FAST SIGNAL RECONSTRUCTION FROM MAGNITUDE STFT SPECTROGRAM BASED ON SPECTROGRAM CONSISTENCY

Jonathan Le Roux[1], *Hirokazu Kameoka*[1], *Nobutaka Ono*[2] *and Shigeki Sagayama*[2]

[1]NTT Communication Science Laboratories, NTT Corporation,
3-1 Morinosato Wakamiya, Atsugi, Kanagawa 243-0198, Japan
[2]Graduate School of Information Science and Technology, The University of Tokyo,
7-3-1 Hongo, Bunkyo-ku, Tokyo 113-8656, Japan
{leroux,kameoka}@cs.brl.ntt.co.jp, {onono,sagayama}@hil.t.u-tokyo.ac.jp

ABSTRACT

The modification of magnitude spectrograms is at the core of many audio signal processing methods, from source separation to sound modification or noise canceling, and reconstructing a natural sounding signal in such situations is thus a very important issue. This article presents recent theoretical and experimental developments on the application to signal reconstruction from a modified magnitude spectrogram of the constraints that an array of complex numbers must verify to be a consistent short-time Fourier transform (STFT) spectrogram, i.e., to be the STFT spectrogram of an actual real-valued signal. We give here further theoretical insights, present several potential variations on our previously introduced algorithm, investigate various techniques to speed up the signal reconstruction process, and present a thorough experimental comparison of the performance of all the considered algorithms.

1. INTRODUCTION

Short-time Fourier transform (STFT) is one of the most fundamental and ubiquitous tools in audio signal processing, and an extremely large amount of techniques developed for a wide range of applications such as noise canceling, source separation, dereverberation or audio modification, are based on or involve the processing of the STFT spectrogram of a waveform. The main reasons for its success are that its computation is both fast and simple to implement, and that it efficiently unveils the time-frequency structure of a signal under the assumption that it is short-term stationary, which is often a good approximation for natural sounds, such as speech or music. Moreover, the STFT is both linear and invertible: an additive mixture of time-domain signals transforms to the corresponding additive mixture of STFT spectrograms, and there is a perfect equivalence between a time-domain signal and its complex STFT spectrogram, as the signal can be exactly reconstructed through a weighted overlap-add procedure [1, 2].

A very important issue is however often overlooked: the STFT spectrogram is a redundant representation, computed by concatenating Fourier transforms of overlapping short-time frames of the signal, and any processing performed in the STFT domain should thus take into account this redundancy. Concretely, starting from a "spectrogram-like" array of complex numbers in the complex time-frequency domain, it is not guaranteed whether there exists a signal in the time domain whose STFT is equal to that array of complex numbers. We shall call consistent spectrograms the arrays for which such a signal exists.

This article is a follow-up to our previous paper on phase estimation from a magnitude spectrogram [3], which formulated the problem of phase estimation as that of minimizing a numerical consistency criterion and derived a fast optimization algorithm. We give here further theoretical insights, present several potential variations on our previously introduced algorithm, make clear the relation between our framework and Griffin and Lim's iterative STFT algorithm [1], and perform a thorough experimental evaluation of the performance of all the considered algorithms in comparison with the iterative STFT algorithm.

After briefly reviewing the concept of spectrogram consistency and the derivation of the numerical consistency criterion, we present the proposed algorithms for fast phase estimation. Finally, we investigate the performance of the proposed methods.

2. CONSISTENT STFT SPECTROGRAMS

2.1. Constraints in the time-frequency domain

We consider in this paper arrays of complex numbers $(H_{m,n}) \in \mathbb{C}^{MN}$, where $m \in [\![0, M-1]\!]$ will correspond to the frame index and $n \in [\![0, N-1]\!]$ to the frequency band index. We assume that these arrays of complex numbers correspond to the (possibly modified) time-frequency representation of some time-domain signals, obtained through the short-time Fourier transform (STFT) with analysis window function w of length N and window shift R, such that $Q = N/R \in \mathbb{N}$. We assume that the inverse STFT, denoted by iSTFT, is performed through the overlap-add procedure using a synthesis window s equal to w, followed by normalization by an R-periodic function $u(t) = \sum_q w^2(t+qR)$ (t denoting the time index), and we note $\tilde{s} = s/u$ the normalized synthesis window. In this setting, a signal can be perfectly reconstructed from its STFT spectrogram through the inverse STFT, and the inverse STFT of a modified STFT spectrogram H leads to a reconstructed signal whose spectrogram is closest to H in a least-squares sense, as shown by Griffin and Lim [1].

In the following, we will call "consistent spectrograms" the elements of $\mathbb{C}^{MN}$ which can be obtained as the STFT spectrogram of a time-domain signal. As the STFT is a redundant representation, not all elements of $\mathbb{C}^{MN}$ are consistent. We characterize here consistency directly in the time-frequency domain. For an ar-

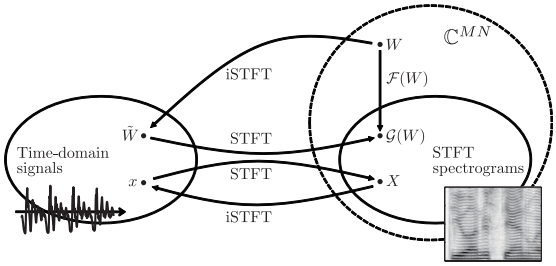


Figure 1: *Illustration of the concept of spectrogram consistency.*

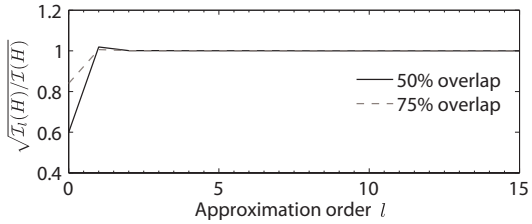


Figure 2: *Example of evolution of the approximated numerical consistency criterion $\mathcal{I}_l(H)$ w.r.t. the truncation order l.*

ray $H \in \mathbb{C}^{MN}$ to be a consistent spectrogram, it needs to be the STFT spectrogram of a signal which by perfect reconstruction is iSTFT(H). The set of consistent spectrograms can thus be described as the kernel (or null space) of the $\mathbb{R}$-linear operator from $\mathbb{C}^{MN}$ to itself defined by

$$\mathcal{F}(H) = \mathcal{G}(H) - H, \tag{1}$$

where $\mathcal{G}(H) = \text{STFT}(\text{iSTFT}(H))$. This result is illustrated in Fig. 1. We can obtain an explicit expression for $\mathcal{F}$ by writing down the computations in the STFT and inverse STFT as

$$\mathcal{F}(H)_{m,n} = \sum_{p=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{q=-(Q-1)}^{Q-1} e^{j2\pi \frac{q}{Q} n} \alpha_{q,p} H_{m-q,n-p}, \tag{2}$$

where frequency bin indices are considered modulo N and

$$\alpha_{q,p} = \frac{1}{N} \sum_{k} w(k)\tilde{s}(k+qR) e^{-j2\pi p \frac{k+qR}{N}} - \delta_p \delta_q, \tag{3}$$

where δ_i is the Kronecker delta.

2.2. Numerical consistency criterion

Instead of enforcing consistency through the "hard" constraints (1), as they may be too restrictive or hard to solve exactly, we relax them by taking the L^2 norm of $\mathcal{F}(H)$, defining a numerical criterion $\mathcal{I}(H) = ||\mathcal{F}(H)||^2$ as follows:

$$\mathcal{I}(H) = \sum_{m,n} \Big| \sum_{p=-\frac{N}{2}}^{\frac{N}{2}-1} \sum_{q=-(Q-1)}^{Q-1} e^{j2\pi \frac{q}{Q} n} \alpha_{q,p} H_{m-q,n-p} \Big|^2. \tag{4}$$

Looking at the actual values of the coefficients $\alpha_{q,p}$ involved in the definition of $\mathcal{F}(H)$, we notice that most of the weight is actually concentrated near $(0,0)$. One can thus approximate the consistency criterion by using only $(2l+1)\times(2Q-1)$ coefficients instead of the total $N \times (2Q-1)$, where $l \ll N/2$:

$$\mathcal{I}_l(H) = \sum_{m,n} \Big| \sum_{q=-(Q-1)}^{Q-1} e^{j2\pi \frac{q}{Q} n} \sum_{|p|\le l} \alpha_{q,p} H_{m-q,n-p} \Big|^2. \tag{5}$$

We investigate how good this approximation is on the example of the magnitude spectrogram of 20 s of speech by a male speaker. Denoting by S the complex spectrogram of the utterance, the array of complex (actually here all real) numbers $H = |S|$ constituted by the magnitude part only is, as the phase information is discarded, not consistent. We show in Fig. 2, for both 50 % and 75 % overlap cases, the evolution of the approximated numerical consistency criterion $\mathcal{I}_l$ w.r.t. the truncation order l, as the square root $\sqrt{\mathcal{I}_l(H)/\mathcal{I}(H)}$ of the ratio of $\mathcal{I}_l(H)$ to the exact criterion $\mathcal{I}(H)$. We used the sine window for the analysis and synthesis windows. In both cases, the numerical criterion $\mathcal{I}(H)$ is roughly equal to the total energy $||H||^2$. Most of the contribution is already taken into account for truncation orders larger than or equal to 2, in both 50 % and 75 % overlap cases. For $l = 2$, the approximation error is roughly equal to 0.12 % for both overlap cases, and it falls below 0.1 % for larger l.

These results show more generally that the values of $\mathcal{F}(H)_{m,n}$ or $\mathcal{G}(H)_{m,n}$ can be computed with very good accuracy as a sum over only a few terms of the form $e^{j2\pi \frac{q}{Q} n} \alpha_{q,p} H_{m-q,n-p}$ instead of convolutions over all the bins involving FFTs. This is the key point which will enable us to develop fast phase estimation algorithms in the following.

3. APPLICATION TO PHASE ESTIMATION

In the problem of phase estimation from a magnitude STFT spectrogram, we are given an array of real non-negative numbers $A_{m,n}$ which are supposedly the magnitude part of an STFT spectrogram, for example obtained through modifications of the magnitude spectrogram of a sound. The goal is to find the "most relevant" phase for A. We formulate here this problem as that of estimating a phase $\phi_{m,n}$ to adjoin to A such that $A_{m,n}e^{j\phi_{m,n}}$ is "as consistent as possible". Based on the above derivation, the problem amounts to minimizing the consistency criterion $\mathcal{I}$ w.r.t. the phase ϕ, with the magnitude A given, defining a new objective function $\tilde{\mathcal{I}}(\phi) = \sum_{m,n} \left|\mathcal{G}(Ae^{j\phi})_{m,n} - A_{m,n}e^{j\phi_{m,n}}\right|^2$ to minimize.

3.1. Griffin and Lim's iterative STFT

In [1], Griffin and Lim formulated the problem as that of estimating a real-valued time-domain signal x such that the magnitude of its STFT is closest to the magnitude spectrogram A in a least-squares sense. To solve it, they proposed the iterative STFT algorithm, which consists in iteratively updating the phase $\phi_{m,n}^{(k)}$ at step k by replacing it with the phase of the STFT of its inverse STFT, $\angle\mathcal{G}(Ae^{j\phi^{(k)}})$, while keeping A fixed.

We can actually interpret the iterative STFT as the application of the auxiliary function method [4] to the minimization of

$\tilde{\mathcal{I}}$. Indeed, $\mathcal{G}(H)$ can be shown [1] to be the closest consistent spectrogram to H in a least-squares sense, i.e., $\forall H$,

$$\sum_{m,n}\left|\mathcal{G}(H)_{m,n}-H_{m,n}\right|^2=\min_{\bar{H}\in\mathrm{Ker}(\mathcal{F})}\sum_{m,n}\left|\bar{H}_{m,n}-H_{m,n}\right|^2 \quad (6)$$

From the above relation, we can see that the function

$$\tilde{\mathcal{I}}^+(\phi,\bar{H})=\sum_{m,n}\left|\bar{H}_{m,n}-A_{m,n}e^{j\phi_{m,n}}\right|^2, \quad (7)$$

is an auxiliary function for $\tilde{\mathcal{I}}$ with auxiliary variable $\bar{H}\in\mathrm{Ker}(\mathcal{F})$. Alternate minimizations of $\tilde{\mathcal{I}}^+$ w.r.t. ϕ and $\bar{H}$ lead to an iterative optimization scheme for $\tilde{\mathcal{I}}(\phi)$ which is none other than Griffin and Lim's update,

$$\phi\leftarrow\angle\mathcal{G}(Ae^{j\phi}). \quad (8)$$

We can also show through this method that $\tilde{\mathcal{I}}(\phi)$ still decreases when updating a few bins only before recomputing $\mathcal{G}(Ae^{j\phi})$.

3.2. Fast minimization of $\tilde{\mathcal{I}}$

The iterative STFT algorithm is based on the computation of STFTs and inverse STFTs, and is thus quite costly. This way of computing $\mathcal{G}(Ae^{j\phi})$ is the bottleneck of the algorithm, and we would like to replace it by a light computation for each bin directly in the time-frequency domain. Using explicit summations of the form of Eq. (2) to replace the FFT-based updates would only raise the computational cost, but we saw in 2.2 that very good approximations involving only a few terms could be derived. We shall now explain in more details the approximations we consider and the many advantages that can be drawn from them.

3.2.1. Approximate optimizations

As mentioned in 2.2, thanks to the concentration of the coefficients $\alpha_{q,p}$ near $(0,0)$, we can approximate the computation of $\mathcal{F}(Ae^{j\phi})_{m,n}$ by using only $(2l+1)\times(2Q-1)$ central coefficients. This approximation enables us to compute directly the update of each bin through the summation of a few terms. Let $H=Ae^{j\phi}$. As $\mathcal{G}(H)=H+\mathcal{F}(H)$, the update (8) becomes

$$\phi_{m,n}\leftarrow\angle\Big(H_{m,n}+\sum_{\substack{p,q\\|p|\le l}}e^{j2\pi\frac{q}{Q}n}\alpha_{q,p}H_{m-q,n-p}\Big), \quad (9)$$

which we will refer to as the "Truncated Update", as it relies on the truncation of the sum involved in the original iterative STFT update. For $l=2$ and a $50\ \%$ overlap, for example, we only consider 5×3 coefficients.

We also investigate another kind of approximation. Noticing that $\alpha_{0,0}$ dominates over the other coefficients, we can attempt to neglect the influence of $\phi_{m,n}$ in all the terms $\mathcal{F}(H)_{m',n'}$ other than $\mathcal{F}(H)_{m,n}$, which is the one where it is multiplied by $\alpha_{0,0}$. We could thus consider, for the update of a bin $H_{m,n}$, looking at the minimization of $|\mathcal{F}(H)_{m,n}|$ only instead of the whole $|\mathcal{I}(H)|$. From Eq. (2), $|\mathcal{F}(H)_{m,n}|$ is minimized w.r.t. $H_{m,n}$ with the other terms fixed when $\alpha_{0,0}H_{m,n}$ is equal to the opposite of the sum of the terms coming from the neighboring bins. As we want here to keep $A_{m,n}$ fixed, the phase $\phi_{m,n}$ should be updated such that $\alpha_{0,0}A_{m,n}e^{j\phi_{m,n}}$ is in opposite direction with the terms coming from the neighboring bins. More precisely, as we can easily see that $\alpha_{0,0}=1/Q-1<0$, the update for bin (m,n) is

$$\phi_{m,n}\leftarrow\angle\Big(\sum_{\substack{(p,q)\neq(0,0)\\p\in[\![-\frac{N}{2},\frac{N}{2}-1]\!]}}e^{j2\pi\frac{q}{Q}n}\alpha_{q,p}H_{m-q,n-p}\Big). \quad (10)$$

Further combining this with the truncation introduced above, we obtain the following update:

$$\phi_{m,n}\leftarrow\angle\Big(\sum_{\substack{(p,q)\neq(0,0)\\|p|\le l}}e^{j2\pi\frac{q}{Q}n}\alpha_{q,p}H_{m-q,n-p}\Big). \quad (11)$$

We will refer to these updates as the "Modified Update" and the "Truncated Modified Update", respectively. Although the way in which these modified updates have been obtained may look rather exotic, they can actually be linked to the original iterative STFT update (8). Indeed, in concise form, the update (10) can also be written, using the operator $\mathcal{G}$,

$$\phi_{m,n}\leftarrow\angle\Big(\mathcal{G}(H)_{m,n}-\frac{1}{Q}H_{m,n}\Big), \quad (12)$$

which is very close to (8), up to the extra term $-\frac{1}{Q}H_{m,n}$. Unfortunately, $\tilde{\mathcal{I}}$ is not in general non-increasing under the modified update. However, we noticed experimentally that the truncated modified update leads to a very efficient minimization procedure for $\tilde{\mathcal{I}}$ when the newly updated values for each bin are used in the subsequent updates of the other bins.

3.2.2. Taking advantage of sparseness

The auxiliary function interpretation of the iterative STFT shows that we can update only a few bins, even only one, before recomputing $\mathcal{G}(Ae^{j\phi})$. Thanks to the low-cost updates derived above, we can efficiently exploit this property by selecting which bins to update. The idea here is to exploit the sparseness of the acoustic signal in the time-frequency domain to limit the updates to bins with a significant magnitude, as those with relatively small magnitude arguably contribute less to the objective function and are less likely to give disturbing perceptual artifacts. We progressively decrease the magnitude threshold above which the bins are updated, starting with the most significant bins and refining afterwards. Obviously, the speed at which this threshold is decreased at each iteration is likely to have a strong impact on performance: at a given iteration, if a very large threshold is used, only a few bins will be updated, leading to a very short computation time for this iteration on one hand, but also limiting the improvement which can be expected from that iteration; if on the contrary a very low threshold is used, most bins will be updated, leading to a significant improvement in the objective function but at the cost of a longer computation time. We experimentally evaluate in 4.3 several types of decreasing contours for the threshold as a function of the iteration number.

We note that further improvement could be obtained by sorting the bins by decreasing magnitude order beforehand and updating them from the largest to the smallest, stopping when the threshold is reached: we could then avoid comparison tests for each bin by precomputing where to stop for each step's threshold by going through the ordered magnitude list once only. We shall however leave such refinements for future developments.

3.2.3. Reducing computational cost using symmetries

We easily notice that $\alpha_{q,-p} = \overline{\alpha_{q,p}}$ and $\alpha_{-q,p} = \overline{\alpha_{q,p}}$. Based on these symmetries and on the fact that, for complex numbers a, b and c, the computation of the quantity $ab + \overline{a}c$ can be performed using only four real-number multiplications instead of the eight real-number multiplications required in general for the sum of two products of two complex numbers, we can reduce the number of multiplications involved in the computation of the truncated updates (9) and (11) by a factor 4.

3.2.4. Update schemes

As, in the truncated updates, the bins are updated one by one through independent low-cost computations, we can use at once the newly updated values for the computation of the subsequent updates of the other bins. We shall call this update scheme "on-the-fly" update, and the usual scheme where updates are computed using only values of the previous step as "stepwise" update. We expect the on-the-fly updates to lead to faster convergence, as the new values are re-used straight away, and incoherences between neighboring bins are thus likely to be smoothed more efficiently. Another benefit of the on-the-fly scheme is that updates are performed "in place", reducing the computation cost and the required memory space.

4. EXPERIMENTAL EVALUATION

We investigate the performances of the various proposed methods and compare them with that of Griffin and Lim's algorithm in terms of the decrease of a normalized inconsistency measure defined as $\mathcal{C}(H) = 10\log_{10}\frac{\mathcal{I}(H)}{\|H\|^2}$ (i.e., the objective function $\mathcal{I}(H)$ represented in decibels with the total energy of the original spectrogram as a reference) w.r.t. the number of iterations as well as w.r.t. the computation time. The task we consider is that of the estimation of the most consistent phase for a given magnitude spectrogram for the purpose of time-scale modification. All methods were implemented in C on a PC with 3.16 GHz CPU running Linux. We used 100 speech utterances (both male and female speakers, taken from Bagshaw's database [5], totaling 5 min and 32 s) and 8 excerpts of music pieces (piano and guitar solos from the RWC music database [6], totaling 3 min and 04 s) as the data. The sampling rate was 16 kHz. The sine window was used as analysis and synthesis window. The time-scaled spectrograms are built through a sliding-block analysis technique described in [3].

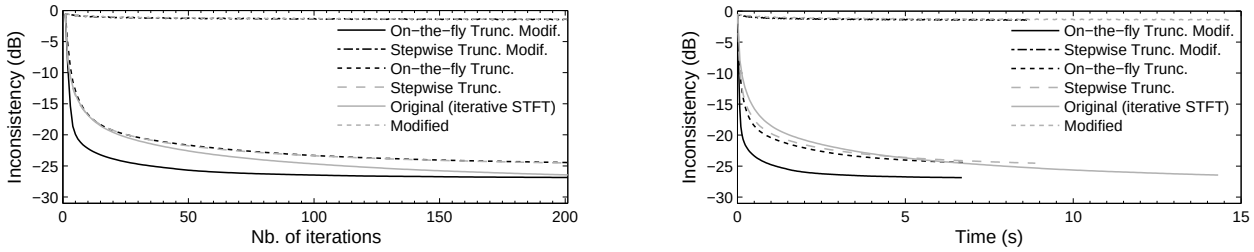
4.1. General performance comparison

In a first experiment, we compare the behaviors of the various methods under several time-scale modification and analysis conditions. The time-scale modification factors f used were 0.7 (slow down by 30 %), 1 (no time-scale modification), and 1.3 (pace up by 30 %). We used a frame length of 1024 samples for all the experiments, and two different settings for the frame shift, namely 512 (50 % overlap) and 256 (75 % overlap). We use the magnitude of the time-scaled spectrograms as the initial point for all the methods, i.e., we set the initial phase to zero. All methods are stopped after 200 iterations. Only the time required by the phase estimation part of the algorithm is reported as other parts are identical. The sparseness threshold at iteration k is set to $ae^{-bk^c}E$, where E denotes the mean magnitude of the given magnitude spectrogram and is introduced for the purpose of normalization. The parameters were set to $a = 100$, $b = 0.1$ and $c = 1$ based on a preliminary experiment on one music piece.

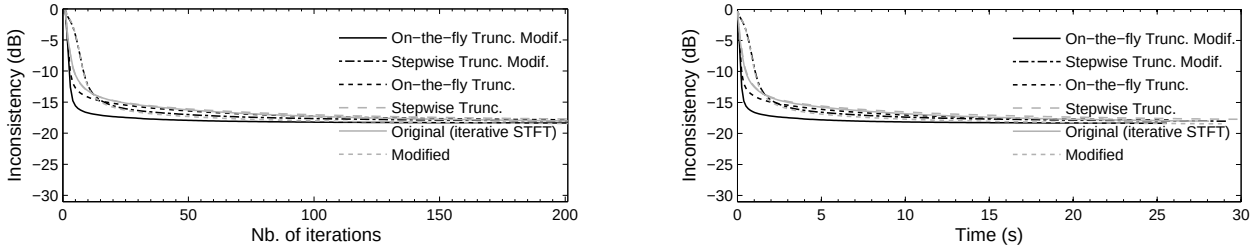
Figure 3 shows the evolution of the normalized inconsistency measure $\mathcal{C}$ w.r.t. the number of iteration (left) and the computation time (right) on the speech data. We can see first in general that the On-the-fly Truncated Modified update (11) (solid black line) constantly outperforms all other methods, and in particular the iterative STFT (8) (solid gray line), in terms of decrease of inconsistency w.r.t. the computation time, and w.r.t. the number of iterations as well in the no sparseness threshold case. However, the Modified update (10) (dotted gray line) and Stepwise Truncated Modified update (dash-dot black line) do not lead to a significant decrease in the 50 % overlap case: it seems that the influence of the extra term $-\frac{1}{Q}H$ is harmful when not used in on-the-fly updates. In the 75 % overlap case, the $\frac{1}{Q}$ factor decreases and these two methods lead to very good results as well. The On-the-fly and Stepwise Truncated versions (9) of the iterative STFT algorithm (dotted black line and dashed gray line) perform similarly to the iterative STFT, but seem to lead to better performance in terms of computational time when a sparseness threshold is introduced, as can be seen in Figs. 3 (d) and (e). Results on the music data showed similar trends.

Several other remarks can be made on these results. We can see by comparing Figs. 3 (a) and (b) that the typical values of $\mathcal{C}$ differ between 50 % and 75 % overlap, the decrease reached for 75 % overlap being smaller than for 50 % overlap, and they should not be directly compared numerically. It turns out that, from our experience based on informal listening experiments, a -17 dB measure with 75 % overlap for example actually tends to sound better than -22 dB with 50 % overlap. From Figs. 3 (b) and (c), obtained with different f, we can see the influence of the modification factor on the values reached: lower $\mathcal{C}$ values are usually obtained for the magnitude spectrogram without time-scale modification ($f = 1$) than for those with modification ($f = 0.7$ and $f = 1.3$). This can be connected to the fact that for $f = 1$ there actually exists a solution to the phase estimation problem, for which $\mathcal{C}$ is equal to $-\infty$, while there may be no exact solution in general for modified magnitude spectrograms. This tendency is especially true for 75 % overlap, where the constraints are stronger. The influence of the introduction of a sparseness threshold can be seen by comparing Figs. 3 (a) and (d), in which the parameters differ only by the value of a. As only a small part of the time-frequency bins are updated in the first few iterations, the decrease of $\mathcal{C}$ w.r.t. the number of iterations is much slower when a sparseness threshold is used. However, each iteration being much faster, overall the performance is much better w.r.t. the computation time than when all bins are updated at each iteration. Looking now at Fig. 3 (e), we see that raising the truncation order to $l = 5$ seems to lead to a slightly better final value, generally for a similar computation time or only slightly longer than for $l = 2$. Each iteration is slightly slower, but larger improvements compensate for this extra computational time per iteration. This is especially true when combined with the use of a sparseness threshold for bin selection.

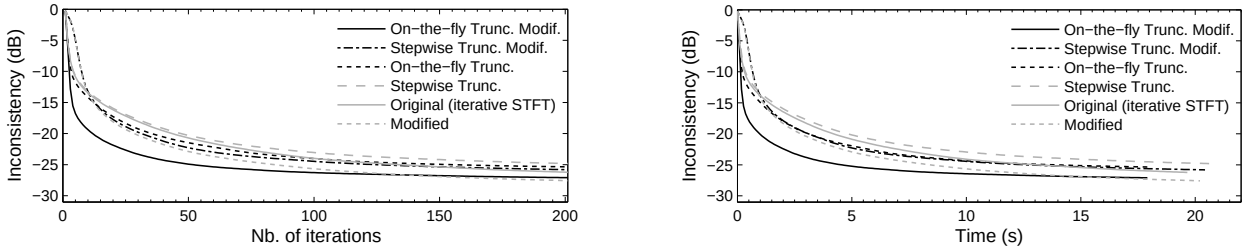
We show in Table 1 and Table 2 the computation time required by the iterative STFT and the proposed On-the-fly Truncated Mod-



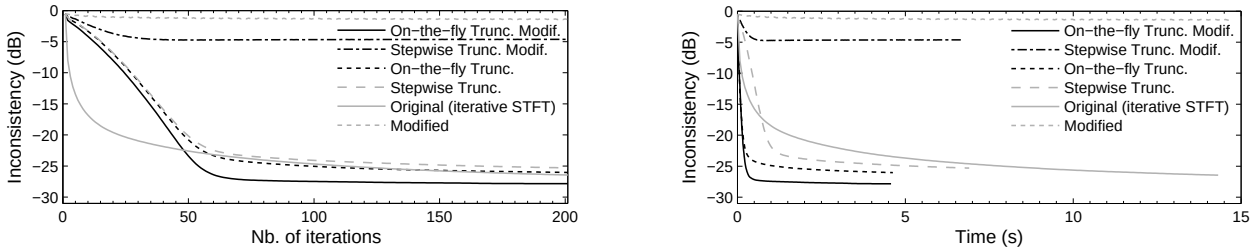
(a) 50 % overlap, $f = 0.7$, $l = 2$, $a = 0$.



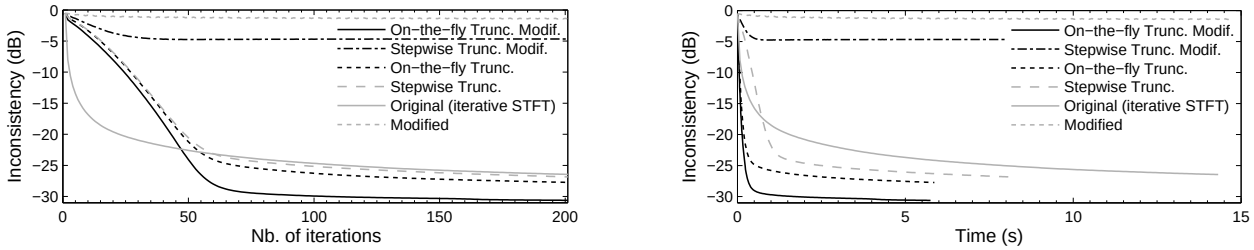
(b) 75 % overlap, $f = 0.7$, $l = 2$, $a = 0$.



(c) 75 % overlap, $f = 1$, $l = 2$, $a = 0$.



(d) 50 % overlap, $f = 0.7$, $l = 2$, $a = 100$.



(e) 50 % overlap, $f = 0.7$, $l = 5$, $a = 100$.

Figure 3: *Evolution of the inconsistency measure $\mathcal{C}$ w.r.t. the number of iterations (left) and the computation time (right) on the speech data.*

ified update (11) to reach a given level of decrease in $\mathcal{C}$, for 50 % overlap and 75 % overlap respectively. The results are averaged on the speech and music data, for $f = 0.7$, and the average output signal length is 41 s. We notice again the very important reduction in computation time brought first by using truncated computations for $\mathcal{G}(H)$ and second by the bin selection scheme: for example, compared with the iterative STFT, the average computation time required to reach -21 dB for 50 % overlap is divided by a factor 3 in the proposed method with $a = 0$ and $l = 5$, and by a factor 60 with $a = 100$.

Table 1: *Computation time (s) required to reach a given decrease of $\mathcal{C}$ (50 % overlap, $f = 0.7$).*

Time to reach	-15 dB	-18 dB	-21 dB
Iterative STFT	0.81 s	2.36 s	10.01 s
Prop., $a=0$, $l=2$	0.09 s	0.43 s	3.52 s
Prop., $a=0$, $l=5$	0.11 s	0.37 s	3.32 s
Prop., $a=100$, $l=2$	0.06 s	0.09 s	0.18 s
Prop., $a=100$, $l=5$	0.06 s	0.08 s	0.16 s

Table 2: *Computation time (s) required to reach a given decrease of $\mathcal{C}$ (75 % overlap, $f = 0.7$).*

Time to reach	-16 dB	-17 dB	-17.5 dB
Iterative STFT	7.75 s	14.41 s	20.84 s
Prop., $a=0$, $l=2$	1.22 s	3.42 s	6.63 s
Prop., $a=0$, $l=5$	1.56 s	4.05 s	7.16 s
Prop., $a=100$, $l=2$	0.49 s	1.76 s	9.84 s
Prop., $a=100$, $l=5$	0.52 s	1.10 s	4.84 s

4.2. Divergence from equilibrium

Consistent spectrograms are invariant by $\mathcal{G}$, and are thus stability points for the iterative STFT algorithm. We investigate here the influence of the truncations on this stability property. The methods we proposed are expected not to be stable anymore and to diverge from the initial consistent spectrogram, but hopefully they should not diverge too much. We performed experiments with both truncated methods (modified or not). We tested both stepwise and on-the-fly updates for the unmodified method, while we dropped the stepwise update for the modified method as we saw in the previous experiment that it did not lead to an effective algorithm. Note that the consistent spectrograms are invariant through the iterative STFT (8) and the modified update (10), which are thus not relevant for the current experiment. We used the unmodified complex STFT spectrogram of each piece of data as initial point, the frame overlap being set to 50 %. The evolution of the measure of inconsistency $\mathcal{C}$ averaged on the music data for the three methods considered is shown in Fig. 4. Similar results were obtained on the speech data. We can see that all the methods slightly diverge from the initial point, although the measure $\mathcal{C}$ stays lower than -30 dB. This is typically lower than the results we obtained in the previous section starting from a magnitude spectrogram with any of the methods, truncated or not, and the divergence can be considered moderate. This result also gives an estimate of the maximum decrease which can be expected from the truncated methods.

4.3. Sparseness threshold experiments

We investigate here the influence of the way the sparseness threshold decreases on the performance of the on-the-fly truncated modified update (11). As a general setting, we consider a time-scale modification factor of 0.7 and 50 % overlap, and the truncation order is set to $l = 2$. We investigate several settings for the decrease of the sparseness threshold introduced in 4.1. At iteration k, only bins whose magnitude is larger than $ae^{-bk^c}E$ are updated. Depending on the values of a, b and c, at each step a certain subset of the time-frequency bins only is updated, and this subset grows at each iteration. The values we investigated were $a \in \{1, 10, 50, 100\}$, $b \in \{0.0001, 0.0005, 0.001, 0.005, 0.01, 0.05, 0.1, 0.5\}$ and $c \in \{1, 2\}$, corresponding to linear and quadratic decreases with various slopes and starting points in the decibel domain.

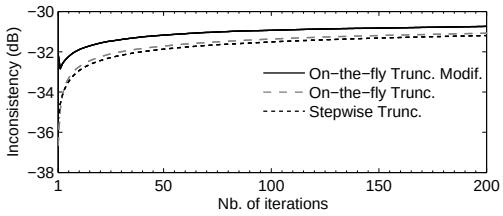
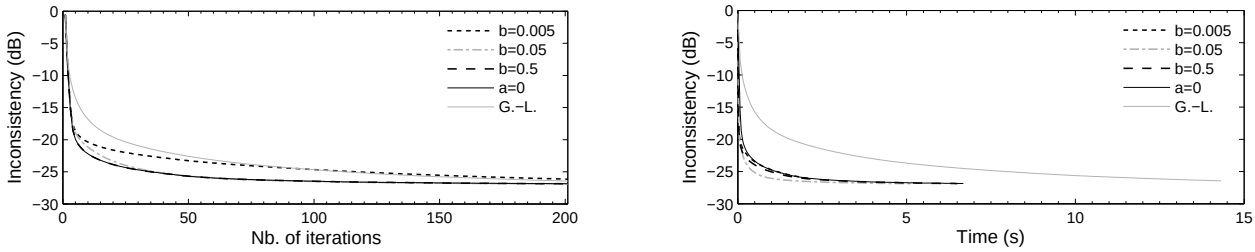


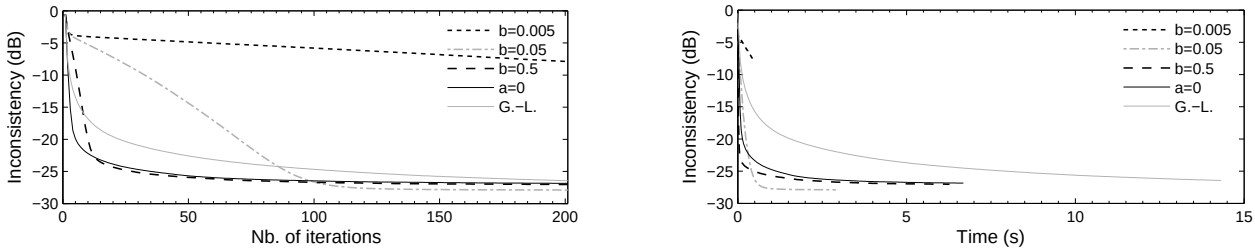
Figure 4: *Divergence of the truncated methods starting from a consistent spectrogram.*

We show in Fig. 5 a few examples of results for various sparseness threshold decrease settings on the speech data (results on music data showed similar behaviors), as the evolution of $\mathcal{C}$ w.r.t. the number of iterations (left) and the computation time (right). We also show for comparison on each plot the results obtained by the proposed truncated modified update without sparseness threshold ($a = 0$) as a dark continuous line, and by the iterative STFT (G.-L.) as a gray continuous line. The top and middle plots both correspond to $c = 1$, and differ by the initial value aE used, with $a = 1$ for the top plots and $a = 50$ for the middle ones. The results for $a = 10$ and $a = 100$ are omitted as they are similar to those for $a = 50$. In the bottom plots, the decrease in the log-domain is quadratic ($c = 2$), and the initial value is set to $aE = 10E$ (10 times the average magnitude). Results for $a \in \{1, 50, 100\}$ are omitted as they show only limited variation to those for $a = 10$. Note that the meaning of the plot styles varies with the plots: we omitted for clarity the results for $b \in \{0.0001, 0.0005, 0.001\}$ in the plots for $c = 1$ as the threshold decreases too slowly to lead to a significant decrease in 200 iterations, and the results for $b \in \{0.01, 0.1\}$ as they showed intermediate behaviors to the plots displayed. Similarly, we omitted the results for $b \in \{0.05, 0.1, 0.5\}$ in the plots for $c = 2$ as the threshold decreases so fast that the results are very close to the no threshold ($a = 0$) curve, and the results for $b \in \{0.0005, 0.005\}$ as they showed intermediate behaviors to the plots displayed.

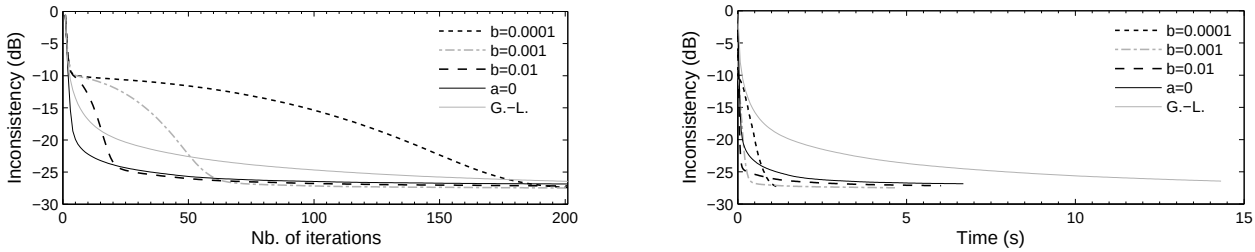
The experiments performed here show that a very good trade-off between updating enough bins so as to make the inconsistency measure decrease and skipping enough bins to make the computation faster can be obtained, leading to very important reductions in the computation time required to reach a given decrease of the inconsistency measure $\mathcal{C}$. Even if the number of iterations necessary to reach a given decrease of the inconsistency measure may be larger with sparseness thresholds than for the iterative STFT or the proposed method without sparseness threshold, as can for example be seen in the left part of Fig. 5 (b) for $b = 0.005$ or Fig. 5 (c) for $b = 0.0001$, the shorter time taken by each iteration when only updating a fraction of the time-frequency bins leads to a faster decrease in terms of computation time, as can be seen in the right part of Figs. 5 (b) and (c). Furthermore, we can see that several settings of a, b and c lead to similar computation times. The performance of the algorithm is thus expected to be robust to the choice of threshold parameters within a wide range of values.



(a) $a = 1, c = 1$



(b) $a = 50, c = 1$



(c) $a = 10, c = 2$

Figure 5: *Evolution of the inconsistency measure $\mathcal{C}$ w.r.t. the number of iterations (left) and the computation time (right) on the speech data for various sparseness threshold decrease settings.*

5. CONCLUSION

We introduced a general framework to assess for the consistency of complex STFT spectrograms, studied the relationship between our framework and Griffin and Lim's iterative STFT algorithm, and developed a fast phase estimation algorithm which can take advantage of the sparseness of the input signal in the time-frequency domain. We performed a thorough experimental evaluation showing that, to reach the same level of quality in the estimation of the phase from a given magnitude spectrogram, our method required much less computation time. Future work will investigate the introduction of consistency concepts in various signal processing algorithms and the use of techniques similar to those introduced here to reduce computational cost.

6. REFERENCES

[1] D. W. Griffin and J. S. Lim, "Signal estimation from modified short-time Fourier transform," *IEEE Trans. ASSP*, vol. 32, no. 2, pp. 236–243, Apr. 1984.

[2] B. Yang, "A study of inverse short-time Fourier transform," in *Proc. ICASSP*, Apr. 2008, pp. 3541–3544.

[3] J. Le Roux, N. Ono, and S. Sagayama, "Explicit consistency constraints for STFT spectrograms and their application to phase reconstruction," in *Proc. SAPA*, Sep. 2008.

[4] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. NIPS*2000*, 2001, pp. 556–562.

[5] P. C. Bagshaw, S. M. Hiller, and M. A. Jack, "Enhanced pitch tracking and the processing of F_0 contours for computer and intonation teaching," in *Proc. Eurospeech*, Sep. 1993, pp. 1003–1006.

[6] M. Goto, "Development of the RWC Music Database," in *Proc. ICA 2004*, vol. I, 2004, pp. 553–556.