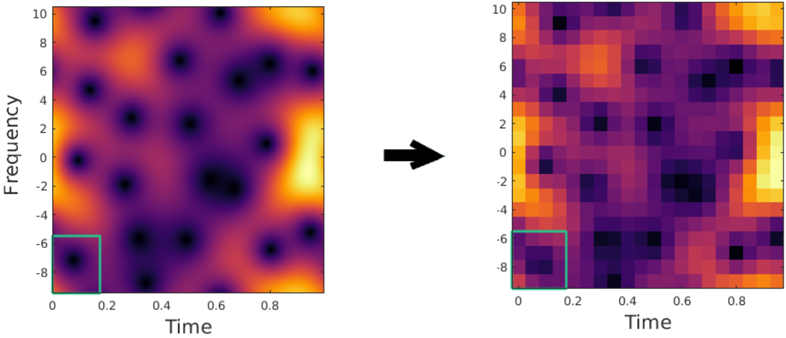


Figure: Continuous STFT (left) and discretized STFT (right).

Preliminaries - The general idea

In this talk:

- Perfect reconstruction of input from discrete coefficients (Invertibility)
- Approximate energy-preservation between input and coefficients (Stability)

The discrete representation forms a frame.

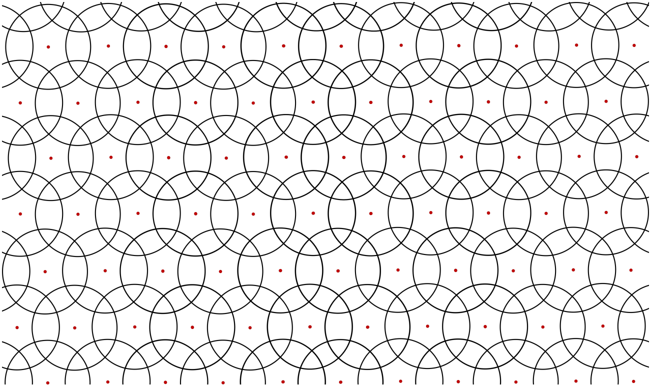
Preliminaries - Recap: STFT discretization

The STFT is generated by time-frequency shifts of a fixed window function g:

$$V_g f(x,\xi) = \langle f, g_{x,\xi} \rangle = \mathcal{F}(f \cdot \overline{g_{x,0}})(\xi), \quad \text{where } g_{x,\xi}(t) = g(t-x)e^{2\pi i \xi t} \tag{1}$$

- Time-frequency resolution is uniform (no dependence on (x,ξ)).
- Discretization on a regular grid ($x = \ell a, \xi = jb$, $j, \ell \in \mathbb{Z}$, $a, b > 0$) is common and works well.
- Very efficient via windowed FFT.

Other discretizations are possible, sometimes similarly efficient, but uncommon. For example, random or on a general lattice:



Consider a family $\Psi = (\psi_\xi)_\xi$ and define

$$V_\Psi f(x,\xi) = \langle f, \psi_\xi(\bullet - x)\rangle = (f * \overline{\psi_\xi(-\bullet)})(x). \tag{2}$$

- (Continuous) linear, time-invariant filter bank.
- Time-frequency representation if each ψ_ξ is time-frequency localized around $(0,\xi)$.
- Time-frequency resolution may change with frequency ξ.

Preliminaries - Wavelets as case study

Consider $\Psi = (\psi_\xi)_\xi$, with $\psi_\xi = \mathbf{D}_{\xi^{-1}}\psi = \sqrt{\xi}\psi(\xi \cdot \bullet)$. Then

$$V_\Psi f(x, \xi) = \langle f, \psi_\xi(\bullet - x)\rangle =: W_\psi f(x, \xi) \tag{3}$$

is the continuous wavelet transform of f with respect to the mother wavelet ψ.

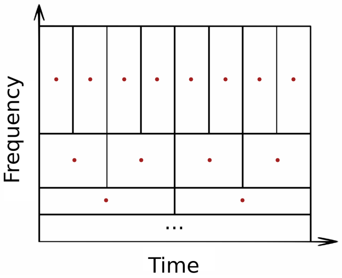


Figure: Illustration of wavelet time-frequency resolution.

Preliminaries - Wavelets as case study

Consider $\Psi = (\psi_\xi)_\xi$, with $\psi_\xi = \mathbf{D}_{\xi^{-1}}\psi = \sqrt{\xi}\psi(\xi \cdot \bullet)$. Then

$$V_\Psi f(x,\xi) = \langle f, \psi_\xi(\bullet - x)\rangle =: W_\psi f(x,\xi) \quad (3)$$

is the continuous wavelet transform of f with respect to the mother wavelet ψ.

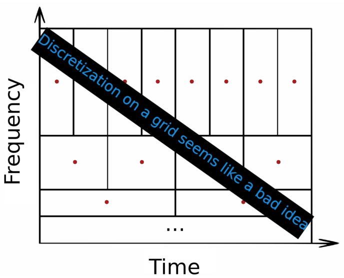


Figure: Illustration of wavelet time-frequency resolution.

Preliminaries - Frames

A countable (or even finite) subset of $(\psi_j)_j \subset \Psi$ is a frame, if there exist constants $0 < A \leq B < \infty$, such that

$$A\|f\|^2 \leq \sum_j |\langle f, \psi_j \rangle|^2 \leq B\|f\|^2, \quad \forall\, f. \tag{4}$$

- Implies that the discretization can be inverted (perfect reconstruction is possible)
- B/A is stability estimate (think condition number)
- Generalization of stable spanning sets (e.g., orthonormal bases)
- There is a dual frame $(\widetilde{\psi_j})_j$, such that $f = \sum_j \langle f, \psi_j \rangle \widetilde{\psi_j}$, for all f.

Discretization of the wavelet transform

Classical wavelet discretization

By convention, $W_\psi f$ is usually discretized to $\log$-uniformly-spaced frequencies $\xi = s^m$, $d > 0,\ j \in \mathbb{Z}$.

- Uniform decimation of $x = \ell d$, $d > 0,\ \ell \in \mathbb{Z}$, e.g., à trous algorithm,
- Dyadic wavelet bases: $s = 2$ and $x = 2^{-j}\ell$, $\ell \in \mathbb{Z}$.

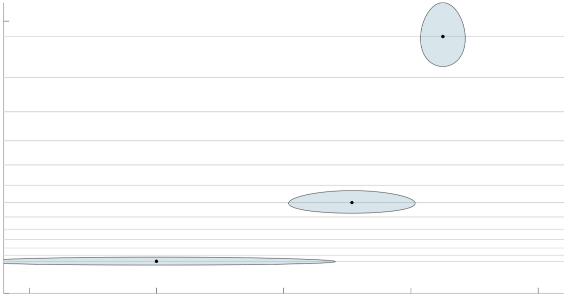


Figure: Illustration of “classical” wavelet discretization.

Classical wavelet discretization - II

More general (constant-Q-like) low-to-moderate redundancy wavelet discretization: $W_\psi f(s^{-j}d\ell, s^j)$, $j, \ell \in \mathbb{Z}$, for some $s.d > 0$.

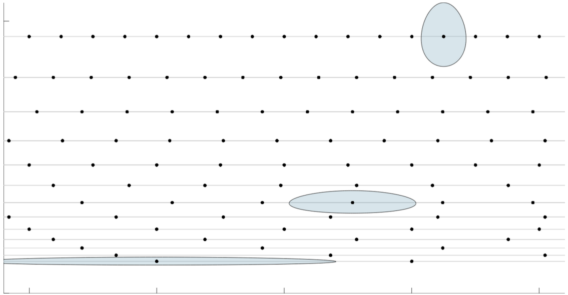


Figure: Illustration of “classical” wavelet discretization.

Issues of non-uniform filter banks

- Coefficients are not aligned
- No meaningful notion of time frames
- No use of efficient matrix calculus or matrix methods
- Dual frame $(\widetilde{\psi_{j,\ell}})_{j,\ell}$ is unstructured in general
- Reconstruction is inefficient or even computationally infeasible

Can we find a wavelet discretization that combines uniform decimation, low redundancy and efficient, perfect reconstruction

Enter Quasi-Random (Low Discrepancy) Sequences

Usually, $(p_n)_{n\in\mathbb{N}_0}$ in a domain Ω (think the unit cube $[0,1)^D$) is a quasi-random sequence or equidistributed, if

$$\lim_{N\to\infty} \frac{\#((p_0,\dots,p_{N-1})\cap B)}{N} = \frac{|B|}{|\Omega|}, \tag{5}$$

for all nice sets $B\subset\Omega$.

The proportion of points falling into B is proportional to the size of B.

Given a set of points $P_N = (p_0, \ldots, p_{N-1})$, its discrepancy is

$$D(P_N) = \sup_{B \in \mathcal{B}} \left| \frac{\#(P_N \cap B)}{N} - \frac{|B|}{|\Omega|} \right|, \tag{6}$$

where $\mathcal{B}$ is the collection of all nice sets.

Discrepancy measures the equidistribution of finite sets.

Informally, a sequence $(p_0, p_1, \ldots) \subset \mathcal{B}$, has low discrepancy, if $D(P_N)$, with $P_N = (p_0, \ldots, p_{N-1})$, is small for all $N \in \mathbb{N}$.

Note: The appropriate notion of “small” depends on $\mathcal{B}$, Ω, and the considered problem.

Main use: Better substitute for random samples in Monte Carlo schemes.

Discrepancy - Canonical Example

Set

$\Omega = [0,1)^D$ and $\mathcal{B} = \{[a_0, b_0) \times \cdot \times [a_{D-1}, b_{D-1}) \ : \ 0 \leq a_j < b_j < 1,$ for all $j = 0, \ldots$

i.e., boxes in the unit cube.

There exist sequences $(p_0, p_1, \ldots) \subset [0,1)^D$ that achieve
$D(P_N) \leq C_D \log(N)^d / N$.

This is optimal for $D = 1, 2$ and conjectured to be optimal for any $D \in \mathbb{N}$.

Construction of Low Discrepancy Sequences

Construction of good sequences can be quite involved, especially if D is large. See, e.g.,

- J. Dick and F. Pillichshammer. *Digital Nets and Sequences: Discrepancy Theory and Quasi–Monte Carlo Integration*
- L. Kuipers and H. Niederreiter. *Uniform Distribution of Sequences*

Luckily, there are very simple constructions for $D = 1$.

Fix some badly approximable number $\alpha \in \mathbb{R} \setminus \mathbb{Q}$, e.g., $\alpha = \frac{1+\sqrt{5}}{2}, \sqrt{2}, \ldots$. Then

$$p_n = \alpha n - \lfloor \alpha n \rfloor, \quad \text{for all } n \in \mathbb{N}_0, \tag{7}$$

defines a low discrepancy sequence.

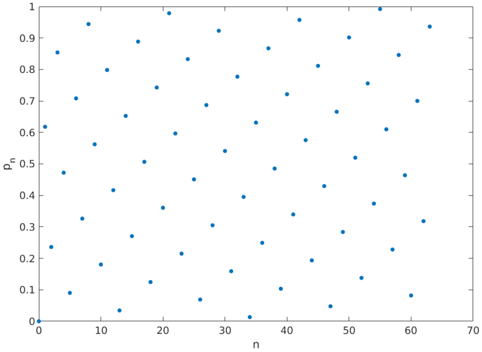


Figure: First 64 elements of the Kronecker sequence with $\alpha = \frac{1+\sqrt{5}}{2}$.

Grid-Based Wavelet Decimation - With Delays

Recall that $W_\psi f(x,\xi)$ represents the time-freuency energy of f around time x and frequency ξ:

- The mother wavelet ψ is localized around time 0 and frequency 1.
- $\psi_\xi(\bullet - x)$ is localized around time x and frequency ξ.

For some step size $d > 0$, a base frequency ξ_b, a frequecy step $q \in 1/\mathbb{N}$, and $(p_n)_n$ a Kronecker sequence, we consider $W_\psi f$ at the points $(d(\ell + p_{j-1}), \xi_\mathrm{b}(1 + jq))_{j\in\mathbb{Z},\ell\in\mathbb{N}}$. Equivalently, we delay $\psi_{\xi_\mathrm{b}(1+jq)}$ by $0 \leq dp_{j-1} < d$ and filter with decimation step d.

Grid + Delays - II

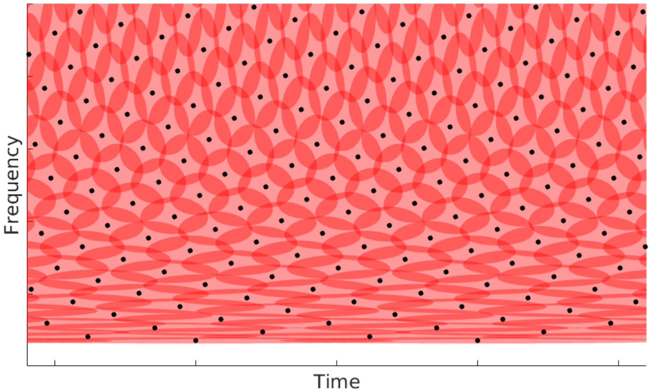


Figure: Time-frequency coverings generated by a uniform grid with Kronecker delays.

Grid + Delays - III

Theorem (Invertible Wavelet Decimation)

If the mother wavelet ψ is (essentially) compactly supported and sufficiently smooth, if ξ_{b} is not too large, and if $(p_n)_{n\in\mathbb{Z}}$ is as before, then there exist positive constants $c_j > 0$, $j \in \mathbb{Z}$, such that the wavelet system $(c_j \cdot \psi_{\xi_j}(\bullet - x_{\ell,j}))_{\ell,j\in\mathbb{Z}}$ with

$$\xi_j = \xi_{\mathrm{b}}(1 + jq), \quad j \geq 0, \quad \textit{and} \quad \xi_j = (1+q)^j \cdot \xi_{\mathrm{b}}, \quad j < 0, \tag{8}$$

and

$$x_{\ell,j} = d(l + p_j), \quad l, j \in \mathbb{Z}, \tag{9}$$

is a frame, provided that $d > 0$ and $q \in 1/\mathbb{N}$ are small enough. In particular, for $j \geq 0$, all c_j can be chosen equal to 1.

Note: In practice, the frequencies ξ_j, $j < 0$, are covered by a (set of) low-pass filter(s).

Grid + Delays - Spectrograms

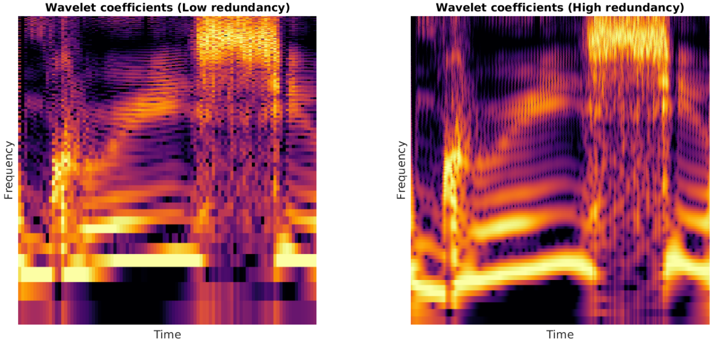


Figure: Spectrogram of grid-based wavelets, with log-scaled frequency axis.

Grid-Based Wavelet Decimation - With Rotation

Grid + Rotation

Let $\alpha \in [0, 1)$ be a badly approximable number and define

$$A\mathbb{Z}^2, \quad \text{with} \quad A = \begin{pmatrix} 1 & -\alpha \\ \alpha & 1 \end{pmatrix}$$

Consider $W_\psi f(x, \xi)$ at all points $(x, \xi) \in A\mathbb{Z}^2 \cap (\mathbb{R} \times \mathbb{R}^+)$.

Grid + Rotation - II

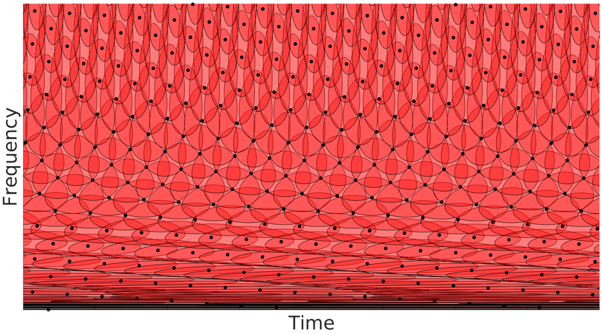


Figure: Time-frequency covering generated by a rotated lattice.

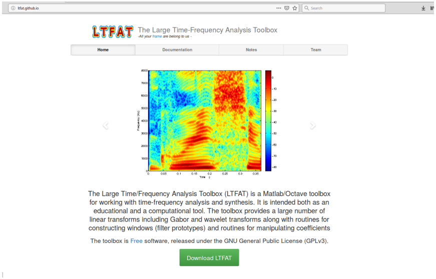
Theorem (Invertible Wavelet Decimation)

If the mother wavelet ψ is (essentially) compactly supported and sufficiently smooth, and let $(\lambda_n)_{n\in\mathbb{N}_0}$, with $\lambda_n = (x_n, \xi_n)$, be any ordering of $A\mathbb{Z}^2 \cap (\mathbb{R}\times\mathbb{R}^+)$, i.e., $(\lambda_n)_{n\in\mathbb{N}_0} = A\mathbb{Z}^2 \cap (\mathbb{R}\times\mathbb{R}^+)$ as sets. Then, for any $\beta > 0$ small enough, the wavelet system

$$(\psi_{\xi_n}(\bullet - x_n))_{n\in\mathbb{N}_0} \tag{10}$$

is a frame.

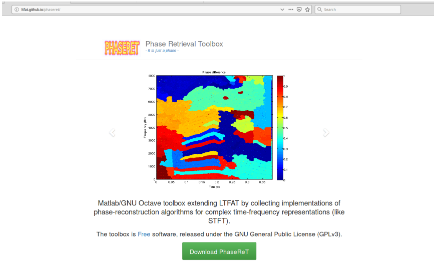
Implementation and more



MATLAB/octave Toolbox for time-frequency analysis (with C backend)

- LTFAT - The Large Time-Frequency Analysis Toolbox - ltfat.github.io
- PHASERET - Phase retrieval toolbox - ltfat.github.io/phaseret/
- libLTFAT - C backend library - ltfat.github.io/libltfat/
- Webpage of this paper - ltfat.github.io/notes/057/

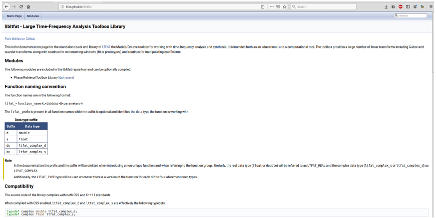
Implementation and more



MATLAB/octave Toolbox for phaseless reconstruction with short-time Fourier transforms (with on- and offline methods and C backend)

- LTFAT - The Large Time-Frequency Analysis Toolbox - ltfat.github.io
- PHASERET - Phase retrieval toolbox - ltfat.github.io/phaseret/
- libLTFAT - C backend library - ltfat.github.io/libltfat/
- Webpage of this paper - ltfat.github.io/notes/057/

Implementation and more



ltfat.github.io/libltfat/

Main Page | Modules

libltfat - Large Time-Frequency Analysis Toolbox Library

Fork libltfat on Github

This is the documentation page for the standalone back end library of LTFAT the Matlab/Octave toolbox for working with time-frequency analysis and synthesis. It is intended both as an educational and a computational tool. The toolbox provides a large number of linear transforms including Gabor and wavelet transforms along with routines for constructing windows (filter prototypes) and routines for manipulating coefficients

Modules

The following modules are included in the libltfat repository and can be optionally compiled:

- Phase Retrieval Toolbox Library libphaseret

Function naming convention

The function names are in the following format:

ltfat_<function_name>[_<d|s|dc|sc>](<parameters>)

The ltfat_ prefix is present in all function names while the suffix is optional and identifies the data type the function is working with:

Data type suffix

Suffix	Data type
d	double
s	float
dc	ltfat_complex_d
sc	ltfat_complex_s

Note

In the documentation the prefix and the suffix will be omitted when introducing a non-unique function and when referring to the function group. Similarly, the real data type (float or double) will be referred to as LTFAT_REAL and the complex data type (ltfat_complex_s or ltfat_complex_d) as LTFAT_COMPLEX.

Additionally, the LTFAT_TYPE type will be used whenever there is a version of the function for each of the four aforementioned types.

Compatibility

The source code of the library complies with both C99 and C++11 standards.

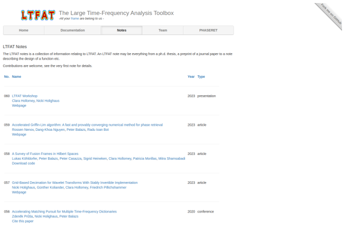
When compiled with C99 enabled, ltfat_complex_d and ltfat_complex_s are effectively the following typedefs:

```
typedef complex double ltfat_complex_d;
typedef complex float ltfat_complex_s;
```

Standalone version of the LTFAT C backend library

- LTFAT - The Large Time-Frequency Analysis Toolbox - ltfat.github.io
- PHASERET - Phase retrieval toolbox - ltfat.github.io/phaseret/
- libLTFAT - C backend library - ltfat.github.io/libltfat/
- Webpage of this paper - ltfat.github.io/notes/057/

Implementation and more



LTFAT The Large Time-Frequency Analysis Toolbox

-All your frame are belong to us -

Home | Documentation | Notes | Team | PHASERET

LTFAT Notes

The LTFAT notes is a collection of information relating to LTFAT. An LTFAT note may be everything from a ph.d. thesis, a preprint of a journal paper to a note describing the design of a function etc.

Contributions are welcome, see the very first note for details.

No.	Name	Year	Type
060	LTFAT Workshop Clara Hollomey, Nicki Holighaus Webpage	2023	presentation
059	Accelerated Griffin-Lim algorithm: A fast and provably converging numerical method for phase retrieval Rossen Nenov, Dang-Khoa Nguyen, Peter Balazs, Radu Ioan Bot Webpage	2023	article
058	A Survey of Fusion Frames in Hilbert Spaces Lukas Köhldorfer, Peter Balazs, Peter Casazza, Sigrid Heineken, Clara Hollomey, Patricia Morillas, Mitra Shamsabadi Download code	2023	article
057	Grid-Based Decimation for Wavelet Transforms With Stably Invertible Implementation Nicki Holighaus, Günther Koliander, Clara Hollomey, Friedrich Pillichshammer Webpage	2023	article
056	Accelerating Matching Pursuit for Multiple Time-Frequency Dictionaries Zdeněk Průša, Nicki Holighaus, Peter Balazs Cite this paper	2020	conference

Manuscript, code and audio examples

- LTFAT - The Large Time-Frequency Analysis Toolbox - ltfat.github.io
- PHASERET - Phase retrieval toolbox - ltfat.github.io/phaseret/
- libLTFAT - C backend library - ltfat.github.io/libltfat/
- Webpage of this paper - ltfat.github.io/notes/057/

A Few References

H. Kempka, M. Schäfer, and T. Ullrich.
General coorbit space theory for quasi-Banach spaces and inhomogeneous function spaces with variable smoothness and integrability.
J. Fourier Anal. Appl., 23(6):1348–1407, 2017.

N. Holighaus, C. Wiesmeyr, and P. Balazs.
Continuous warped time-frequency representations—coorbit spaces and discretization.
Applied and Computational Harmonic Analysis, 47(3):975–1013, 2019.

N. Holighaus, G. Koliander, C. Hollomey, and F. Pillichshammer.
Grid-Based Decimation for Wavelet Transforms With Stably Invertible Implementation.
IEEE/ACM Transactions on Audio, Speech, and Language Processing, 31:789–801, 2023.

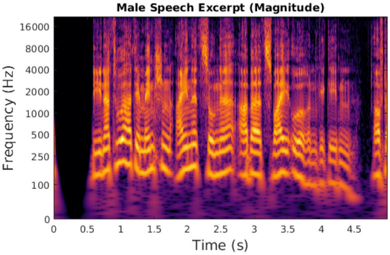
N. Holighaus, and G. Koliander.
Rotated time-frequency lattices are sets of stable sampling for continuous wavelet systems.
2023 International Conference on Sampling Theory and Applications (SampTA), New Haven, CT, USA, 2023.

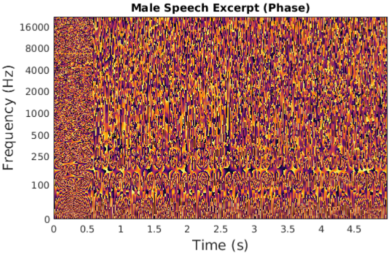
Thank you for your attention!

If you want more, I can talk about non-iterative phaseless reconstruction for wavelet transforms.

Non-iterative phaseless reconstruction for wavelet transforms

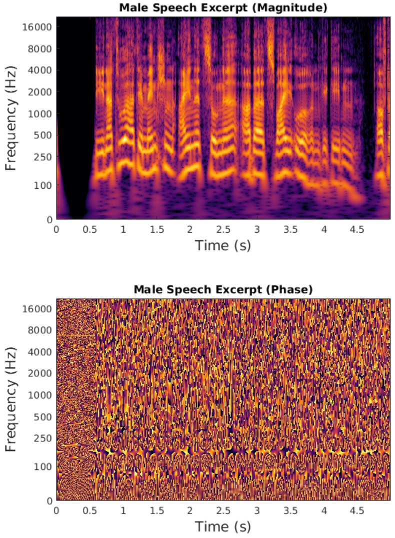
Phaseless reconstruction/Phase retrieval





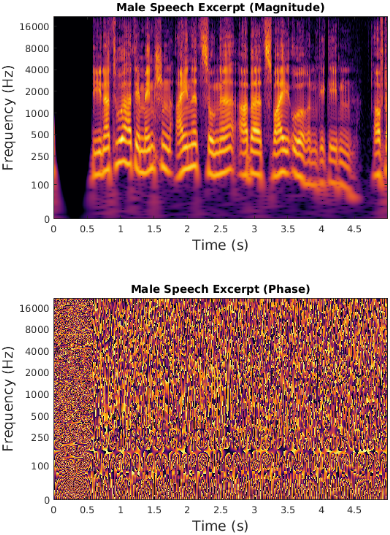
- complex-valued representations as magnitude and phase
- modification/generation of magnitudes comparably easy
- phase invalid after magnitude-based processing
- signal reconstruction from magnitude is a challenging problem
- theoretical guarantees are rare
- algorithms are computationally expensive, often require costly iteration

Phaseless reconstruction/Phase retrieval



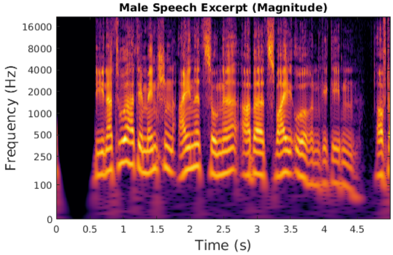
- complex-valued representations as magnitude and phase
- modification/generation of magnitudes comparably easy
- phase invalid after magnitude-based processing
- signal reconstruction from magnitude is a challenging problem
- theoretical guarantees are rare
- algorithms are computationally expensive, often require costly iteration

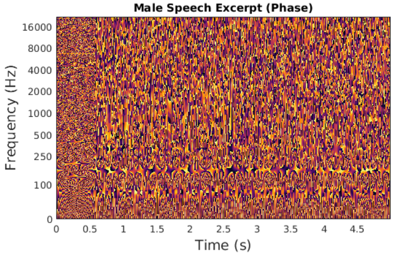
Phaseless reconstruction/Phase retrieval



- complex-valued representations as magnitude and phase
- modification/generation of magnitudes comparably easy
- phase invalid after magnitude-based processing
- signal reconstruction from magnitude is a challenging problem
- theoretical guarantees are rare
- algorithms are computationally expensive, often require costly iteration

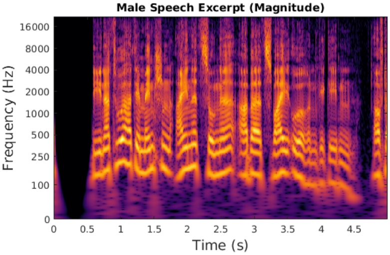
Phaseless reconstruction/Phase retrieval

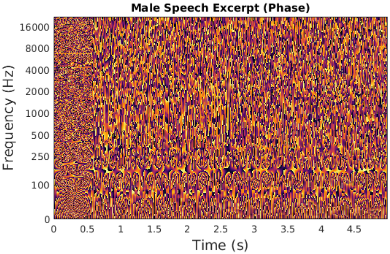




- complex-valued representations as magnitude and phase
- modification/generation of magnitudes comparably easy
- phase invalid after magnitude-based processing
- signal reconstruction from magnitude is a challenging problem
- theoretical guarantees are rare
- algorithms are computationally expensive, often require costly iteration

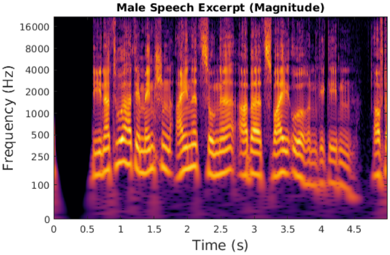
Phaseless reconstruction/Phase retrieval

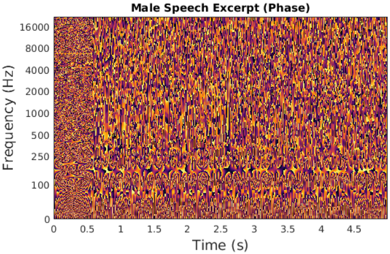




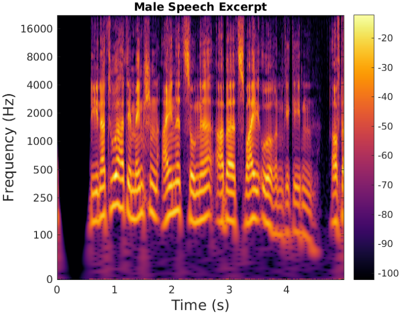
- complex-valued representations as magnitude and phase
- modification/generation of magnitudes comparably easy
- phase invalid after magnitude-based processing
- signal reconstruction from magnitude is a challenging problem
- theoretical guarantees are rare
- algorithms are computationally expensive, often require costly iteration

Phaseless reconstruction/Phase retrieval



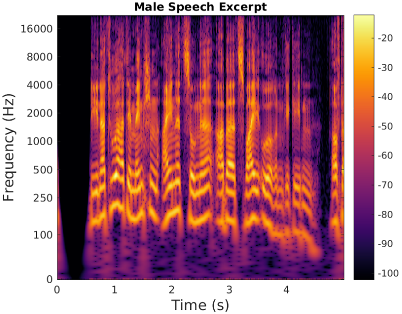


- complex-valued representations as magnitude and phase
- modification/generation of magnitudes comparably easy
- phase invalid after magnitude-based processing
- signal reconstruction from magnitude is a challenging problem
- theoretical guarantees are rare
- algorithms are computationally expensive, often require costly iteration

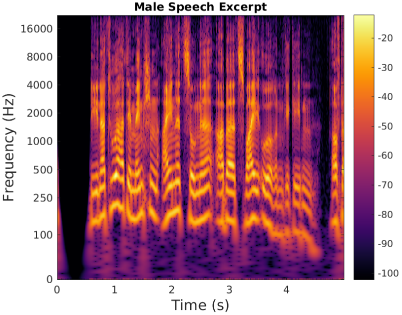


- redundant wavelet/constant-Q transforms
- (usually) fixed number of channels per octave
- possibility of decimation
- invertibility (in the frequency range of interest)
- high-fidelity synthesis requires good magnitude and phase estimates

Wavelet transforms

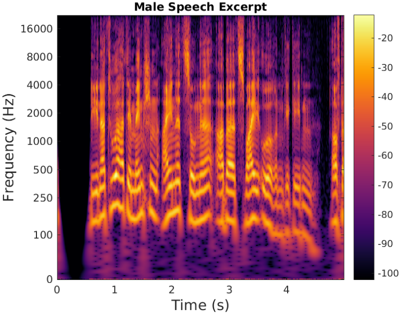


- redundant wavelet/constant-Q transforms
- (usually) fixed number of channels per octave
- possibility of decimation
- invertibility (in the frequency range of interest)
- high-fidelity synthesis requires good magnitude and phase estimates



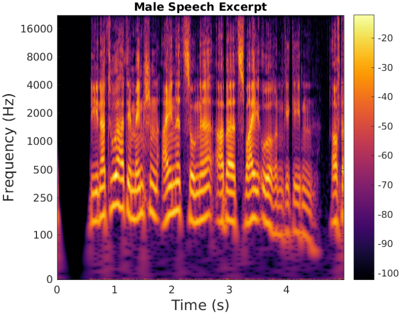
- redundant wavelet/constant-Q transforms
- (usually) fixed number of channels per octave
- possibility of decimation
- invertibility (in the frequency range of interest)
- high-fidelity synthesis requires good magnitude and phase estimates

Wavelet transforms



- redundant wavelet/constant-Q transforms
- (usually) fixed number of channels per octave
- possibility of decimation
- invertibility (in the frequency range of interest)
- high-fidelity synthesis requires good magnitude and phase estimates

Wavelet transforms



- redundant wavelet/constant-Q transforms
- (usually) fixed number of channels per octave
- possibility of decimation
- invertibility (in the frequency range of interest)
- high-fidelity synthesis requires good magnitude and phase estimates

Problems of wavelet phaseless reconstruction

- generic algorithms usually require expensive iteration
- wavelet-specific algorithms are rare
- solutions are not unique, at best up to an additive constant
- convergence to a good solution can rarely be guaranteed
- algorithms often introduce audible artifacts, even under optimal conditions
- provably unstable in areas of small magnitude

Problems of wavelet phaseless reconstruction

- generic algorithms usually require expensive iteration
- wavelet-specific algorithms are rare
- solutions are not unique, at best up to an additive constant
- convergence to a good solution can rarely be guaranteed
- algorithms often introduce audible artifacts, even under optimal conditions
- provably unstable in areas of small magnitude

Problems of wavelet phaseless reconstruction

- generic algorithms usually require expensive iteration
- wavelet-specific algorithms are rare
- solutions are not unique, at best up to an additive constant
- convergence to a good solution can rarely be guaranteed
- algorithms often introduce audible artifacts, even under optimal conditions
- provably unstable in areas of small magnitude

Problems of wavelet phaseless reconstruction

- generic algorithms usually require expensive iteration
- wavelet-specific algorithms are rare
- solutions are not unique, at best up to an additive constant
- convergence to a good solution can rarely be guaranteed
- algorithms often introduce audible artifacts, even under optimal conditions
- provably unstable in areas of small magnitude

Problems of wavelet phaseless reconstruction

- generic algorithms usually require expensive iteration
- wavelet-specific algorithms are rare
- solutions are not unique, at best up to an additive constant
- convergence to a good solution can rarely be guaranteed
- algorithms often introduce audible artifacts, even under optimal conditions
- provably unstable in areas of small magnitude

Problems of wavelet phaseless reconstruction

- generic algorithms usually require expensive iteration
- wavelet-specific algorithms are rare
- solutions are not unique, at best up to an additive constant
- convergence to a good solution can rarely be guaranteed
- algorithms often introduce audible artifacts, even under optimal conditions
- provably unstable in areas of small magnitude

The phase-magnitude relationship for wavelet transforms

The wavelet transform of a signal s with respect to the mother wavelet ψ is

$$W_\psi s(x,y) = \frac{1}{\sqrt{y}} \int_{\mathbb{R}} s(t)\overline{\psi\left(\frac{t-x}{y}\right)}\, dt, \text{ for all } x \in \mathbb{R}, y \in \mathbb{R}^+, \tag{11}$$

with magnitude $M_\psi^s := |W_\psi s| \geq 0$ and phase and $\phi_\psi^s := \arg(W_\psi s) \in \mathbb{R}$. If

$$\hat{\psi}(\xi) = \begin{cases} \xi^{\frac{\alpha-1}{2}} e^{-2\pi\xi} e^{i\beta\log\xi} & \xi \in \mathbb{R}^+, \\ 0 & \text{otherwise,} \end{cases} \tag{12}$$

for some $\alpha > -1, \beta \in \mathbb{R}$, then

$$\begin{aligned} \frac{\partial}{\partial x}\phi_\psi^s(x,y) &= -\frac{\partial}{\partial y}\log(M_\psi^s)(x,y) + \frac{\alpha}{2y} \text{ and} \\ \frac{\partial}{\partial y}\phi_\psi^s(x,y) &= \frac{\partial}{\partial x}\log(M_\psi^s)(x,y) - \frac{\beta}{y}. \end{aligned} \tag{13}$$

The phase can be recovered from the magnitude up to a constant by simple integration.

Phase reconstruction by differentiation and integration

In practice:

- wavelet transform obtained from a sampled (discrete) signal and wavelet
- only finitely many wavelet coefficients are available
- phase reconstruction is unstable when the magnitude is small

A solution:

- approximate differentiation by finite differences
- approximate integration by an easy quadrature rule
- adaptive, patchwise integration
  - begin at points of large magnitude
  - avoid areas of small magnitude
  - use heap of positions sorted by magnitude

Phase reconstruction by differentiation and integration

In practice:

- wavelet transform obtained from a sampled (discrete) signal and wavelet
- only finitely many wavelet coefficients are available
- phase reconstruction is unstable when the magnitude is small

A solution:

- approximate differentiation by finite differences
- approximate integration by an easy quadrature rule
- adaptive, patchwise integration
  - begin at points of large magnitude
  - avoid areas of small magnitude
  - use heap of positions sorted by magnitude

Phase reconstruction by differentiation and integration

In practice:

- wavelet transform obtained from a sampled (discrete) signal and wavelet
- only finitely many wavelet coefficients are available
- phase reconstruction is unstable when the magnitude is small

A solution:

- approximate differentiation by finite differences
- approximate integration by an easy quadrature rule
- adaptive, patchwise integration
  - begin at points of large magnitude
  - avoid areas of small magnitude
  - use heap of positions sorted by magnitude

Phase reconstruction by differentiation and integration

In practice:

- wavelet transform obtained from a sampled (discrete) signal and wavelet
- only finitely many wavelet coefficients are available
- phase reconstruction is unstable when the magnitude is small

A solution:

- approximate differentiation by finite differences
- approximate integration by an easy quadrature rule
- adaptive, patchwise integration
  - begin at points of large magnitude
  - avoid areas of small magnitude
  - use heap of positions sorted by magnitude

Phase reconstruction by differentiation and integration

In practice:

- wavelet transform obtained from a sampled (discrete) signal and wavelet
- only finitely many wavelet coefficients are available
- phase reconstruction is unstable when the magnitude is small

A solution:

- approximate differentiation by finite differences
- approximate integration by an easy quadrature rule
- adaptive, patchwise integration
  - begin at points of large magnitude
  - avoid areas of small magnitude
  - use heap of positions sorted by magnitude

Phase reconstruction by differentiation and integration

In practice:

- wavelet transform obtained from a sampled (discrete) signal and wavelet
- only finitely many wavelet coefficients are available
- phase reconstruction is unstable when the magnitude is small

A solution:

- approximate differentiation by finite differences
- approximate integration by an easy quadrature rule
- adaptive, patchwise integration
  - begin at points of large magnitude
  - avoid areas of small magnitude
  - use heap of positions sorted by magnitude

The algorithm - Phase Gradient Heap integration (PGHI)

Input: Magnitude M_s of wavelet coefficients, estimates $\Delta_\psi^{\tilde{\phi},x,s}$ and $\Delta_\psi^{\tilde{\phi},\xi,s}$ of the partial phase derivatives, relative tolerance tol.

Output: Phase estimate $(\tilde{\phi}_\psi^s)_{\text{est}}$.

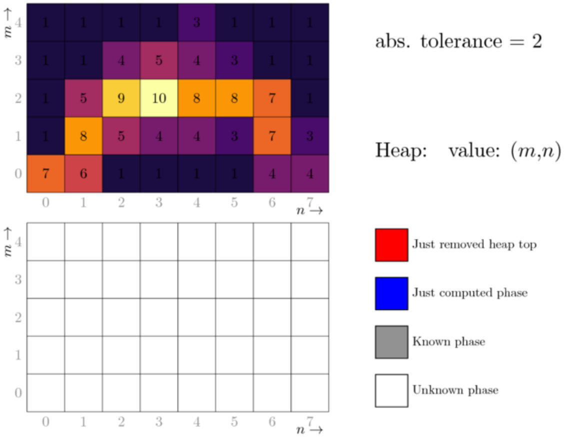
$abstol \leftarrow tol \cdot \max\left(M_s[n,k]\right)$;
Create set $\mathcal{I} = \{(n,k) : M_s[n,k] > abstol\}$;
Assign random values to $(\tilde{\phi}_\psi^s)_{\text{est}}(n,k)$ for $(n,k) \notin \mathcal{I}$;
Construct a self-sorting max heap for (n,k) pairs;
while $\mathcal{I}$ is not $\emptyset$ **do**
    if heap is empty **then**
        Move $(n_m, k_m) = \arg\max_{(n,k)\in\mathcal{I}} \left(M_s[n,k]\right)$ from $\mathcal{I}$ into the heap;
        $(\tilde{\phi}_\psi^s)_{\text{est}}(n_m, k_m) \leftarrow 0$;
    end
    while heap is not empty **do**
        $(n,k) \leftarrow$ remove the top of the heap;
        foreach $(n_{\mathrm{n}}, k_{\mathrm{n}})$ in $\mathcal{N}_{n,k} \cap \mathcal{I}$ **do**

$$(\tilde{\phi}_\psi^s)_{\mathrm{est}}[n_{\mathrm{n}}, k_{\mathrm{n}}] = (\tilde{\phi}_\psi^s)_{\mathrm{est}}[n,k] + \frac{\xi_{k_{\mathrm{n}}} - \xi_k}{2}\left(\Delta_\psi^{\tilde{\phi},\xi,s}[n,k] + \Delta_\psi^{\tilde{\phi},\xi,s}[n_{\mathrm{n}}, k_{\mathrm{n}}]\right) + \frac{a_{\mathrm{d}}(n_{\mathrm{n}} - n)}{2\xi_{\mathrm{s}}}\left(\Delta_\psi^{\tilde{\phi},x,s}[n,k] + \Delta_\psi^{\tilde{\phi},x,s}[n_{\mathrm{n}}, k_{\mathrm{n}}]\right).;$$

            Move $(n_{\mathrm{n}}, k_{\mathrm{n}})$ from $\mathcal{I}$ into the heap;
        end
    end
end

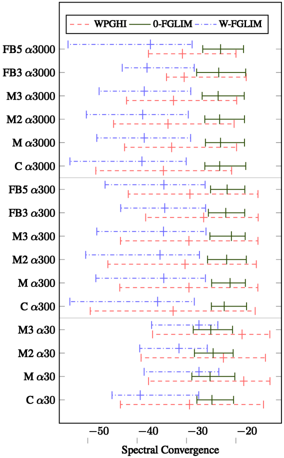
Algorithm 1: Wavelet Phase Gradient Heap Integration

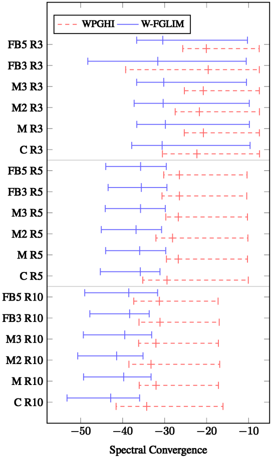
Phase Gradient Heap integration visualized



Step 0: Unknown phase

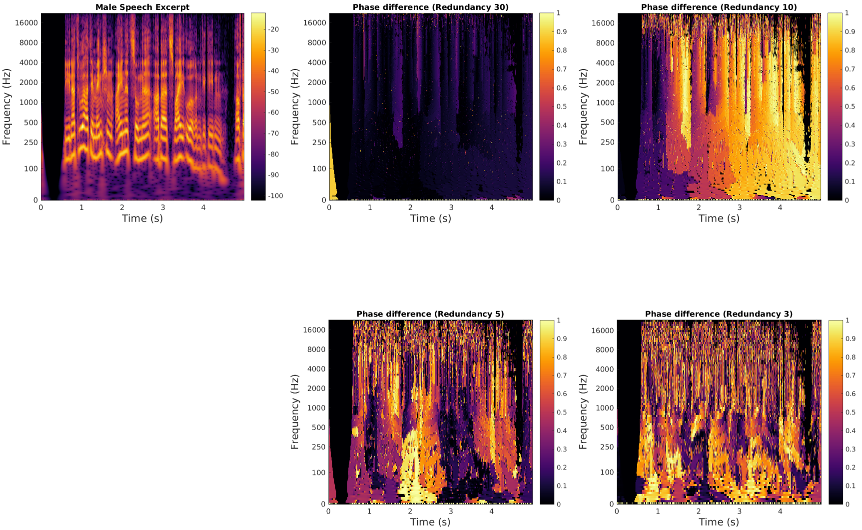
Numerical evaluation





Numerical evaluation on a subset of the SQAM dataset for wavelet PGHI, fast Griffin-Lim (0-initialization), fast Griffin-Lim (WPGHI-initialization). Spectral convergence measures the relative spectral error in dB.

Phase differences vs. redundancy



A continuous STFT for discrete signals

A continuous STFT for discrete signals

- Consider finite signals as periodic δ-trains (tempered distributions):

$$(a_1, \ldots, a_N) \sim \sum_{n=1}^{N} a_n \epsilon_n =: \phi \quad \text{with } \epsilon_n := \sum_{k \in \mathbb{Z}} \delta_{\frac{n}{N}+k}$$

- Take a nice window function (e.g., Gaussian) g.

Then

$$\mathbf{V}_g \varphi(x, \xi) = \sum_{n=0}^{N-1} a_n e^{-2\pi i \xi \frac{n}{N}} \mathbf{Z}\overline{g}\left(\frac{n}{N} - x, \xi\right), \tag{14}$$

where $\mathbf{Z}$ denotes the Zak transform.

Relation to the discrete STFT

With $\mathrm{f_N} = (a_1, \ldots, a_N)^T \in \mathbb{C}^N$, we have

$$\mathbf{V}_g\varphi(x,\xi) = \exp(-2\pi i r_\xi n_x/N) \cdot \mathbf{STFT}_{\mathrm{g_N}^{(r_x,r_\xi)}} \mathrm{f_N}[n_x, m_\xi], \quad (15)$$

where $x = n_x/N + r_x$ and, $\xi = m_\xi + r_\xi$, with $r_x \in [0, 1/N)$, $r_\xi \in [0,1)$, and

$$\mathrm{g_N}^{(r_x,r_\xi)} = \mathbf{P}_N(\mathbf{M}_{r_\xi}\mathbf{T}_{r_x} g),$$

with the periodization-and-sampling operator $\mathbf{P}_N f[n] = \sum_{j\in\mathbb{Z}} f(n/N - j)$.
In particular, we have

$$\mathbf{V}_g\varphi(k/N, l) = \mathbf{STFT}_{\mathrm{g_N}} \mathrm{f_N}[k,l], \quad \text{with } \mathrm{g_N} = \mathbf{P}_N g.$$

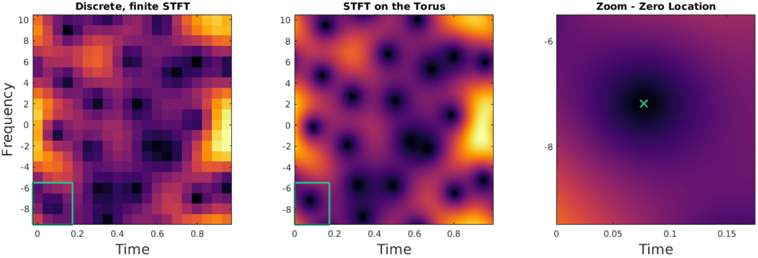


Figure: Comparison between the spectrograms obtained from the discrete STFT and the STFT on the torus, and their application for detecting zeros of the STFT: Here, f_N was sampled from a complex Gaussian noise process $\mathcal{N}(0, I_N) + i\mathcal{N}(0, I_N)$, $N = 20$. The right panel shows a zoom into the lower-left part of the spectrogram (indicated by the teal border), where the location of a spectrogram zero is marked.