

Real-Time Spectrogram Inversion Using Phase Gradient Heap Integration

Zdeněk Průša¹ and Peter L. Søndergaard²

^{1,2}Acoustics Research Institute, Austrian Academy of Sciences, Vienna, Austria

² Oticon A/S, Smørum, Denmark





- 1 Motivation
- 2 Algorithm description
- 3 Results
- 4 Live demonstration



Nontrivial modification of spectrogram:

- Time stretching/pitch shifting
- Source separation/denoising
- Missing data inpainting
- Comb filter-free channel mixing

Short-time Fourier transform:

$$(\mathcal{V}_g f)(\omega, t) = \int_{\mathbb{R}} f(\tau + t)g(\tau)e^{-i2\pi\omega\tau} d\tau, \quad \omega, t \in \mathbb{R} \quad (1)$$

$$= M_g^f(\omega, t)e^{i\Phi_g^f(\omega, t)}, \quad (2)$$

Clearly:

$$\log(\mathcal{V}_g f)(\omega, t) = \log M_g^f(\omega, t) + i\Phi_g^f(\omega, t). \quad (3)$$

Gaussian window: $\varphi_{\gamma}(t) = \left(\frac{\gamma}{2}\right)^{-\frac{1}{4}} e^{-\pi\frac{t^2}{\gamma}}$

Log-magnitude gradient: $\nabla \log M_{\varphi_{\gamma}}^f(\omega, t) = \left(\frac{\partial \log M_{\varphi_{\gamma}}^f(\omega, t)}{\partial \omega}, \frac{\partial \log M_{\varphi_{\gamma}}^f(\omega, t)}{\partial t} \right)$

Phase gradient: $\nabla \Phi_{\varphi_{\gamma}}^f(\omega, t) = \left(\frac{\partial \Phi_{\varphi_{\gamma}}^f(\omega, t)}{\partial \omega}, \frac{\partial \Phi_{\varphi_{\gamma}}^f(\omega, t)}{\partial t} \right)$

Relationship between the gradients^{1,2,3}

$$\frac{\partial \Phi_{\varphi_{\gamma}}^f(\omega, t)}{\partial \omega} = -\gamma \frac{\partial}{\partial t} \log M_{\varphi_{\gamma}}^f(\omega, t) \quad (4)$$

$$\frac{\partial \Phi_{\varphi_{\gamma}}^f(\omega, t)}{\partial t} = \frac{1}{\gamma} \frac{\partial}{\partial \omega} \log M_{\varphi_{\gamma}}^f(\omega, t) + 2\pi\omega. \quad (5)$$

- 1 Michael R. Portnoff, "Magnitude-phase relationships for short-time Fourier transforms based on Gaussian analysis windows," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '79*, Apr. 1979, vol. 4, pp. 186–189.
- 2 F. Auger, E. Chassande-Mottin, and P. Flandrin, "On phase-magnitude relationships in the short-time Fourier transform," *IEEE Signal Processing Letters*, vol. 19, no. 5, pp. 267–270, May 2012
- 3 Zdeněk Průša, Peter Balazs, and Peter L. Søndergaard, "A Non-iterative Method for (Re)Construction of Phase from STFT magnitude," 2016, Preprint available from <http://lftfat.github.io/notes/lftfatnote040.pdf>.

Gradient theorem (aka Fundamental theorem of calculus for line integrals):

$$\Phi_{\varphi_\gamma}^f(\omega, t) - \Phi_{\varphi_\gamma}^f(\omega_0, t_0) = \int_0^1 \nabla \Phi_{\varphi_\gamma}^f(r(\tau)) \cdot \frac{dr}{d\tau}(\tau) d\tau, \quad (6)$$

where $r(\tau) = [r_\omega(\tau), r_t(\tau)]$ is a parametric representation of any curve starting at (ω_0, t_0) and ending at (ω, t) .

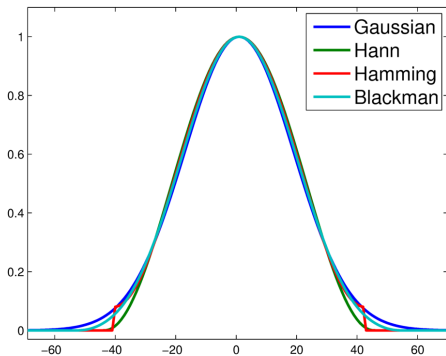
Discrete STFT

$$(V_g f)(m, n) = \sum_{l \in \mathcal{J}} f(l + na)g(l)e^{-i2\pi ml/M} \quad (7)$$

$$= s(m, n)e^{i\phi(m, n)} \quad (8)$$

γ for non-Gaussian windows

$$\gamma = C_g \cdot \text{len}(g)^2 \quad (9)$$



The (scaled) discrete STFT phase gradient $\nabla\phi = (\phi_\omega, \phi_t)$ is estimated from $s_{\log} = \log(s)$ using centered finite differences.

Algorithm 1: Phase gradient heap integration – PGHI

Input: Phase gradient $\nabla\phi(m, n) = (\phi_\omega(m, n), \phi_t(m, n))$, magnitude of coefficients $s(m, n)$, relative tolerance tol .

Output: Estimate of the DGT phase $\hat{\phi}(m, n)$.

Set $\mathcal{J} = \{(m, n) : s(m, n) > tol \cdot \max(s(m, n))\}$;

Assign random values to $\hat{\phi}(m, n)_{(m, n) \notin \mathcal{J}}$;

Construct a self-sorting *heap* for (m, n) pairs;

while \mathcal{J} is not \emptyset **do**

if *heap* is empty **then**

 Insert $(m, n)_{\max} = \arg \max (s(m, n)_{(m, n) \in \mathcal{J}})$ into the *heap*;

$\hat{\phi}(m, n)_{\max} \leftarrow 0$ and remove $(m, n)_{\max}$ from \mathcal{J} ;

end

while *heap* is not empty **do**

$(m, n) \leftarrow$ remove the top of the *heap*;

if $(m + 1, n) \in \mathcal{J}$ **then**

$\hat{\phi}(m + 1, n) \leftarrow \hat{\phi}(m, n) + \frac{1}{2} (\phi_\omega(m, n) + \phi_\omega(m + 1, n))$;

 Remove $(m + 1, n)$ from \mathcal{J} and insert it into *heap*;

end

if $(m - 1, n) \in \mathcal{J}$ **then**

$\hat{\phi}(m - 1, n) \leftarrow \hat{\phi}(m, n) - \frac{1}{2} (\phi_\omega(m, n) + \phi_\omega(m - 1, n))$;

 Remove $(m - 1, n)$ from \mathcal{J} and insert it into *heap*;

end

end

end

if $(m, n + 1) \in \mathcal{J}$ **then**

$\hat{\phi}(m, n + 1) \leftarrow \hat{\phi}(m, n) + \frac{1}{2} (\phi_t(m, n) + \phi_t(m, n + 1))$;

 Remove $(m, n + 1)$ from \mathcal{J} and insert it into *heap*;

end

if $(m, n - 1) \in \mathcal{J}$ **then**

$\hat{\phi}(m, n - 1) \leftarrow \hat{\phi}(m, n) - \frac{1}{2} (\phi_t(m, n) + \phi_t(m, n - 1))$;

 Remove $(m, n - 1)$ from \mathcal{J} and insert it into *heap*;

end

Algorithm 2: Real-Time Phase Gradient Heap Integration for n -th frame

Input: Phase time derivative $\phi_t(m, n)$ and magnitude $s(m, n)$ of frames n and $n - 1$, phase frequency derivative $\phi_\omega(m, n)$ for frame n , estimated phase $\hat{\phi}(m, n)$ for frame $n - 1$ and relative tolerance tol .

Output: Phase estimate $\hat{\phi}(m, n)$ for frame n .

$abstol \leftarrow tol \cdot \max(s(m, n) \cup s(m, n - 1))$;

Create set $\mathcal{J} = \{(m, n) : s(m, n) > abstol\}$;

Assign random values to $\hat{\phi}(m, n)_{(m, n) \notin \mathcal{J}}$;

Construct a self-sorting *heap* for (m, n) tuples;

Insert $(m, n - 1)$ for $m = (m : s(m, n - 1) > abstol)$ into the *heap*;

while \mathcal{J} is not \emptyset **do**

while *heap* is not empty **do**

$(m_{heap}, n_{heap}) \leftarrow$ remove the top of the *heap*;

if $n_{heap} == n - 1$ **then**

if $(m_{heap}, n) \in \mathcal{J}$ **then**

$\hat{\phi}(m_{heap}, n) \leftarrow$

$\hat{\phi}(m_{heap}, n - 1) + \frac{1}{2} (\phi_t(m_{heap}, n - 1) + \phi_t(m_{heap}, n))$;

 Insert (m_{heap}, n) into the *heap* and remove it from \mathcal{J} ;

end

end

end

end

if $n_{heap} == n$ **then**

if $(m_{heap} + 1, n) \in \mathcal{J}$ **then**

$\hat{\phi}(m_{heap} + 1, n) \leftarrow$

$\hat{\phi}(m_{heap}, n) + \frac{1}{2} (\phi_\omega(m_{heap}, n) + \phi_\omega(m_{heap} + 1, n))$;

 Insert $(m_{heap} + 1, n)$ into the *heap* and remove it from \mathcal{J} ;

end

if $(m_{heap} - 1, n) \in \mathcal{J}$ **then**

$\hat{\phi}(m_{heap} - 1, n) \leftarrow$

$\hat{\phi}(m_{heap}, n) - \frac{1}{2} (\phi_\omega(m_{heap}, n) + \phi_\omega(m_{heap} - 1, n))$;

 Insert $(m_{heap} - 1, n)$ into the *heap* and remove it from \mathcal{J} ;

end

end

Comparison with SPSI and RTISI-LA

EBU SQAM database, $M = 2048$



Overlap 3/4	Gauss	Hann	Hamming	Blackman
SPSI	-17.82	-16.72	-16.53	-17.62
Proposed (0)	-17.76	-17.50	-17.58	-17.82
RTISI (0)	-22.80	-21.75	-21.08	-22.82
Proposed (1)	-20.91	-20.25	-20.07	-20.76
RTISI-LA (1)	-26.08	-25.14	-24.01	-26.63

Overlap 15/16	Gauss	Hann	Hamming	Blackman
SPSI	-17.99	-17.16	-16.82	-17.92
Proposed (0)	-26.13	-23.48	-22.33	-25.93
RTISI (0)	-20.18	-19.70	-19.01	-20.35
Proposed (1)	-26.83	-23.50	-22.47	-26.21
RTISI-LA (1)	-17.85	-16.66	-16.12	-17.71

Overlap 7/8	Gauss	Hann	Hamming	Blackman
SPSI	-17.88	-17.09	-16.79	-17.79
Proposed (0)	-21.79	-21.10	-21.01	-21.78
RTISI (0)	-21.15	-20.20	-19.53	-21.07
Proposed (1)	-24.87	-22.74	-21.98	-24.59
RTISI-LA (1)	-22.11	-19.90	-19.14	-21.90

$$E = 20 \log_{10} \frac{\|s - |V_g \hat{f}|\|_{\text{fro}}}{\|s\|_{\text{fro}}}$$

SPSI: Gerald T. Beaugerard, Mithila Harish, and Lonce Wyse, "Single pass spectrogram inversion," in *IEEE International Conference on Digital Signal Processing (DSP)*, 2015, July 2015, pp. 427–431.

RTISI(-LA): Xinglei Zhu, Gerald T. Beaugerard, and Lonce Wyse, "Real-time signal estimation from modified short-time Fourier transform magnitude spectra," in *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1645–1653, July 2007.

DEMO

Complementary material (Matlab code, extended comparison, sound examples):

`http://ltfat.github.io/notes/043`

Matlab/GNU Octave toolboxes:

- LTFAT – Large Time-Frequency Analysis Toolbox

`http://ltfat.github.io`

- PHASERET – Phase Retrieval Toolbox

`http://ltfat.github.io/phaseret`

Thank you for your attention.

Questions?