Explicit consistency constraints for STFT spectrograms and their application to phase reconstruction

Jonathan Le Roux, Nobutaka Ono and Shigeki Sagayama

Graduate School of Information Science and Technology, The University of Tokyo, Japan
{leroux,onono,sagayama}@hil.t.u-tokyo.ac.jp

Abstract

As many acoustic signal processing methods, for example for source separation or noise canceling, operate in the magnitude spectrogram domain, the problem of reconstructing a perceptually good sounding signal from a modified magnitude spectrogram, and more generally to understand what makes a spectrogram consistent, is very important. In this article, we derive the constraints which a set of complex numbers must verify to be a consistent STFT spectrogram, i.e. to be the STFT spectrogram of a real signal, and describe how they lead to an objective function measuring the consistency of a set of complex numbers as a spectrogram. We then present a flexible phase reconstruction algorithm based on a local approximation of the consistency constraints, explain its relation with phase-coherence conditions devised as necessary for a good perceptual sound quality, and derive a real-time time scale modification algorithm based on sliding-block analysis. Finally, we show how inconsistency can be used to develop a spectrogram-based audio encryption scheme.

Index Terms: Short-time Fourier transform, Phase reconstruction, Spectrogram modification, Phase coherence

1. Introduction

Many acoustical signal processing techniques, developed for a wide range of applications such as source separation[1, 2, 3, 4], noise canceling [5], time-scale and pitch-scale modifications or more generally audio modification [6], involve a processing of the magnitude spectrogram, whether it be a short-time Fourier transform (STFT) spectrogram, or a spectrogram obtained using other transforms such as constant-Q transforms for example. To be able to produce a perceptually satisfactory sounding signal, phase information is necessary but usually not available. In many situations, such as frequency-domain-based methods for time-scale modification or for reconstruction of missing parts of an acoustic signal [6, 7], phase must be partially or totally reconstructed. Sometimes, as in source separation, based on a sparseness assumption on the repartition of acoustic energy in the time-frequency space, the phase of a mixture can be used as a rough estimation of the phase when reconstructing each extracted component using the estimated magnitude spectrograms. However, in both cases, incoherences between the phase and the magnitude spectrogram from which we want to reconstruct a signal lead in general to perceptually disturbing artifacts. Moreover, the magnitude spectrogram of the reconstructed signal may actually be very different from the one we intended to reconstruct a signal from. An effective method for phase reconstruction would thus have many applications and broaden the range of situations where magnitude spectrogram based techniques can be applied.

In order to be able to reconstruct the phase of a spectrogram such that it is as coherent as possible with a given magnitude, one first needs to understand what “coherent” means in such a context. It is thus important to have an explicit criterion judging the coherence of the phase, i.e., to quantify the consistency of a set of complex numbers. We will focus here on the STFT spectrogram. In the following, we will call consistent STFT spectrogram a set of complex numbers which has been obtained as the STFT spectrogram of a real signal, and inconsistent STFT spectrogram a set of complex numbers which cannot be obtained as such. In this paper, we derive explicit consistency constraints for STFT spectrograms as a simple linear system with coefficients depending on the window length, the frame shift and the analysis and synthesis windows used to build the spectrogram or which the spectrogram is assumed to have been obtained from.

The iterative STFT algorithm [8] introduced by Griffin and Lim is the reference for phase reconstruction based on a modified magnitude STFT spectrogram. Its principle is to find the closest consistent STFT spectrogram to a given modified magnitude spectrogram. Here, we propose a flexible phase reconstruction algorithm based on the derived consistency constraints. It is conceptually close to the iterative STFT algorithm, but the computational cost is reduced by focusing on local phase coherence conditions and by enabling at each iteration the update of each time-frequency bin’s phase independently. This freedom in the selection of which bin to update gives an extra flexibility to our algorithm, and we believe that it could thus be embedded easily in a wide range of other signal processing techniques.

We will first review in Section 2 the perfect reconstruction conditions on the analysis and synthesis windows, then derive in Section 3 the consistency constraints for STFT spectrograms. In Section 4, we will introduce the algorithm for phase reconstruction, based on the optimization of an objective function defined using the consistency constraints, and show how it can be used to develop a real-time time-scale modification algorithm. Finally, in Section 5, we will explain how inconsistent spectrograms can be used to perform audio encryption, the synthesis window acting as a decoding key.

2. Necessary condition on the window functions for perfect reconstruction

Let $(X(t))_{t\in\mathbb{Z}}$ be a digital signal. We review here the conditions for perfect reconstruction of the signal through STFT and inverse STFT [8]. Let N be the window length, R the window shift, W the analysis window function and S the synthesis window function. We suppose that W and S are zero outside the interval $0 \le t \le N-1$. We assume that the window length N is an integer multiple of the shift R, and we note $Q = N/R$. The

STFT at frame m is defined as the discrete Fourier transform (DFT) of the windowed short-time signal $W(t-mR)X(t)$ (with the phase origin at the start of the frame, $t=mR$).

The inverse STFT procedure consists in Fourier-inverting each frame of the STFT spectrogram, multiplying each obtained (periodic) short-time signal by a synthesis window and summing together all the windowed short-time signals. On a particular frame $mR \leq t \leq mR+N-1$, this leads to a reconstructed signal $Y(t)$ given by

$$\begin{aligned} Y(t) =& S(t-mR)W(t-mR)X(t) \\ &+\sum_{q=1}^{Q-1} S(t-(m-q)R)W(t-(m-q)R)X(t) \\ &+\sum_{q=1}^{Q-1} S(t-(m+q)R)W(t-(m+q)R)X(t) \end{aligned}$$

where the three terms on the right-hand side are respectively the contribution of the inverse transforms of frame m, the overlapping frames on the left and the overlapping frames on the right. As the contributions of frames with an index difference larger than Q do not overlap, by equating $Y(t)=X(t)$ for all t, we obtain as in [8] the following necessary condition for perfect reconstruction

$$1=\sum_{q=0}^{Q-1} W(t-qR)S(t-qR). \qquad (1)$$

3. Derivation of the consistency constraints for STFT spectrograms

Let $(H(m,n))_{0\leq m\leq M-1, 0\leq n\leq N-1}$ be a set of complex numbers, where m will correspond to the frame index and n to the frequency band index, and W and S be analysis and synthesis windows verifying the perfect reconstruction conditions (1) for a frame shift R. For the set H to be a consistent STFT spectrogram, it needs to be the STFT spectrogram of a signal $X(t)$. But by consistency, this signal can be none other than the result of the inverse STFT of the set $(H(m,n))$. A necessary and sufficient condition for H to be a consistent spectrogram is thus for it to be equal to the STFT of its inverse STFT. The point here is that, for a given window length N and a given frame shift, if we denote the inverse STFT by iSTFT, the operation iSTFT ∘ STFT from the space of real signals to itself is the identity, while STFT ∘ iSTFT from $\mathbb{C}^{M\times N}$ to itself is not.

Let us derive consistency constraints for STFT spectrograms based on this consideration, by explicitly stating that a spectrogram must be equal to the STFT of its inverse STFT. If we focus on a single frame, this leads to the following computation. For convenience of notation, we introduce the shifted index $k=t-mR$. Let us first work out the contribution of frame m. Its inverse DFT is given by

$$h_m(k)=\frac{1}{N}\sum_{n=0}^{N-1} H(m,n)e^{j2\pi n\frac{k}{N}} \qquad (2)$$

which is first windowed by the synthesis window $S(k)$ to recover a short-time signal $l_m(k)=S(k)h_m(k)$ that will later be overlap-added to its neighbors to obtain the inverse STFT signal $X(t)$.

Similarly, for frame $m+q$ we obtain

$$l_{m+q}(k)=\frac{1}{N}S(k-qR)\sum_{n=0}^{N-1} H(m+q,n)e^{j2\pi n\frac{k-qR}{N}}. \qquad (3)$$

The short-time signals $l_{m+q}(k)$ are added, leading to the inverse STFT of H for $mR \leq t \leq mR+N-1$. This signal is then windowed by the analysis window $W(k)$, and the DFT is computed to obtain the STFT. By equating the result to the original set $H(m,n)$, we obtain a set of equations which are the conditions we are looking for. For $0\leq n' \leq N-1$,

$$\begin{aligned} H(m,n') =& \frac{1}{N}\sum_{k} W(k)e^{-j2\pi k\frac{n'}{N}} \\ &\Big\{S(k)\sum_{n=0}^{N-1} H(m,n)e^{j2\pi n\frac{k}{N}} \\ &+\sum_{q=1}^{Q-1} S(k+qR)\sum_{n=0}^{N-1} H(m-q,n)e^{j2\pi n\frac{k+qR}{N}} \\ &+\sum_{q=1}^{Q-1} S(k-qR)\sum_{n=0}^{N-1} H(m+q,n)e^{j2\pi n\frac{k-qR}{N}}\Big\}. \end{aligned} \qquad (4)$$

By introducing the coefficients

$$\alpha_q^{(R)}(p)=\frac{1}{N}\sum_{k} W(k)S(k+qR)e^{-j2\pi p\frac{k+qR}{N}}-\delta_p\delta_q, \qquad (5)$$

where $-(N-1)\leq p\leq N-1$ and δ_i is the Kronecker delta ($\delta_0=1$ and $\delta_i=0$ for $i\neq 0$), we can rewrite this set of equations as a linear system and obtain the consistency constraints we are looking for.

Theorem. *For an analysis window W and a synthesis window S verifying the perfect reconstruction conditions (1) for a frame shift R, a set of complex numbers $H\in\mathbb{C}^{M\times N}$ is a consistent spectrogram if and only if, $\forall m\in[\![0,M-1]\!], \forall n'\in[\![0,N-1]\!]$,*

$$\begin{aligned} 0=\sum_{n=0}^{N-1}\Big[&\alpha_0^{(R)}(n'-n)H(m,n) \\ &+\sum_{q=1}^{Q-1} e^{j2\pi\frac{qR}{N}n'}\alpha_q^{(R)}(n'-n)H(m-q,n) \\ &+\sum_{q=1}^{Q-1} e^{-j2\pi\frac{qR}{N}n'}\alpha_{-q}^{(R)}(n'-n)H(m+q,n)\Big], \end{aligned} \qquad (6)$$

or more concisely:

$$\sum_{q=-(Q-1)}^{Q-1} e^{j2\pi\frac{qR}{N}n'}\big(\alpha_q^{(R)}*H\big)(m-q,n')=0, \qquad (7)$$

where the convolution acts on the second parameter of H and the coefficients $\alpha_q^{(R)}$ are defined by (5).

The above theorem summarizes in simple mathematic terms the fact that a consistent STFT spectrogram must be equal to the STFT of its inverse STFT.

4. Phase reconstruction for a modified STFT spectrogram

4.1. Objective function for phase reconstruction problems

Equation (7) represents the relation between a set of complex numbers and the STFT of its inverse STFT. The L^2 norm of its left member, i.e. the difference between H and the STFT of its inverse STFT,

$$\mathcal{I}(H) = \sum_{m,n} \Big| \sum_{q=-(Q-1)}^{Q-1} e^{j2\pi \frac{qR}{N} n} \big(\alpha_q^{(R)} * H\big)(m-q,n)\Big|^2, \quad (8)$$

is equal to zero for a consistent STFT spectrogram as stated in (7), and can be considered as a criterion on the consistency of a set of complex numbers considered as an STFT spectrogram.

In the problem of phase reconstruction, we are given a set of real non-negative numbers $A_{m,n}$ which are supposedly the amplitude part of an STFT spectrogram, for example obtained through modifications of the power spectrum of a sound. The goal is to estimate the phase $P_{m,n}$ to adjoin to A such that $A_{m,n}e^{jP_{m,n}}$ is as close as possible to be a consistent STFT spectrogram.

Based on the above derivation, this amounts to minimizing the objective function $\mathcal{I}$ w.r.t. the phase P, with the amplitude A given:

$$\tilde{\mathcal{I}}(P) = \sum_{m,n} \Big| \sum_{q=-(Q-1)}^{Q-1} e^{j2\pi \frac{qR}{N} n} \alpha_q^{(R)} * A_{m-q,n} e^{jP_{m-q,n}} \Big|^2, \quad (9)$$

If an estimation of the phase, for example the phase of the mixture when dealing with source separation, is available, it can be used as initial setting for P.

In [8], Griffin and Lim presented the iterative STFT algorithm, which consists in iteratively updating the phase $P_{m,n}^{(k)}$ at step k by replacing it with the phase of the STFT of its inverse STFT while keeping the magnitude A. The algorithm is illustrated in Fig. 1, where $x^{(k+1)}$ denotes the inverse STFT of $A_{m,n}e^{jP_{m,n}^{(k)}}$, $\hat{x}^{(k+1)}$ the STFT of $x^{(k+1)}$, and $P_{m,n}^{(k+1)}$ the phase of $\hat{x}^{(k+1)}$, $P_{m,n}^{(k+1)} = \angle \hat{x}^{(k+1)}$.

They showed that this procedure estimates a real signal x which minimizes (at least locally) the distance

$$d(x,A) = \sum_{m,n} \Big| |\hat{x}|_{m,n} - A_{m,n} \Big|^2, \quad (10)$$

i.e., the squared error between the magnitude of the STFT $\hat{x}$ of x, and the magnitude spectrogram A. As can be seen in Fig. 1, we shall note that the objective function $\tilde{\mathcal{I}}$ measures a slightly different quantity from the distance (10), but that the iterative STFT algorithm also converges to a minimum of (9). Indeed, both distances become equivalent near the convergence, as one can show that $d(x^{(k+1)}, A) \leq \tilde{\mathcal{I}}(P^{(k)}) \leq d(x^{(k)}, A)$ [8]. However, the objective function $\tilde{\mathcal{I}}$ we introduced has the advantages to be explicit, and in its general version (8) not to be limited to phase reconstruction problems with fixed magnitude. We believe that, thanks to its explicitness, it provides a flexible framework to be used inside other signal processing algorithms dealing with spectrograms. Here, it enables us to derive a simplified algorithm for phase reconstruction, as we shall now explain.

4.2. Direct optimization of $\tilde{\mathcal{I}}$

The iterative STFT algorithm, as mentioned above, can be used to minimize $\tilde{\mathcal{I}}$. However, this can be considered as an indirect minimization, and it is worth looking at a direct minimization of $\tilde{\mathcal{I}}$ through classical optimization methods. This will indeed provide us with the freedom to modify/approximate the objective function on one hand, and to select how each bin will be dealt with on the other. For example, if only some parts of the spectrogram must have their phase reconstructed, iterative STFT does not allow to keep the other parts unchanged and reconstruct the phase only where it is necessary while taking into account boundary conditions between the regions. This can be simply performed with the framework we develop here by updating only the bins whose phase is considered not reliable.

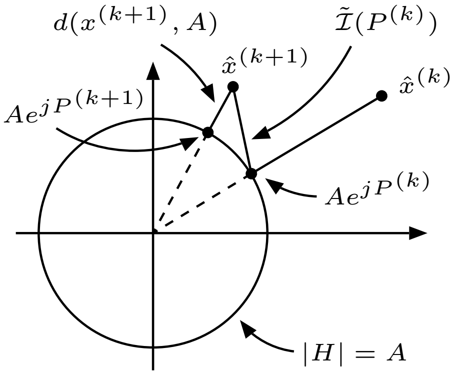

Figure 1: Illustration of the iterative STFT algorithm and the relation between the objective function $\tilde{\mathcal{I}}$ and the distance $d(x, A)$.

4.3. Approximate objective function and phase coherence

Here, we will make the following two approximations. We will first neglect the influence of $P_{m,n}$ in all the terms other than the one where it corresponds to DC (i.e., where it is multiplied by $\alpha_0^{(R)}(0)$). The motivation behind this first approximation is that the coefficient $\alpha_0^{(R)}(0)$ dominates over the other coefficients. By assuming the other phase terms fixed, we will then update each bin's phase $P_{m,n}$ so that $\alpha_0^{(R)}(0)A_{m,n}e^{jP_{m,n}}$ is in opposite direction with the terms coming from the neighboring bins, while keeping its amplitude $A_{m,n}$ fixed. This corresponds to performing a coordinate descent method [9]. More precisely, the update for bin (m, n') is

$$P_{m,n'} \leftarrow -s\angle\Big(\sum_{(n,q)\neq(n',0)} \alpha_q^{(R)}(n'-n)H(m-q,n)\Big), \quad (11)$$

where $s = 1$ if $\alpha_0^{(R)}(0) > 0$ and $s = -1$ if $\alpha_0^{(R)}(0) < 0$.

Furthermore, we notice that most of the weight in the coefficients $\alpha_q^{(R)}(p)$ is actually concentrated near $(0, 0)$ (with p considered modulo N), as can be seen in Fig. 2 for a window length $N = 512$ and a frame shift $R = 256$, with a Hanning analysis window and a rectangular synthesis window. One can thus approximate the consistency conditions by using only $l \times (2Q-1)$ coefficients instead of the total $N \times (2Q-1)$, where $l \ll N$. This approximation is motivated as well by the importance of local phase coherences, in particular the so-called "horizontal" and "vertical" coherences, to obtain a perceptually good reconstructed signal, and can be considered close to phase locking techniques [6, 10, 11]. Horizontal coherence refers to phase consistency within each frequency channel, i.e., to the fact that in frequency band n, phase roughly evolves at a speed corresponding to n, and vertical coherence refers to phase consistency across channels, in particular to the fact that in a time frame m, the phases at bins n and $n+1$ are roughly equal.

This approximation enables us to compute directly the update of each bin through the summation of a few terms, instead of the whole convolution which would be involved if using all the terms. The update becomes:

$$P_{m,n'} \leftarrow -s\angle\Big(\sum_{\substack{(n,q)\neq(n',0)\\ |n|\leq l}} \alpha_q^{(R)}(n'-n) H(m-q,n) \Big), \quad (12)$$

where frequency indices are understood modulo N. For $l = 2$ and a 50% overlap, for example, we only consider 5×3 coefficients.

4.4. Taking advantage of sparseness

As evoked above, using a direct optimization of the objective function $\tilde{\mathcal{I}}$ enables us to select which bins to update. This can be the key to deal with problems where only a part of the spectrogram has to have its phase reconstructed, but it can also in general be used to lower the computational cost. Indeed, we can use the sparseness of the acoustic signal to limit the updates to bins with a significant amplitude, or progressively decrease the amplitude threshold above which the bins are updated, starting with the most significant bins and refining afterwards. This idea can be related to the peak picking techniques in [6, 10].

4.5. Further simplifications

The number of operations involved in the computation of the updates (12) can be further reduced by noticing symmetries in the coefficients $\alpha_q^{(R)}$. First, without any assumption on the analysis and synthesis windows, it is obvious from (5) that

$$\alpha_q^{(R)}(-p) = \overline{\alpha_q^{(R)}(p)}. \quad (13)$$

When the analysis and synthesis windows are symmetric and such that $W(0) = 0$, the coefficients have still more symmetries. Indeed, we notice that, from (5),

$$\begin{aligned}
\alpha_q^{(R)}(p) &= \frac{1}{N}\sum_{k=0}^{N-1} W(k)S(k+qR)e^{-j2\pi p\frac{k+qR}{N}}\\
&= \frac{1}{N}\sum_{k=0}^{N-1} W(N-k)S(N-(k+qR))e^{-j2\pi p\frac{k+qR}{N}}\\
&= \frac{1}{N}\sum_{k'=1}^{N} W(k')S(k'-qR)e^{j2\pi p\frac{k'-qR}{N}}\\
&= \overline{\alpha_{-q}^{(R)}(p)}, \qquad (14)
\end{aligned}$$

as the difference between the last two lines is $\frac{1}{N}(W(N)S(N-qR)-W(0)S(-qR))e^{-j2\pi p\frac{qR}{N}}$, which is zero under the above assumptions.

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 4 real multiplications instead of the 8 real multiplications required for the general sum of two products of two complex numbers, we can reduce the number of multiplications involved in the computation.

4.6. Optimization of the analysis/synthesis windows

As the updates (12) are approximate versions of the updates (11) based on the observation that the weight in the coefficients $\alpha_q^{(R)}(p)$ is concentrated in small values of p (modulo N), finding analysis and synthesis windows which concentrate as much weight as possible in a given range of coefficients can lead to a better approximation. We investigated this idea and performed an optimization of the analysis/synthesis windows for a 50% overlap and for $l = 2$, to maximize the L^2 norm of the 5×3 coefficients considered, assuming the analysis and synthesis windows were equal and symmetric. Quite remarkably, the window we obtained was very similar to the square root of the Hanning window, as can be shown in Fig. 3 for a window length of 512 samples. We thus used the square root of the Hanning window in the experiments we conducted. The central coefficients for this window are shown in Fig. 4.

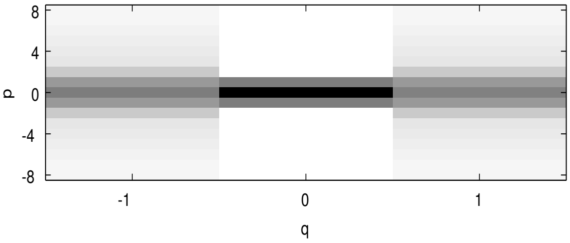

Figure 2: Magnitude of the central coefficients $\alpha_q^{(R)}(p)$ for $N = 512$, $R = 256$, a Hanning analysis window and a rectangular synthesis window.

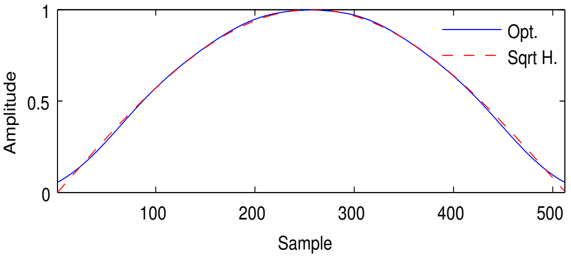

Figure 3: Comparison of the optimized window and the square root Hanning window for $N = 512$, $R = 256$ and $l = 2$.

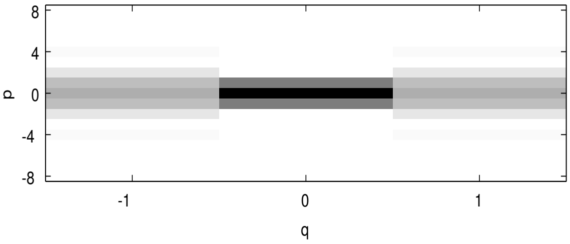

Figure 4: Magnitude of the central coefficients $\alpha_q^{(R)}(p)$ for $N = 512$, $R = 256$, and a square root Hanning analysis and synthesis window.

4.7. Time-scale modification

4.7.1. Need for an efficient frequency-domain algorithm

Many methods for time-scale and pitch-scale modification of acoustic signals have been proposed, and the interest on this subject intensified in recent years with the increase in the commercial application of such techniques. So far, most commercial implementations rely on time-domain methods, usually variations on Synchronous Overlap and Add (SOLA) or Pitch Synchronous Overlap and Add (PSOLA) techniques [12]. Their advantages are a low computational cost and good quality results for small modification factors (smaller than $\pm 20\%$ or $\pm 30\%$) and monophonic sounds. For larger factors, polyphonic sounds or nonpitched signals, however, the quality of the results drops severely. On the other hand, frequency-domain methods, such as the phase vocoder [13], are not limited to such constraints, but they involve a much higher computational cost and introduce artifacts of their own [6]. These artifacts have been shown to be mainly connected to phase incoherences, and special care must thus be taken when estimating the phases in the modified signal's STFT spectrogram. The iterative STFT algorithm of Griffin and Lim has been proposed as a way to correct such phase incoherences, although the computational cost and the slow speed of convergence have been obstacles to its adoption in commercial applications. The algorithm we introduced is a flexible alternative to iterative STFT, and by an active use of sparseness and the reduction of the number of multiplications involved at each step, should lead to a lower computational cost.

4.7.2. Sliding-block analysis for real-time processing

Inspired by an idea in [14], we derive a real-time optimization scheme for the objective function introduced above based on a sliding-block analysis. As illustrated in Fig. 5, the spectrogram is not processed all at once, but progressively from left to right, making it possible to change the parameters while sound is being played. In the particular case of time scale modification, this leads to the following procedure. The waveform to be time-scaled is read N samples at a time, where N is the window length. The STFT transform of this incoming frame is computed and adjoined to the frames of STFT spectrogram already computed, at the extreme right. If the block size is set to b and the frame shift to R, at a given time, we keep $b + 2Q$ frames, where $Q = N/R$ is the number of overlapping frames: the b central frames are updated using the algorithm derived above, through update equations (12), while the Q already processed frames on the left and the Q yet to be processed frames on the right are kept fixed and only used in the computations of the updates of the b central frames. Once the update has been performed, the frames are shifted to the left, and the frame which just exited the central block is inverse-DFTed and overlap-added, after windowing by the synthesis window, to the already computed part of the time-scaled waveform. The determination of the start of the next N sample part of incoming signal to be read is made in accordance with the time scale modification factor f such that the average shift for the incoming signal is fR, while keeping an integer shift at each step. The procedure is then iterated. The number of iterations performed on each frame is equal to the block size.

4.7.3. Experimental evaluation

We implemented the proposed method and the iterative STFT algorithm and compared their convergence speed on the time-scale modification of the first 23s of Chopin's Nocturne no.2. The time-scale modification factor was set to 0.7, the frame length to 1024 and the frame shift to 512, for a final length of approximately 32s. We used a 5×3 approximation of the coefficients $\alpha_q^{(R)}(p)$ for our algorithm. We ran our algorithm in two different experimental conditions, one where all the bins are updated at each iteration, and the other relying on sparseness, where at iteration k, only bins whose amplitude is larger than Ae^{-Bk} are updated. Here, we used $A = 1$ and $B = 0.005$, which were determined experimentally.

A comparison of the speed of convergence w.r.t. the number of iterations (or, equivalently, the block size) is shown in Fig. 6. The objective function $\mathcal{I}(H)$ is used as a measure of convergence, and represented in decibels, with the initial value as a reference. One can see that, although our algorithm is based on an approximation of the original objective function $\mathcal{I}(H)$, it outperforms the iterative STFT algorithm in terms of speed of convergence w.r.t. the number of iterations. The sparse version of our method has a slower convergence speed, close to the iterative STFT algorithm. This could be expected as only a part of the bins are updated. We also compared the computation times of the three methods, measuring only the time required by the phase reconstruction part of the algorithm as the other parts are identical. With our implementations, for 200 iterations, the iterative STFT algorithm took 35.4s, our method with full updates 31s, and our method using sparseness 2.5s. In terms of flexibility, speed of convergence and computation time, our algorithm thus outperforms the iterative STFT algorithm.

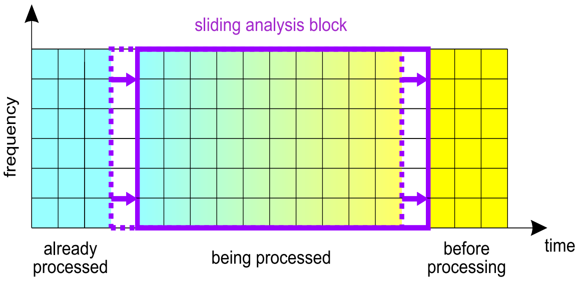

Figure 5: Illustration of the sliding-block analysis principle.

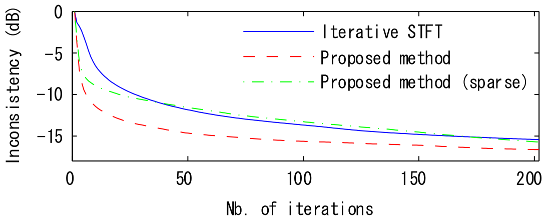

Figure 6: Comparison of the evolution of the inconsistency measure $\mathcal{I}(H)$ for the iterative STFT algorithm and the proposed method.

5. Audio encryption based on inconsistent STFT spectrograms

We can design a family of encryption codes based on any perfect reconstruction analysis/synthesis window couple by using a jammer in the STFT space. The key idea here is that, starting from any set of complex numbers $H \in \mathbb{C}^{M\times N}$, one can build an inconsistent STFT spectrogram with non-zero "energy" such that its inverse STFT is identically zero. In other words, for a given frame shift R and any perfect reconstruction analysis/synthesis window couple, there exists a family of sets of complex numbers in $\mathbb{C}^{M\times N}$ whose inverse STFT is identically silence. Indeed, starting from $H \in \mathbb{C}^{M\times N}$, let x_H be its inverse STFT signal. The STFT X_H of x_H is a consistent spectrogram, whose inverse STFT is also x_H. Thus the inverse STFT of $X_H - H$ is identically 0, although in general the L^2 norm of $X_H - H$ is not. The important point to note here is that this is only true for the synthesis window which was used in building X_H, and that in general for any other window the inverse STFT will not lead to silence.

We can apply this procedure to a set of random complex numbers to obtain a very "noisy" inconsistent spectrogram which, with the correct synthesis window, leads to silence when

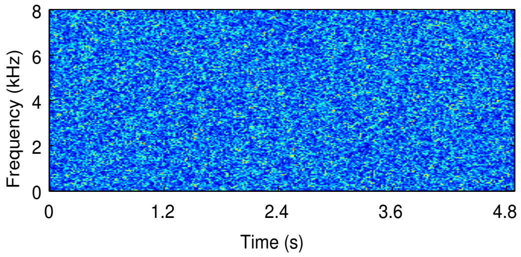

Figure 7: Magnitude of the inconsistent spectrogram.

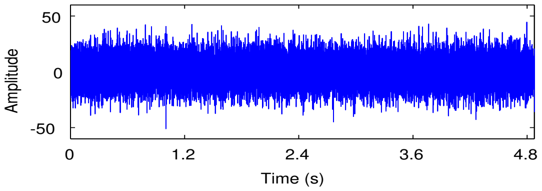

Figure 8: Waveform of the inverse STFT of the inconsistent spectrogram using a Hanning window.

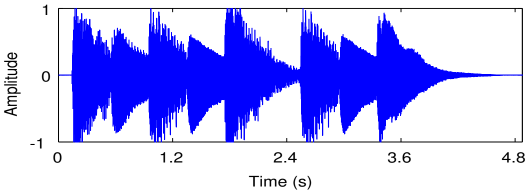

Figure 9: Waveform of the inverse STFT of the inconsistent spectrogram using the correct square root Hanning window.

inverse STFT-ed. Now, if this inconsistent noisy spectrogram, multiplied by a large coefficient, is added to the coherent spectrogram (built using the same window couple) of a speech or music sound for example, we obtain a set of complex numbers in which the power coming from the speech or music sound is masked and hardly detectable. An example of the magnitude of such a spectrogram is shown in Fig. 7, with the square root Hanning window as analysis/synthesis window, a window length $N = 512$ and window shift $R = 256$. The random set of complex numbers was generated by randomly modifying the phase of the spectrogram of a Gaussian white noise signal with standard deviation 1. Multiplied by a coefficient 100, it was added to the spectrogram of a computer generated music piece consisting of a mixture of piano and trumpet with a 16kHz sampling rate [3]. If the inverse STFT is performed with a different window than the one used to build the jammer spectrogram, the obtained signal is more or less noise. This can be seen in Fig. 8 where the inverse STFT of the spectrogram in Fig. 7 is computed using the Hanning window, leading to a Signal to Noise Ratio (SNR) of about -30dB. However, if the correct synthesis window is used, the jammer part of the spectrogram cancels off and the original speech or music sound is perfectly recovered (i.e., up to quantization error), as shown in Fig. 9. The synthesis window function thus acts as a key to decrypt the spectrogram and retrieve the hidden message.

Another way to produce interesting results, whose potential should be further investigated, is to start with an audio signal (white noise was used above, but speech or music can also be used), randomly change the phase of its STFT, and use the obtained set of complex numbers as a root for the procedure. The "hidden" sound can be heard for the correct window, while for other windows a distorted version of the root will be heard.

Although such issues as dynamic range limitation or codebreaking by a window optimization based on the minimization of the output power should be considered, we are now considering the potential applications of this technique as an encryption system.

6. Conclusion

We derived explicit consistency constraints for STFT spectrograms and showed how they could be used to develop a flexible phase reconstruction algorithm. We applied this algorithm for time scale modification, and explained how to perform a real-time processing based on sliding-block analysis. Finally, we showed how inconsistent STFT spectrograms could be used to hide sounds in the spectrogram domain.

Future works include the derivation of similar constraints for other transforms, such as the constant-Q transform, and the application of the proposed method to phase reconstruction of spectrograms whose phase is partially reliable, such as in gap interpolation problems [7]. We shall also consider using different schemes for the selection of the bins to update, for example by using different selectivities depending on the frequency.

7. References

[1] T. Virtanen, "Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria," *IEEE Trans. Audio, Speech, and Language Proc.*, vol. 15, no. 3, pp. 1066–1074, Mar. 2007.

[2] P. Smaragdis, "Discovering auditory objects through non-negativity constraints," in *Proc. SAPA*, 2004.

[3] M. N. Schmidt and M. Mørup, "Nonnegative matrix factor 2-D deconvolution for blind single channel source separation," in *Proc. ICA 2006*, Apr. 2006, pp. 700–707.

[4] J. Le Roux, H. Kameoka, N. Ono, A. de Cheveigné, and S. Sagayama, "Single channel speech and background segregation through harmonic-temporal clustering," in *Proc. WASPAA*, Oct. 2007, pp. 279–282.

[5] S. Boll, "Suppression of acoustic noise in speech using spectral subtraction," *IEEE Trans. Acoustics, Speech, and Signal Proc.*, vol. 27, pp. 113–120, Apr. 1979.

[6] J. Laroche and M. Dolson, "Improved phase vocoder time-scale modification of audio," *IEEE Trans. Speech and Audio Proc.*, vol. 7, no. 3, pp. 323–332, May 1999.

[7] J. Le Roux, H. Kameoka, N. Ono, A. de Cheveigné, and S. Sagayama, "Computational auditory induction by missing-data non-negative matrix factorization," in *Proc. SAPA*, Sep. 2008.

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

[9] W. I. Zangwill, *Nonlinear Programming: A Unified Approach*. Englewood Cliffs, NJ: Prentice Hall, 1969.

[10] T. Karrer, E. Lee, and J. Borchers, "PhaVoRIT: A phase vocoder for real-time interactive time-stretching," in *Proc. ICMC*, Nov. 2006, pp. 708–715.

[11] M. S. Puckette, "Phase-locked vocoder," in *Proc. WASPAA*, 1995.

[12] E. Moulines and J. Laroche, "Non-parametric techniques for pitch-scale and time-scale modification of speech," *Speech Communication*, vol. 16, pp. 175–206, 1995.

[13] M. Dolson, "The phase vocoder: A tutorial," *Comput. Music J.*, vol. 10, pp. 14–27, 1986.

[14] N. Ono, K. Miyamoto, H. Kameoka, and S. Sagayama, "A real-time equalizer of harmonic and percussive components in music signals," in *Proc. ISMIR 2008*, Sep. 2008.