Towards High Quality Real-Time Signal Reconstruction from STFT Magnitude

Zdeněk Průša

***Abstract*—An efficient algorithm for real-time signal reconstruction from the magnitude of the Short-time Fourier transform (STFT) is presented. The algorithm combines advantages of two previously published algorithms and the reconstructed signals exhibit very good perceptual quality.**

We present an extensive comparison with the state-of-the-art algorithms which shows that the proposed method outperforms others by far in settings capable of producing high quality signals.

***Index Terms*—Time-frequency, Short-time Fourier transform, phase reconstruction, real-time**

I. Introduction

In time-frequency signal processing, it is a common practice to work with the magnitude of STFT only. However, as soon as the reconstruction is desired, one has to deal with phase. Often, it is sufficient to reuse the unmodified phase to recover the signal [1], however some spectrogram modifications might invalidate the phase and the reconstruction procedure may therefore lead to disturbing artifacts [2]. In some cases, the original phase is not available at all [3]. STFT phase retrieval algorithms alleviate the problem by allowing disposal of the existing phase completely and constructing a new valid phase from scratch taking the modified magnitude.

Unfortunately, to date, available STFT phase retrieval algorithms cannot always be expected to be sufficient in all respects. For example, some algorithms require the knowledge of the whole signal and they typically need a large number of costly iterations to produce a good result [4], [5], [6], [7]. This fact disqualifies them from being used in any real-time or interactive applications. Algorithms which are capable of processing signals in real-time i.e. in the frame-by-frame manner with bounded delay [8], [9], [10], [11], tend to produce noticeable artefacts such as “phasiness”[2], metallic ringing, echo etc. for specific classes of audio signals.

In this work, we propose a real-time phase reconstruction algorithm which outperforms the state-of-the-art algorithms by a large margin in the senses of both the objective error measure and the perceived quality of the reconstruction.

We compare our method with the following algorithms, which can be considered to be the state-of-the-art. The Real-Time Iterative Spectrogram Inversion (RTISI) algorithm was first published in [12] and later improved by including look-ahead frames in [13], [8] (RTISI-LA). The algorithm was later modified slightly in the line of work of Gnann and Spiertz in [14], [15], [16] (GSRTISI-LA). From the point of view of this paper, the crucial property of GSRTISI-LA is that it allows defining initial estimate of the phase of the newest look-ahead frame.

Another algorithm called Real-Time Phase Gradient Heap Integration (RTPGHI) was presented in [17]. It takes completely different approach and its main property is that it is not iterative and therefore efficient. As it turns out, it is also a suitable candidate for providing the initial phase guess for GSRTISI-LA.

In this paper, we combine strengths of both RTPGHI and GSRTISI-LA to perform high quality signal reconstruction. To that end, in addition to the RTPGHI initialization, we further generalize GSRTISI-LA such that the analysis window, window overlap, number of frequency bins and the number of look-ahead frames can be chosen freely and independent of each other (to the extent specified in the next section) in our variant of the algorithm.

Because we aim for a high reconstruction quality, we exclude algorithms such as the ones presented in [11], [9], [10] from the comparisons in this study. Although they are much faster than the iterative algorithms, they simply do not produce results of sufficient quality.

In the spirit of reproducible research, the implementation of the algorithms, audio examples as well as scripts reproducing the experiments from this manuscript are freely available at http://ltfat.github.io/notes/048. The code depends on our Matlab/GNU Octave[18] packages LTFAT [19], [20] (version 2.1.3 or above) and PHASERET (version 0.2.0 or above). Both toolboxes are open-source and they can be obtained from http://ltfat.github.io and http://ltfat.github.io/phaseret, respectively.

The paper is organized as follows. In section II we recall essential formulas for computing STFT analysis and synthesis, section III contains description of the proposed algorithm and, finally, in section IV we compare the proposed method with the state-of-the-art algorithms.

II. STFT and its Inverse

The discrete STFT of an input signal $f \in \ell^2(\mathbb{Z})$ using analysis window $g \in \ell^2(\mathbb{Z})$ is defined as

$$c_n(m) = \left(\mathcal{V}_g f\right)_n(m) = \sum_{l\in\mathbb{Z}} f(l+na)\overline{g(l)}\mathrm{e}^{-\mathrm{i}2\pi ml/M}, \quad (1)$$

where overline denotes complex conjugation, M is a finite number of frequency channels indexed as $m = 0, \dots, M-1$, $n \in \mathbb{Z}$ is a time-frame index and the parameter a is the time step (window shift) in samples. The window g will further

Z. Průša* is with the Acoustics Research Institute, Austrian Academy of Sciences, Wohllebengasse 12–14, 1040 Vienna, Austria, email: zdenek.prusa@oeaw.ac.at (corresponding address).

Manuscript received April 19, 2005; revised August 26, 2015.

This work was supported by the Austrian Science Fund (FWF) START-project FLAME (“Frames and Linear Operators for Acoustical Modeling and Parameter Estimation”; Y 551-N13).

be considered to be real, whole-point symmetric and finitely supported such that the sum index set can be reduced to

$$\mathcal{I} = \left\{-\lfloor gl/2 \rfloor, \ldots, \lceil gl/2 \rceil - 1\right\}, \tag{2}$$

where gl is the length of the window support and the unique peak of the window is at index $l = 0$.

The synthesis window $\widetilde{g}$ can be obtained as

$$\widetilde{g} = \frac{1}{M}\frac{g}{\sum_{n\in\mathbb{Z}} g^2(\cdot - na)}, \tag{3}$$

if the following conditions are met: support of the window g is less or equal to the number of frequency channels i.e. $gl \leq M$ and there is some window overlap i.e. $gl > a$. Under these assumptions, the sum in the denominator is nonzero and a-periodic, $\widetilde{g}$ and g have equal time support and the following holds

$$\sum_{n\in\mathbb{Z}} (g\widetilde{g})(\cdot - na) \equiv 1/M. \tag{4}$$

Please refer e.g. to [21], [22], [23] for a thorough mathematical treatment of the invertibility of discrete STFT (Discrete Gabor transform) based on the theory of frames. In the terms of Gabor frame theory the window computed using (3) is the *canonical dual window* and the inequality $gl \leq M$ is referred to as the *painless* condition.

Having the synthesis window $\widetilde{g}$, the individual time-frames f_n can be recovered using

$$f_n(l) = \begin{cases} \widetilde{g}(l)\sum_{m=0}^{M-1} c_n(m)\mathrm{e}^{\mathrm{i}2\pi ml/M} & \text{for all } l \in \mathcal{I}, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

A part of the signal recovered from frames up to and including frame N using the overlap-add procedure is given as

$$\widetilde{f}_N = \sum_{n=-\infty}^{N} f_n(\cdot - na) \tag{6}$$

and, clearly, the original signal can be recovered using $f = \widetilde{f}_\infty$.

A. Computing STFT and its Inverse

In this section, we recall how to compute STFT and how to reconstruct the frames according to (1) and (5) using the FFT algorithm. In order to be able to work with finite vectors, we introduce operator $V_\mathcal{I} : \ell^2(\mathbb{Z}) \to \mathbb{R}^{gl}$ which acts as follows

$$(V_\mathcal{I} f)(l) = f(-\lfloor gl/2 \rfloor + l) \text{ for } l = 0, \ldots, gl - 1 \tag{7}$$

and the converse operator $S_\mathcal{I} : \mathbb{R}^{gl} \to \ell^2(\mathbb{Z})$

$$(S_\mathcal{I} f)(-\lfloor gl/2 \rfloor + l) = \begin{cases} f(l) & \text{for } l = 0, \ldots, gl - 1 \\ 0 & \text{all } l \in \mathbb{Z}\backslash\{0, \ldots, gl - 1\}. \end{cases} \tag{8}$$

We further define the “postpad” operator $\mathrm{P}_{K,Q} : \mathbb{R}^K \to \mathbb{R}^Q$

$$\left(\mathrm{P}_{K,Q} f\right)(l) = \begin{cases} f(l) & \text{for } l = 0, \ldots, \min(K, Q) - 1 \\ 0 & \text{for } l = K, \ldots, Q - 1 \text{ if } Q > K \end{cases} \tag{9}$$

and the circular shift operator $\mathrm{C}_k : \mathbb{R}^M \to \mathbb{R}^M$

$$(\mathrm{C}_k f)(l) = f(l - k \bmod M) \text{ for } l = 0, \ldots, M - 1. \tag{10}$$

Equation (1) (without the now superfluous conjugation) can be efficiently evaluated as follows

$$\left(\mathcal{V}_g f\right)_n = \mathcal{F}\,\mathrm{C}_{-\lfloor gl/2 \rfloor}\mathrm{P}_{gl,M}\left(V_\mathcal{I} f(\cdot + na) \cdot V_\mathcal{I} g\right), \tag{11}$$

where $\mathcal{F}$ denotes unscaled discrete Fourier transform

$$(\mathcal{F}f)(m) = \sum_{l=0}^{M-1} f(n)\mathrm{e}^{-\mathrm{i}2\pi ml/M} \text{ for } m = 0, \ldots, M - 1 \tag{12}$$

computed using the FFT algorithm.

Similarly, (5) can be computed by reversing the order of the operations and by inverting their arguments

$$f_n = \mathrm{S}_\mathcal{I}\left(\mathrm{V}_\mathcal{I}\widetilde{g} \cdot \mathrm{P}_{M,gl}\mathrm{C}_{\lfloor gl/2 \rfloor}\mathcal{F}^{-1} c_n\right), \tag{13}$$

where $\mathcal{F}^{-1}$ denotes unscaled inverse discrete Fourier transform

$$\left(\mathcal{F}^{-1} f\right)(l) = \sum_{m=0}^{M-1} f(n)\mathrm{e}^{\mathrm{i}2\pi ml/M} \text{ for } l = 0, \ldots, M-1 \tag{14}$$

computed using the FFT algorithm.

III. Algorithms

Further, we will denote the magnitude of coefficients of n-th frame as $s_n = |c_n|$. The goal of real-time phase reconstruction algorithms is to estimate coefficients $\widetilde{c}_n$, which can in turn be plugged in (5) thus recovering the time frame.

A. Overview of RTPGHI

The RTPGHI algorithm [17] is a real-time capable version of the PGHI algorithm published in [24]. Both algorithms are based on the relationship of the gradients of the phase and the logarithm of the magnitude of STFT and employ an adaptive integration scheme to recover the phase. The best performance can be achieved by using the Gaussian window, but other windows can be used as well. The RTPGHI algorithm comes in two versions requiring one (RTPGHI(1)) or zero (RTPGHI(0)) look-ahead frames respectively. For the details please see the above mentioned references.

B. Generalized GSRTISI-LA with RTPGHI Initialization

The original version of GSRTISI-LA is restricted to a single type of window to be used for both the analysis and the synthesis (i.e. $\widetilde{g} = g$), and, moreover, the authors seem to had always used 75% window overlap and 3 look-ahead frames. In this section, we present a variant of GSRTISI-LA which admits any analysis window and allows free choice of the window overlap length and the number of frequency channels (up to conditions presented in sec. II). As it was already mentioned, our variant employs RTPGHI to estimate the initial phase.

Assuming RTPGHI(1) is used for initialization, the algorithm processes one time-frame at a time, taking into account N_{LA} future frames. $N_{\mathrm{LA}} - 1$ look-ahead frames are used for the basic version of the GSRTISI-LA algorithm, one additional look-ahead frame is required by RTPGHI.

Algorithm 1: RTPGHI(1) + GSRTISI-LA($N_{\mathrm{LA}}-1$), $n-$th time frame

Input: Number of look-ahead frames N_{LA}, number of iterations $itNo$, magnitude of STFT coefficients $s_{n,\ldots,n+N_{\mathrm{LA}}}$
Output: Time frame f_n.

Compute $f_{n+N_{\mathrm{LA}}-1}$ using (5) and coefficients $\widetilde{c}_{n+N_{\mathrm{LA}}-1}$ estimated using the RTPGHI algorithm (requires $s_{n+N_{\mathrm{LA}}}$);
for $it = 1, 2, \ldots, itNo$ **do**
  for $p = N_{\mathrm{LA}}-1, \ldots, 0$ **do**
    Compute $\widetilde{f}_{n+N_{\mathrm{LA}}-1}$ using (6);
    $t \leftarrow \left(\mathcal{V}_{g_p}\widetilde{f}_{n+N_{\mathrm{LA}}-1}\right)_{n+p}$;
    $c_{n+p} \leftarrow s_{n+p}\, t/|t|$;
    Compute f_{n+p} using (5);
  end
end

In addition to windows g and $\widetilde{g}$, the algorithm requires N_{LA} additional analysis windows $g_{0,\ldots,N_{\mathrm{LA}}-1}$ which are obtained as

$$g_k = M\frac{g}{g_{\text{sum}}(\cdot + ka)}, \text{ where } g_{\text{sum}} = \sum_{n=-\infty}^{N_{\mathrm{LA}}-1} (g\widetilde{g})(\cdot - na). \tag{15}$$

The proposed algorithm RTPGHI(1) + GSRTISI-LA($N_{\mathrm{LA}}-1$) is formally written in Alg. 1. An extension to the version RTPGHI(0) + GSRTISI-LA(N_{LA}) is straightforward.

Even though we operate with infinite sum bound, in practice, due to the finite support of the windows, it is sufficient to work with time frames in the range $n-N_{\mathrm{LB}}, \ldots, n+N_{\mathrm{LA}}$, where $N_{\mathrm{LB}} = \lceil gl/a \rceil - 1$ is the number of "look-back" frames.

Please note that other types of phase initialization are indeed possible. The authors of the original version of GSRTISI-LA proposed to do simple *phase unwrapping* in [16]. Another option is to employ different algorithm such as [9] in place of RTPGHI. However, in our experience neither of these two approaches brings considerable improvements over not doing the initialization at all and often, it is even harmful.

C. Real-time Deadline, Delay and Computational Complexity

The basic rule of real-time audio processing is to never cause audio dropouts. That means that the worst case execution time for a single output frame must be less than a/f_{s} (time frame shift divided by the sampling rate) seconds, or even less in some settings. This fact limits the number of iterations $itNo$ which can be done, but the exact number it is entirely dependent on the computing power of the device. Since the number of look-ahead will be varied, we will further use the number of *per-frame* iterations.

The typical delay of the real-time STFT analysis-synthesis scheme is equal to the length of the window gl. Each look-ahead frame of the phase reconstruction algorithm increases the delay by the length of the window shift a, therefore the overall input-output delay is $(gl + aN_{\mathrm{LA}})/f_{\mathrm{s}}$ seconds.

IV. Experiments

In the experiments we used the following error measure, previously refered to as *spectral convergence* [25]

$$\mathcal{C} = \sqrt{\frac{\sum_{n=0,m=0}^{N-1,M-1}\left(s_n(m) - \left|\left(\mathcal{V}_g\widetilde{f}\right)_n(m)\right|\right)^2}{\sum_{n=0,m=0}^{N-1,M-1} s_n(m)^2}}, \tag{16}$$

where s_n is the target magnitude of coefficients of n-th time frame, $\widetilde{f}$ is the reconstructed signal and N denotes the total number of time frames of the finite signal. The transform $\mathcal{V}_g$ uses the same values for the parameters (g, a and M) as the one used to obtain s. Values in decibels are obtained by $20\log_{10}\mathcal{C}$.

For the tests, we used the SQAM database [26] which consists of 70 recordings sampled at 44.1 kHz. Only the first 10 seconds from the first channel of each sound sample was used in the evaluation.

In our experience, a substantial window overlap is necessary in order to produce results of high perceptual quality. Therefore, in our tests, we use 87.5% window overlap, or, more specifically, time step size $a = 256$ in conjunction with the fixed number of frequency channels $M = 2048$, and the Gaussian window as the analysis window truncated at 1% of its height such that $gl = 2048$. Using even higher window overlap further improves the results.

Please note that whenever we refer to the average error in dB we mean $20\log_{10}\frac{1}{70}\sum_{k=1}^{70}\mathcal{C}_k$, where $\mathcal{C}_k$ is the error of k-th sound excerpt obtained from (16). Averaging errors already converted to dB (which is occasionally done in other contributions) produces even better (lower) errors for all the algorithms.

A. Performance Comparison

In the real-time setting, there is room only for a limited number of iterations, but since it is device dependent, we will evaluate the performance of the algorithms for up to 200 per-frame iterations. Fig. 1 shows the average spectral convergence depending on the number of iterations and the number of look-ahead frames $N_{\mathrm{LA}} \in \{0, 1, 2, 3, 7\}$ given in the brackets. $N_{\mathrm{LA}} = 7$ is the maximum number of look-ahead frames which directly overlap with the current frame. The setting was intentionally chosen as the one used in Fig. 4a in [24] to allow a direct comparison. Note that the proposed algorithm with $N_{\mathrm{LA}} = 7$ outperforms even the best of the offline-only algorithms!

From the graphs one can observe that a common behavior of the iterative algorithms is that the average spectral convergence initially decreases rapidly and from a certain number of iterations it starts to levels off. This phenomenon could be explained by the fact that some signals in the database reach "convergence" at some point while others continue to improve.

Further, one can observe that the proposed algorithm clearly outperforms others in settings using 2 or more look-ahead frames.

The scores for individual files for $N_{\mathrm{LA}} = 2$ ($N_{\mathrm{LA}} = 1$ in case of RTPGHI) and 24 per-frame iterations, as well as

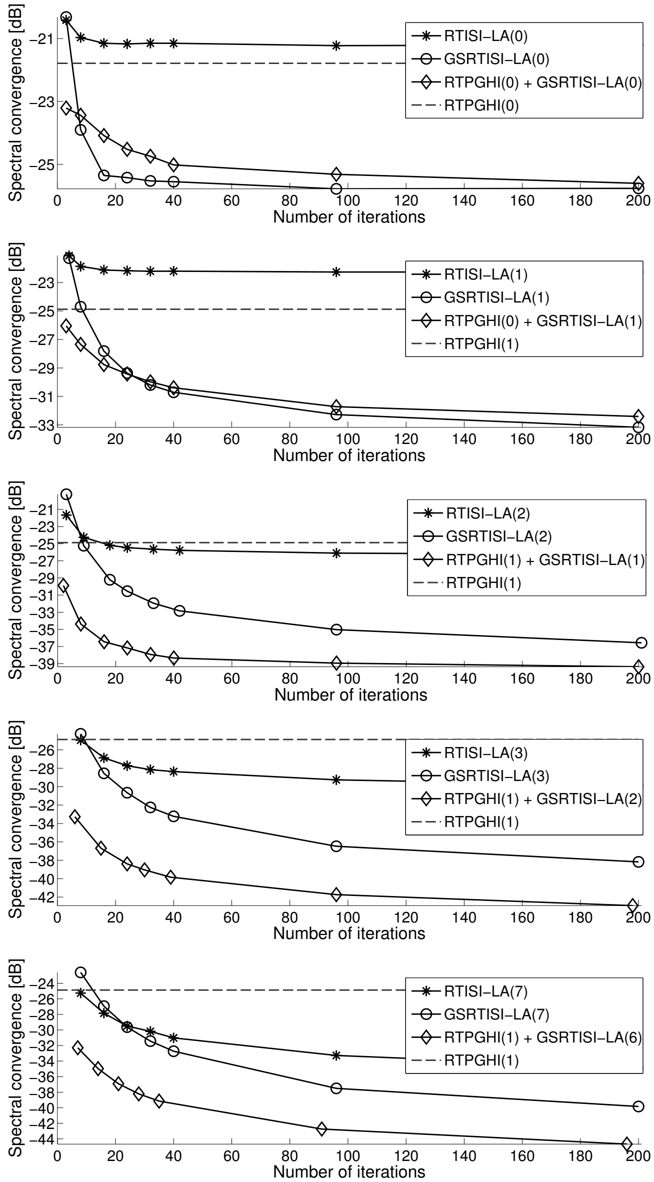

Fig. 1: Comparison of algorithms.

sound examples, can be found at the accompanying web page http://ltfat.github.io/notes/048. Additionally, a box plot of the results is depicted in Fig. 2.

When inspecting results obtained for individual sound excerpts, one can notice that the iterative algorithms struggle with reconstructing recordings of percussion instruments such as claves and castanets and also with attacks of transients in general. Conveniently, the RTPGHI algorithm performs very well in such cases and the combination with GSRTISI-LA inherits and even improves upon the behavior as indicated by the low maximum error in Fig. 2.

A real-time demo allowing one-to-one comparison of the algorithms is available in the PHASERET toolbox as `demo_blockproc_phaseret2.m`.

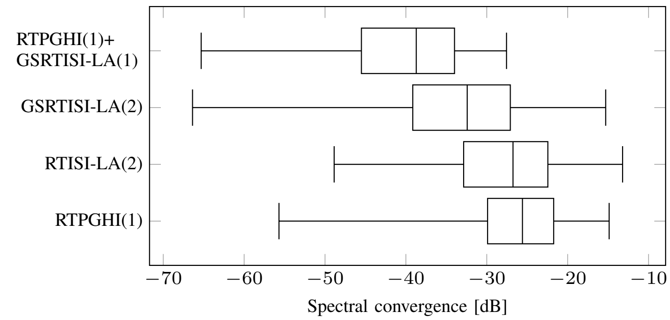

Fig. 2: Box plot of the errors obtained for $N_{\mathrm{LA}} = 2$ ($N_{\mathrm{LA}} = 1$ in case of RTPGHI). The whiskers denote the minimum and the maximum from the 70 sound excerpts.

V. Conclusion

It has been shown that the combination of GSRTISI-LA and RTPGHI outperforms either of the individual algorithms as well as the RTISI-LA algorithm as soon as enough look-ahead frames is used (see Fig. 1).

Although we have only shown objective error measures in this paper, according to our informal listening tests, the quality of the reconstructed signal reflects the error measure improvement. An interested reader can verify this claim by listening to the sound samples found at the accompanying webpage or by running `demo_blockproc_phaseret2.m` using custom audio examples.

Acknowledgment

The author thanks Pavel Rajmic, Thibaud Necciari and Nicki Holighaus for their valuable comments.

REFERENCES

[1] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *Trans. Audio, Speech and Lang. Proc.*, vol. 15, no. 3, pp. 1066–1074, Mar. 2007. [Online]. Available: http://dx.doi.org/10.1109/TASL.2006.885253

[2] J. Laroche and M. Dolson, "Phase-vocoder: about this phasiness business," in *Applications of Signal Processing to Audio and Acoustics, 1997. 1997 IEEE ASSP Workshop on*, Oct 1997, pp. 4 pp.–.

[3] P. Smaragdis, B. Raj, and M. Shashanka, "Missing data imputation for time-frequency representations of audio signals," *Journal of Signal Processing Systems*, vol. 65, no. 3, pp. 361–370, 2011.

[4] D. Griffin and J. Lim, "Signal estimation from modified short-time Fourier transform," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 32, no. 2, pp. 236–243, Apr 1984.

[5] N. Perraudin, P. Balazs, and P. Søndergaard, "A fast Griffin-Lim algorithm," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), IEEE Workshop on*, Oct 2013, pp. 1–4.

[6] R. Decorsiere, P. Søndergaard, E. MacDonald, and T. Dau, "Inversion of auditory spectrograms, traditional spectrograms, and other envelope representations," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 23, no. 1, pp. 46–56, Jan 2015.

[7] J. Le Roux, H. Kameoka, N. Ono, and S. Sagayama, "Fast signal reconstruction from magnitude STFT spectrogram based on spectrogram consistency," in *Proc. 13th Int. Conf. on Digital Audio Effects (DAFx-10)*, Sep. 2010, pp. 397–403.

[8] X. Zhu, G. T. Beauregard, and L. Wyse, "Real-time signal estimation from modified short-time Fourier transform magnitude spectra," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 5, pp. 1645–1653, July 2007.

[9] G. T. Beauregard, M. Harish, and L. Wyse, "Single pass spectrogram inversion," in *Digital Signal Processing (DSP), IEEE International Conference on*, July 2015, pp. 427–431.

[10] P. Margon, R. Badeau, and B. David, "Phase reconstruction of spectrograms with linear unwrapping: application to audio signal restoration," in *Proc. 23rd European Signal Processing Conference (EUSIPCO 2015)*, Aug 2015.

[11] M. Chami, J. Di Martino, L. Pierron, and E. H. Ibn Elhaj, "Real-Time Signal Reconstruction from Short-Time Fourier Transform Magnitude Spectra Using FPGAs," in *5th. International Conference on Information Systems and Economic Intelligence - SIIE 2012*, Djerba, Tunisia, Feb. 2012. [Online]. Available: https://hal.inria.fr/hal-00761783

[12] G. T. Beauregard, X. Zhu, and L. Wyse, "An efficient algorithm for real-time spectrogram inversion," in *Proc. 8th International Conference on Digital Audio Effects (DAFx-05)*, Sep. 2005.

[13] X. Zhu, G. T. Beauregard, and L. Wyse, "Real-time iterative spectrum inversion with look-ahead," in *Proc. IEEE International Conference on Multimedia and Expo*, 2006.

[14] V. Gnann and M. Spiertz, "Comb-filter free audio mixing using STFT magnitude spectra and phase estimation," in *Proc. 11th Int. Conf. on Digital Audio Effects (DAFx-08)*, Sep. 2008.

[15] ——, "Inversion of STFT magnitude spectrograms with adaptive window lengths," in *Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP '09*, Apr. 2009, pp. 325–328.

[16] ——, "Improving RTISI phase estimation with energy order and phase unwrapping," in *Proc. 13th International Conference on Digital Audio Effects (DAFx-10)*, Sep. 2010.

[17] Z. Průša and P. L. Søndergaard, "Real-Time Spectrogram Inversion Using Phase Gradient Heap Integration," in *Proc. Int. Conf. Digital Audio Effects (DAFx-16)*, Sep 2016.

[18] J. W. Eaton, D. Bateman, S. Hauberg, and R. Wehbring, *GNU Octave version 4.0.0 manual: A high-level interactive language for numerical computations*, 2015. [Online]. Available: http://www.gnu.org/software/octave/doc/interpreter

[19] P. L. Søndergaard, B. Torrésani, and P. Balazs, "The Linear Time Frequency Analysis Toolbox," *International Journal of Wavelets, Multiresolution Analysis and Information Processing*, vol. 10, no. 4, 2012.

[20] Z. Průša, P. L. Søndergaard, N. Holighaus, C. Wiesmeyr, and P. Balazs, "The Large Time-Frequency Analysis Toolbox 2.0," in *Sound, Music, and Motion*, ser. Lecture Notes in Computer Science. Springer International Publishing, 2014, pp. 419–442.

[21] T. Strohmer, *Numerical Algorithms for Discrete Gabor Expansions*. Birkhäuser Boston, 1998, ch. 8, pp. 267–294.

[22] P. L. Søndergaard, "Finite discrete Gabor analysis," Ph.D. dissertation, Technical University of Denmark, 2007, available from: http://ltfat.github.io/notes/ltfatnote003.pdf.

[23] ——, "Efficient algorithms for the discrete Gabor transform with a long FIR window," *J. Fourier Anal. Appl.*, vol. 18, no. 3, pp. 456–470, 2012.

[24] Z. Průša, P. Balazs, and P. L. Søndergaard, "A Non-iterative Method for STFT Phase (Re)Construction," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2016, in preparation. Preprint will be available at http://ltfat.github.io/notes/ltfatnote040.pdf.

[25] N. Sturmel and L. Daudet, "Signal reconstruction from STFT magnitude: A state of the art," *Proc. 14th Int. Conf. Digital Audio Effects (DAFx-11)*, pp. 375–386, 2011.

[26] "Tech 3253: Sound Quality Assessment Material recordings for subjective tests," The European Broadcasting Union, Geneva, Tech. Rep., Sept. 2008. [Online]. Available: https://tech.ebu.ch/docs/tech/tech3253.pdf