Phase Reconstruction for Time-Frequency Inpainting

A. Marina Krémé[1,2]([✉]), Valentin Emiya[2], and Caroline Chaux[1]

[1] Aix Marseille Univ, CNRS, Centrale Marseille, I2M, Marseille, France
[2] Aix Marseille Univ, Université de Toulon, CNRS, LIS, Marseille, France
{A.Marina.Kreme,Valentin.Emiya,Caroline.Chaux}@univ-amu.fr

Abstract. We address the problem of phase inpainting, i.e. the reconstruction of partially-missing phases in linear measurements. We thus aim at reconstructing missing phases of some complex coefficients assuming that the phases of the other coefficients as well as the modulus of all coefficients are known. The mathematical formulation of the inverse problem is first described and then, three methods are proposed: a first one based on the well known Griffin and Lim algorithm and two other ones based on positive semidefinite programming (SDP) optimization methods namely PhaseLift and PhaseCut, that are extended to the case of partial phase knowledge. The three derived algorithms are tested with measurements from a short-time Fourier transform (STFT) in two situations: the case where the missing data are distributed uniformly and indepedently at random and the case where they constitute holes with a given width. Results show that the knowledge of a subset of phases contributes to improve the signal reconstruction and to shorten the convergence of the optimization process.

Keywords: Audio · Time-frequency · Missing data · Inpainting
Phase reconstruction · SDP optimization
Short-time Fourier transform · PhaseLift · PhaseCut

1 Introduction

Time-frequency inpainting is an inverse problem where the goal is to estimate a subset of masked coefficients in a time-frequency complex-valued matrix from the observation of the remaining coefficients. A natural strategy consists in performing a spectrogram inpainting stage, where the amplitude of the missing coefficients are estimated, followed by a phase inpainting stage, where the missing phases are estimated. While spectrogram inpainting has been addressed in several works [11,13,18], phase inpainting has not been addressed by advanced methods and thus remains a challenge. Indeed, phase reconstruction is known to be a difficult task generally posed as a non-convex problem. Many works have

been proposed to reconstruct the phase of all the time-frequency coefficients from their amplitude and may be extended to the phase inpainting problem. A first set of phase reconstruction methods relies on alternate projections [7–10] among which the Griffin and Lim (GL) algorithm [10] is widely used in audio processing. Its success may be due to the simplicity of its implementation and the low computational cost of its iterations. However, its performance is limited by a slow convergene towards a local minimum. Higher reconstruction performance has been reached by semidefinite programming (SDP) approaches, at the cost of much higher time and space complexities. In particular, PhaseLift [4] and PhaseCut [19] methods have been proposed for any linear operator and further studies [3,12] have established their good performance in the case of the short-time Fourier transform (STFT). While yet other phase reconstruction algorithms have been recently proposed [1,5,6,14–17], we focus on extending original GL and SDP approaches to phase inpainting.

The organization of the paper is as follows. In Sect. 2, the phase inpainting problem is formalized and we propose three dedicated algorithms: Griffin and Lim for phase inpainting (GLI), PhaseLift for phase inpainting (PLI) and PhaseCut for phase inpainting (PCI). These three algorithms are the extensions of existing algorithms, in which we add the knowledge of the partially observed phases. While the algorithms are introduced in the general case of any linear operator, Sect. 3 is dedicated to their specific implementation with the STFT operator. In Sect. 4, some experiments in small dimensions with various ratios of missing data and several mask shapes illustrate their performance and their limitations. Finally, conclusions and perspectives are drawn in Sect. 5.

2   Proposed Phase Inpainting Algorithms

2.1   Phase Inpainting Problem

For a signal $\mathbf{x} \in \mathbb{C}^N$, we consider K complex linear measurements $\mathbf{A}\mathbf{x} = [\langle \mathbf{a}_k, \mathbf{x} \rangle]_{k=1}^{K} \in \mathbb{C}^K$ where $\mathbf{a}_1, \ldots, \mathbf{a}_K \in \mathbb{C}^N$ and $\mathbf{A} = [\mathbf{a}_1, \ldots, \mathbf{a}_K]^H$. While the specific case of the STFT operator is used in Sects. 3 and 4, the general case of any linear operator is addressed throughout Sect. 2. We assume that we observe both the magnitude and the phase of a subset of measurements while only the magnitude of the remaining measurements is available. The location of these subsets is given by a binary mask $\boldsymbol{m} \in \{0,1\}^K$: $\boldsymbol{m}[k] = 1$ if both the magnitude and the phase of measurement k are known and $\boldsymbol{m}[k] = 0$ if only its magnitude is known.

Throughout the document we will adopt the following notations: for $\mathbf{b} \in \mathbb{C}^K$ $\angle \mathbf{b}$ denotes its phase, $\bar{\mathbf{b}}$ its conjugate. For $\boldsymbol{m} \in \{0,1\}^K$, $\neg$ denotes the negation. Denoting by $\mathrm{supp}\,(\boldsymbol{m})$ the support of $\boldsymbol{m}$, let $\mathbf{b} \in \mathbb{C}^K$ be the vector containing the fully known coefficients $\mathbf{b}[k]$ for $k \in \mathrm{supp}\,(\boldsymbol{m})$, and the known amplitudes $\mathbf{b}[k]$ for $k \in \mathrm{supp}\,(\neg\boldsymbol{m})$. Then the phase inpainting problem is given by

$$\text{Find } \mathbf{x} \in \mathbb{C}^N \text{ s.t. } \begin{cases} \langle \mathbf{a}_k, \mathbf{x} \rangle & = \mathbf{b}[k], \forall k \in \mathrm{supp}\,(\boldsymbol{m}) \\ |\langle \mathbf{a}_k, \mathbf{x} \rangle| & = \mathbf{b}[k], \forall k \; \mathrm{supp}\,(\neg\boldsymbol{m}) \end{cases} \tag{1}$$

2.2   Griffin and Lim Algorithm for Phase Inpainting (GLI)

We propose an extension of the Griffin and Lim algorithm [10] to solve approximately problem (1) by taking into account the known phases. The algorithm is described in Algorithm 1, ∘ denoting the Hadamard product. It mainly relies on alternating a projection onto the span of the linear operator using projector $\boldsymbol{\Pi_a}$ and a projection onto the known magnitude and phase constraints. The initialization of this algorithm may be done with random phases for coefficients with unknown phase.

Algorithm 1. Griffin and Lim algorithm for phase inpainting (GLI)

Require:
  binary mask $\boldsymbol{m} \in \{0,1\}^K$
  observation $\mathbf{b} \in \mathbb{C}^K$ such that $\begin{cases} \mathbf{b}[k] \in \mathbb{C}, & \forall k \in \text{supp}\,(\boldsymbol{m}) \text{ (fully known coefficients)} \\ \mathbf{b}[k] \in [0,\infty[, & \forall k \in \text{supp}\,(\neg\boldsymbol{m}) \text{ (known magnitudes)} \end{cases}$
  projector onto the span of the linear operator $\boldsymbol{\Pi_a}$
  initial phases $\boldsymbol{\varphi}_0 \in [0, 2\pi[^K$
  number of iterations n_{iter}
Output: complete estimated measurements $\mathbf{y}^{(n_{\text{iter}})}$
  $\boldsymbol{\varphi} \leftarrow \boldsymbol{m} \circ \angle\mathbf{b} + (1 - \boldsymbol{m}) \circ \boldsymbol{\varphi}_0 \quad \forall k \in \text{supp}\,(\boldsymbol{m})$
  $\mathbf{y}^{(0)} \leftarrow \mathbf{b} \circ \exp(\imath\boldsymbol{\varphi}) \quad \forall k \in \text{supp}\,(\neg\boldsymbol{m})$
  for $i \in \{1, 2, \dots, n_{\text{iter}}\}$ **do**
    $\mathbf{z}^{(i)} \leftarrow \boldsymbol{\Pi_a}\left(\mathbf{y}^{(i-1)}\right)$
    $\boldsymbol{\varphi}^{(i)} \leftarrow \boldsymbol{m} \circ \angle\mathbf{b} + (1 - \boldsymbol{m}) \circ \angle\mathbf{z}^{(i)}$ {Project onto phase constraints}
    $\mathbf{y}^{(i)} \leftarrow \mathbf{b} \circ \exp(\imath\boldsymbol{\varphi}^{(i)})$ {Project onto magnitude constraints}
  end for

2.3   PhaseLift for Phase Inpainting (PLI)

The second proposed approach is based on lifting and SDP. The quadratic constraints in problem (1) become linear by means of a projection in a large dimensional space where the variable is a semidefinite positive matrix $\mathbf{X} \succeq 0$. The PhaseLift method [4] is adapted in order to address phase inpainting, which results in Proposition 1.

Proposition 1. *With notations of problem* (1), *let* $\mathbf{A}_{lk} = \mathbf{a}_l\mathbf{a}_k^H$ *for* $l, k \in \{1, \dots, K\}$. *Using the lifting* $\mathbf{X} = \mathbf{x}\mathbf{x}^H$, *problem* (1) *is equivalent to:*

$$\min_{\mathbf{X} \in \mathbb{C}^{N \times N}} \text{Rank}(\mathbf{X}) \ s.t. \begin{cases} \text{Trace}(\mathbf{A}_{lk}\mathbf{X}) = \mathbf{b}[k]\bar{\mathbf{b}}[l], & \forall l, k \in \text{supp}\,(\boldsymbol{m}) \\ \text{Trace}(\mathbf{A}_{kk}\mathbf{X}) = \mathbf{b}^2[k], & \forall k \in \text{supp}\,(\neg\boldsymbol{m}) \\ \mathbf{X} \succeq 0 \end{cases} \quad (2)$$

and can be relaxed as :

$$\min_{\mathbf{X} \in \mathbb{C}^{N \times N}} \text{Trace}(\mathbf{X}) \ s.t. \begin{cases} \text{Trace}(\mathbf{A}_{lk}\mathbf{X}) = \mathbf{b}[k]\bar{\mathbf{b}}[l], & \forall l, k \in \text{supp}\,(\boldsymbol{m}) \\ \text{Trace}(\mathbf{A}_{kk}\mathbf{X}) = \mathbf{b}^2[k], & \forall k \in \text{supp}\,(\neg\boldsymbol{m}) \\ \mathbf{X} \succeq 0 \end{cases} \quad (3)$$

Proof. The proof can be conducted in three steps:

1. Assume that $\mathbf{x}$ satisfies (1). For $k, l \in \text{supp}(\boldsymbol{m})$, the phase constraint is obtained by considering that

$$\mathbf{b}[k]\bar{\mathbf{b}}[l] = \text{Trace}(\mathbf{a}_k^H \mathbf{x}\mathbf{x}^H \mathbf{a}_l) = \text{Trace}(\mathbf{a}_l \mathbf{a}_k^H \mathbf{x}\mathbf{x}^H) = \text{Trace}(\mathbf{A}_{lk} X)$$

For $k \in \text{supp}(\neg\boldsymbol{m})$, the magnitude constraint is obtained similarly.

2. Problem (1) can then be reformulated as

$$\text{Find } \mathbf{X} \in \mathbb{C}^{N \times N} \quad \text{s.t.} \quad \begin{cases} \text{Trace}(\mathbf{A}_{lk}\mathbf{X}) = \mathbf{b}[k]\bar{\mathbf{b}}[l], & \forall l, k \in \text{supp}(\boldsymbol{m}) \\ \text{Trace}(\mathbf{A}_{kk}\mathbf{X}) = \mathbf{b}^2[k], & \forall k \in \text{supp}(\neg\boldsymbol{m}) \\ \text{Rank}(\mathbf{X}) = 1 \\ \mathbf{X} \succeq 0 \end{cases}$$

which is equivalent to problem (2).

3. Since the rank is not convex, one may finally relax the rank by the nuclear norm to obtain Problem (3). □

Formulation (3) is called PhaseLift for phase inpainting (PLI). The objective function and equality constraints are linear and the domain $\mathbf{X} \succeq 0$ is a convex cone. One may notice that only phase differences appear, in the first contraint, to exploit the known phases. In the particular case $\text{supp}(\boldsymbol{m}) = \emptyset$, the original PhaseLift problem [4] is obtained.

Finally, from the solution $\mathbf{X}$ of problem (3), $\mathbf{x}$ can be estimated as $\sqrt{\lambda_{\max}}\mathbf{z}_{\max}$ where $\mathbf{z}_{\max}$ is the eigenvector associated with the largest eigenvalue $\lambda_{\max}$ of $\mathbf{X}$.

In order to solve the PLI problem (3), we use Matlab toolbox TFOCS [2]. Two solvers may be used: `solver_sSDP` that performs trace minimization under linear constraints as in (3), or `solver_TraceLS` that solves unconstrained problems of the form $\min_{\mathbf{X} \succeq 0} \lambda \, \text{Trace}(\mathbf{X}) + \frac{1}{2}\|\mathcal{A}(\mathbf{X}) - \beta\|^2$ with

$$\mathcal{A} : \mathbf{X} \mapsto \begin{bmatrix} vec\left([\text{Trace}(\mathbf{A}_{lk}\mathbf{X})]_{l,k\in\text{supp}(\boldsymbol{m})}\right) \\ [\text{Trace}(\mathbf{A}_{kk}\mathbf{X})]_{k\in\text{supp}(\neg\boldsymbol{m})} \end{bmatrix}, \beta = \begin{bmatrix} vec\left([\mathbf{b}[k]\bar{\mathbf{b}}[l]]_{l,k\in\text{supp}(\boldsymbol{m})}\right) \\ [\mathbf{b}[k]]_{k\in\text{supp}(\neg\boldsymbol{m})} \end{bmatrix}. \tag{4}$$

2.4   PhaseCut for Phase Inpainting (PCI)

The third and last proposed algorithm is also an SDP optimization algorithm, namely PhaseCut for phase inpainting (PCI), which is an extension of the original PhaseCut designed for phase retrieval [19].

As in [19], problem (1) is reformulated by explicitly splitting the amplitude and phase variables, so that one may optimize only on the phase vector $\mathbf{u} \in \mathbb{C}^K$ such that $\forall k, |\mathbf{u}[k]| = 1$. We use the lifting $\mathbf{U} = \mathbf{u}\mathbf{u}^H$ to obtain Proposition 2.

Proposition 2. *Using notations of problem* (1), *let* $\boldsymbol{\Gamma} = \text{Diag}(\mathbf{c}^H)(I - \mathbf{A}\mathbf{A}^\dagger)\text{Diag}(\mathbf{c})$ *with* $\mathbf{c} \in \mathbb{C}^K$ *is defined by* $\mathbf{c}[k] = |\mathbf{b}[k]|, \forall k$. *Then problem* (1) *is equivalent to*

$$\min_{\mathbf{U} \in \mathbb{C}^{K \times K}} \text{Trace}(\mathbf{U}\boldsymbol{\Gamma}) \ s.t. \begin{cases} \text{Diag}(\mathbf{U}) = \mathbb{1} \\ \mathbf{U}[k_1, k_2] = \frac{\mathbf{b}[k_1]}{|\mathbf{b}[k_1]|} \frac{\bar{\mathbf{b}}[k_2]}{|\mathbf{b}[k_2]|}, \forall k_1, k_2 \in \text{supp}(\boldsymbol{m}) \\ \text{Rank}(\mathbf{U}) = 1 \\ \mathbf{U} \succeq 0 \end{cases} \quad (5)$$

and may be relaxed into a convex problem by dropping the rank constraint as

$$\min_{\mathbf{U} \in \mathbb{C}^{K \times K}} \text{Trace}(\mathbf{U}\boldsymbol{\Gamma}) \ s.t. \begin{cases} \text{Diag}(\mathbf{U}) = \mathbb{1} \\ \mathbf{U}[k_1, k_2] = \frac{\mathbf{b}[k_1]}{|\mathbf{b}[k_1]|} \frac{\bar{\mathbf{b}}[k_2]}{|\mathbf{b}[k_2]|}, \forall k_1, k_2 \in \text{supp}(\boldsymbol{m}) \\ \mathbf{U} \succeq 0 \end{cases} \quad (6)$$

Proof. Using the amplitude vector $\mathbf{c}$ and the phase vector $\mathbf{u}$, problem (1) becomes

$$\text{Find } \mathbf{x} \in \mathbb{C}^N, \mathbf{u} \in \mathbb{C}^K \text{ s.t. } \begin{cases} \mathbf{A}\mathbf{x} & = \text{Diag}(\mathbf{c})\mathbf{u} \\ \mathbf{u}[k] & = e^{\imath \angle \mathbf{b}[k]} \forall k \in \text{supp}(\boldsymbol{m}) \\ |\mathbf{u}[k]| & = 1 \forall k \end{cases} \quad (7)$$

which is equivalent to

$$\min_{\mathbf{x} \in \mathbb{C}^N, \mathbf{u} \in [0, 2\pi[^K} \|\mathbf{A}\mathbf{x} - \text{Diag}(\mathbf{c})\mathbf{u}\|_2^2 \text{ s.t. } \begin{cases} \mathbf{u}[k] & = e^{\imath \angle \mathbf{b}[k]} \forall k \in \text{supp}(\boldsymbol{m}) \\ |\mathbf{u}[k]| & = 1 \forall k \end{cases}$$
$$(8)$$

Given that $\mathbf{A}\mathbf{x} = \text{Diag}(\mathbf{c})\mathbf{u}$ implies $\mathbf{x} = \mathbf{A}^\dagger \text{Diag}(\mathbf{c})\mathbf{u}$, then $\|\mathbf{A}\mathbf{x} - \text{Diag}(\mathbf{c})\mathbf{u}\|_2^2 = \mathbf{u}^H \boldsymbol{\Gamma} \mathbf{u}$, thus (8) is equivalent to (5) which can be relaxed into (6). $\qquad\square$

Formulation (6) is called PhaseCut for phase inpainting (PCI). As for PLI, phase differences appear in the constraints that involve known phases. In the particular case where all phases are unknown $(\text{supp}(\boldsymbol{m}) = \emptyset)$, contraints $\mathbf{U}[k_1, k_2] = \frac{\mathbf{b}[k_1]}{|\mathbf{b}[k_1]|} \frac{\bar{\mathbf{b}}[k_2]}{|\mathbf{b}[k_2]|}$ disappear and the original PhaseCut problem [19] is obtained $\mathbf{x}$.

Finally, from the solution $\mathbf{U}$ of problem (6), signal $\mathbf{x}$ is estimated as $\mathbf{x} = \mathbf{A}^\dagger \text{Diag}(\mathbf{c}) e^{\imath \angle \mathbf{u}_{\max}}$ where $\mathbf{u}_{\max}$ is an eigenvector associated to the largest eigenvalue of $\mathbf{U}$.

In order to solve PCI problem (6), we adapt the block coordinate descent algorithm proposed in [19] from [20], as given in Algorithm 2. By picking coordinates i in $\text{supp}(\boldsymbol{m})$ instead of $\{1, \ldots, K\}$, all unknown coefficients in $\mathbf{U}$, and only them, are updated.

Algorithm 2. PhaseCut for phase inpainting (PCI) : BCD algorithm

---

Require:
  binary mask $\boldsymbol{m} \in \{0,1\}^K$

  observation $\mathbf{b} \in \mathbb{C}^K$ such that $\begin{cases} \mathbf{b}[k] \in \mathbb{C}, & \forall k \in \text{supp}\,(\boldsymbol{m}) \text{ (fully known coefficients)} \\ \mathbf{b}[k] \in [0,\infty[, & \forall k \in \text{supp}\,(\neg\boldsymbol{m}) \text{ (known magnitudes)} \end{cases}$

  number of iterations n_{iter}
  barrier parameter $\nu > 0$
Output: $\mathbf{U} \in \mathbb{C}^{K \times K}$
  {Initialization}
  $\mathbf{c} \leftarrow \boldsymbol{m} \circ \mathbf{b} + (1 - \boldsymbol{m}) \circ \mathbf{b}$
  $\boldsymbol{\Gamma} \leftarrow \text{Diag}(\mathbf{c}^H)(I - \mathbf{A}\mathbf{A}^{\dagger})\text{Diag}(\mathbf{c})$

  for $1 \le k, l, \le K, \mathbf{U}\,[k, l] \leftarrow \begin{cases} 1 & \text{if } k = l \\ \frac{\mathbf{b}[k]}{|\mathbf{b}[k]|}\frac{\overline{\mathbf{b}[l]}}{|\mathbf{b}[l]|} & \text{if } k, l \in \text{supp}\,(\boldsymbol{m}) \\ 0 & \text{otherwise} \end{cases}$

  {Main loop}
  for n_{iter} iterations **do**
     pick $i \in \{1, \ldots, K\} \setminus \text{supp}\,(\boldsymbol{m})$
     $\mathbf{x} \leftarrow \mathbf{U}_{i^c,i^c}\boldsymbol{\Gamma}_{i^c,i}$ and $\gamma \leftarrow \mathbf{x}^H \boldsymbol{\Gamma}_{i^c,i}$
     $\mathbf{U}_{i^c,i}, \mathbf{U}_{i^c,i}^H \leftarrow \begin{cases} -\sqrt{\frac{1-\nu}{\gamma}}\mathbf{x} & \text{if } \gamma > 0 \\ 0 & \text{otherwise} \end{cases}$
  end for

---

3   Implementation Issues Specific to the STFT

Phase Inpainting Problem with STFT Measurements. The STFT of a signal $\mathbf{x} \in \mathbb{C}^N$ is defined for frame index $t \in \{0, \ldots, T-1\}$ and frequency index $\nu \in \{0, \ldots, F-1\}$ as $\text{STFT}[t, \nu] = \langle \mathbf{x}, \mathbf{a}_{t,\nu} \rangle = \mathbf{a}_{t,\nu}^H \mathbf{x}$ where $\mathbf{a}_{t,\nu} = \left[\mathbf{w}[n - th]e^{2\iota\pi\frac{\nu}{F}n}\right]_{n=0}^{N-1} \in \mathbb{C}^K$, $\mathbf{w}$ being the analysis window and h the so-called *hop size* between two successive frames. Hence the $K = FT$ measurements are indexed by $k = (t, \nu)$: measurements may be seen equivalently either as a doubly-indexed vector or as a matrix. A simple reshaping operation can be used to switch between representations, and with a small abuse of notations, both of them are used without explicit distinction in this paper. The STFT phase inpainting problem in time-frequency is thus given by

$$\text{Find } \mathbf{x} \in \mathbb{C}^N \text{ s.t. } \begin{cases} \langle \mathbf{x}, \mathbf{a}_{t,\nu} \rangle = \mathbf{b}[t, \nu], & \forall (t, \nu) \in \text{supp}\,(\boldsymbol{m}) \\ |\langle \mathbf{x}, \mathbf{a}_{t,\nu} \rangle| = \mathbf{b}[t, \nu], & \forall (t, \nu) \in \text{supp}\,(\neg\boldsymbol{m}) \end{cases} \tag{9}$$

GLI Implementation. The GLI algorithm is obtained by setting $\boldsymbol{\Pi}_{\mathbf{a}} : \mathbf{y} \mapsto \text{STFT}\left(\text{STFT}^{-1}(\mathbf{y})\right)$ where STFT^{-1} is the (pseudo-)inverse operator for the STFT computed from the canonical dual window of $\mathbf{w}$.

PLI Implementation. We used `solver_TraceLS` of TFOCS library, which happened to be faster than `solver_sSDP`. The implementation of the direct operator $\mathcal{A}$ defined in (4) and of its adjoint can be more efficient using fast Fourier transforms (FFT) as follows. We have $\text{Trace}(\mathbf{A}_{lk}\mathbf{X}) = \left(\text{STFT}_{row}\left(\overline{\text{STFT}_{col}(\mathbf{X})}\right)\right)^H [k, l]$ for $k, l \in \{1, \ldots, K\}$, where $\text{STFT}_{row}(\mathbf{X})$ denotes the STFT on the columns of $\mathbf{X}$ and $\text{STFT}_{col}(\mathbf{X})$ the STFT on the

rows of $\mathbf{X}$. Hence one may compute $\mathcal{A}(\mathbf{X})$ from only $2N$ STFT's. By denoting by $k_0 = \#\mathrm{supp}(\boldsymbol{m})$ the number of known phases, the adjoint operator $\mathcal{A}^* : \mathbb{C}^{k_0^2+K-k_0} \to \mathbb{C}^{N\times N}$ is such that $\mathcal{A}^*(\mathbf{y}) = \left(\mathrm{STFT}_{row}^* \left(\overline{\mathrm{STFT}_{col}^*(\mathbf{Y})}\right)\right)^H$ where STFT^* is the adjoint of the STFT operator and $\mathbf{Y} \in \mathbb{C}^{K\times K}$ is defined by $\mathbf{Y}(\boldsymbol{m}, \boldsymbol{m}) = reshape(\mathbf{y}(1:k_0^2), k_0, k_0)$ and $\mathbf{Y}(\sim \boldsymbol{m}, \sim \boldsymbol{m}) = \mathrm{Diag}(\mathbf{y}(k_0^2+1:K))$. It thus requires $2K$ calls to STFT^*.

PCI Implementation. Each iteration of Algorithm 2 for PCI requires K calls to one direct STFT and one inverse STFT, using FFT's.

4   Experiments

All experiences are available on mad.lis-lab.fr. Experiments in small dimensions are conducted on a signal with length $N = 128$ composed of the sum of two linear chirps with normalized frequency ranges $(0, 0.8)$ and $(0.8, 0.6)$, a dirac located at sample 64 and white Gaussian noise at a signal-to-noise ratio of $10\,\mathrm{dB}$. The STFT is generated with a Hann window with length 16, a hop size of 8 samples (i.e., $T = 16$ frames) and $F = 32$ frequency bins, resulting in $K = 512$ measurement in a 32×16 time-frequency matrix. In a first experiment, masks for missing phases are generated randomly and uniformly among the measures, with various ratios of missing phases. In second experiment, the ratio of missing phases is fixed at 30% and missing phases are grouped in holes of a given width, with randomly distributed centers, the widths varying between 1 and 9 coefficients. Figure 1 illustrates the STFT of the signal and of one generated mask.
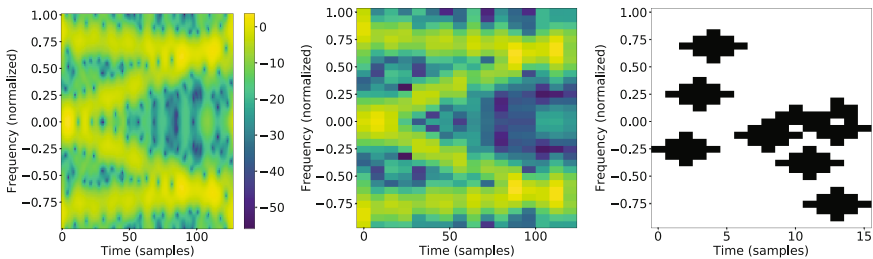

Fig. 1. Spectrogram of the signal (left, smoothed with $T = F = 128$; middle, with $T = 16$ and $F = 32$ as set in the experiment) and example of a mask with random holes of width 5 in black (right).

Algorithms are used with the following settings. For GLI, $n_{\mathrm{iter}} = 6000$. For PLI, $\lambda = 10^{-30}$ and TFOCS is used with a maximum of 5000 iterations, no restart, $tol = 10^{-10}$. For PCI, $\nu = 10^{-14}$ and $n_{\mathrm{iter}} = 10^5$. A baseline approach is also used, denoted as Random Phase Inpainting (RPI) and consisting in filling the missing phases by drawing random values independently and uniformly in $[0, 2\pi\,[$.

Performance is assessed in terms of relative reconstruction error up to a global phase shift, defined by $E_{dB}(\mathbf{x}, \hat{\mathbf{x}}) = 20 \log_{10} \min_\theta \frac{\|\mathbf{x} - e^{i\theta}\hat{\mathbf{x}}\|_2}{\|\mathbf{x}\|_2}$ where $\mathbf{x}$ denotes the original signal and $\hat{\mathbf{x}}$ the reconstructed one.

Results are shown in Fig. 2, where the reconstruction errors from all methods can be compared, as a function of the ratio of missing phases, for each experiment. The known phases clearly contribute to improve the signal reconstruction. For isolated missing phases (left figure), one can see that below 40% missing phases, GLI and PCI achieves perfect reconstruction while PLI performs very good but not perfect. Beyond 40% missing phases, SDP methods PLI and PCI perform better than GLI, with a much better performance for PLI. For holes with a larger width at 30% missing phases (right figure), one can see that GLI may achieve poor reconstruction due to local minima. SDP methods offer very good performance, with a reconstruction error generally below $-50\,\mathrm{dB}$.
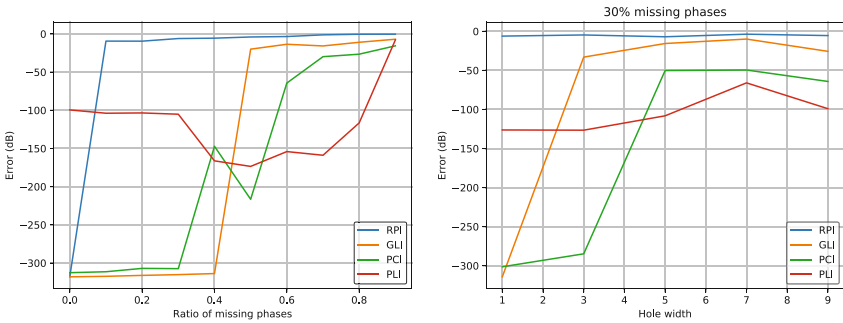
Fig. 2. Reconstruction error as a function of the ratio of missing phases randomly distributed (left) or, for 30% missing phases, as a function of the width of randomly distributed holes (right).

The convergence and running time of each method have been checked as follows. For GLI, it was checked visually and manually that the algorithm converges before the maximum number of iterations, with a running time lower than one second for each call. For PLI, similarly, it has been checked that the algorithm stops before the maximum number of iterations is reached, with a running time up to 6 h for one call when the number of missing phases is large. For PCI, convergence may be observed in Fig. 3 by representing the reconstruction error as a function of the iterations. As for PLI, the running time until convergence is all the more reduced as many phases are known, lasting about 4 h for 10^5 iterations.
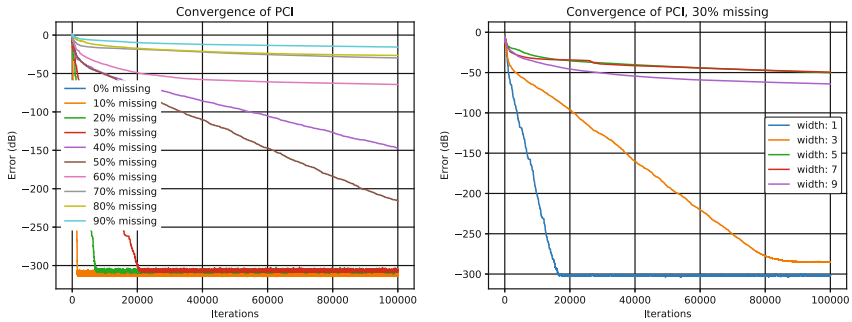
Fig. 3. Illustration of the convergence of PCI by representing the reconstruction error as a function of the iterations in the same two settings as in Fig. 2.

5   Conclusion and Perspectives

We have considered the phase inpainting problem in which a subset of measurements have missing phases that must be recovered. We have proposed three dedicated algorithms, which are extensions of existing algorithms, namely Griffin and Lim, PhaseLift and PhaseCut, adapted to the phase inpainting problem by incorporating the partial phase information as constraints in the optimization process. Those algorithms have been implemented using fast transforms in the case of the STFT. Experiments in small dimensions confirm that SDP methods perform better than Griffin and Lim algorithm, in particular when the problem is difficult (more unknown phases, larger holes). Even if those methods are very time consuming, it also appears that the knowledge of a subset of phases result in a faster convergence.

Experiments may be extended to the noisy case where only approximate values are available known phases and amplitudes. Time and space complexity of SDP approaches being very large, they cannot be applied to typical audio signals for which dimensions are higher than those used in the proposed experiments. In order to benefit from SDP results, one may investigate the adaptation of SDP algorithms to process only a local time-frequency region instead of the whole STFT matrix. Other algorithms may be designed for phase inpainting. In particular, some recent contributions to phase retrieval [1,5,6,14–17] may be adapted and may give good performance without the computational limits of SDP methods.

References

1. Bahmani, S., Romberg, J.: Phase retrieval meets statistical learning theory: a flexible convex relaxation. arXiv:1610.04210v2 (2017)
2. Becker, S., Candès, E.J., Grant, M.: Templates for convex cone problems with applications to sparse signal recovery. Technical report, Department of Statistics, Stanford University (2010)

3. Bendory, T., Eldar, Y.C., Boumal, N.: Non-convex phase retrieval from STFT measurements. IEEE Trans. Inform. Theory **64**(1), 467–484 (2018)
4. Candès, E.J., Eldar, Y., Strohmer, T., Voroninski, V.: Phase retrieval via matrix completion. SIAM Rev. **57**(2), 225–251 (2015)
5. Candès, E.J., Li, X., Soltanolkotabi, M.: Phase retrieval via wirtinger flow: theory and algorithms. IEEE Trans. Inform. Theory **61**(4), 1985–2007 (2015)
6. A. Dremeau and F. Krzakala. Phase recovery from a bayesian point of view: The variational approach. In IEEE Trans. Acous., Speech Signal Process. IEEE, Apr 2015
7. Fienup, J.R.: Reconstruction of an object from the modulus. Opt. Lett. **3**, 27–29 (1978)
8. Fienup, J.R.: Phase retrieval algorithms : a comparison. Appl. Opt. **21**(15), 2758–2769 (1982)
9. Gerchberg, R.W., Saxton, W.: A practical algorithm for the determination of phase from image and diffraction plane pictures. Optik **35**, 237–246 (1972)
10. Griffin, D., Lim, J.: Signal estimation from modified short-time Fourier transform. IEEE Trans. Acous. Speech Signal Process. **32**(2), 236–243 (1984)
11. Hamon, R., Emiya, V., Févotte, C.: Convex nonnegative matrix factorization with missing data. In: Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP) (2016)
12. Jaganathan, K., Eldar, Y.C., Hassibi, B.: STFT phase retrieval: uniqueness guarantees and recovery algorithms. IEEE J. Sel. Topics Signal Process. **10**(4), 770–781 (2016)
13. Le Roux, J., Kameoka, H., Ono, N., de Cheveigné, A., Sagayama, S.: Computational auditory induction as a missing-data model-fitting problem with bregman divergence. Speech Commun. **53**, 658–676 (2010)
14. Metzler, C.A., Sharma, M.K., Nagesh, S., Baraniuk, R.G., Cossairt, O., Veeraraghavan, A.: Coherent inverse scattering via transmission matrices: efficient phase retrieval algorithms and a public dataset. In: 2017 IEEE International Conference on Computational Photography (ICCP). IEEE, May 2017
15. Netrapalli, P., Jain, P., Sanghavi, S.: Phase retrieval using alternating minimization. Adv. Neural Inf. Process. Syst. **26**, 2796–2804 (2013)
16. Prusa, Z., Balazs, P., Sondergaard, P.: A non-iterative method for reconstruction of phase from stft magnitude. IEEE Trans. Audio Speech Lang. Process. **25**, 1154–1164 (2017)
17. Rajaei, B., Gigan, S., Krzakala, F., Daudet, L.: Robust phase retrieval with the swept approximate message passing (prSAMP) algorithm. Image Process. On Line **7**, 43–55 (2017)
18. Smaragdis, P., Raj, B., Shashanka, M.: Missing data imputation for spectral audio signals. In: Proceedings of MLSP, Grenoble, France (2009)
19. Waldspurger, I., d'Aspremont, A., Mallat, S.: Phase recovery maxcut and complex semidefinite programming. Math. Prog. **149**(12), 47–81 (2015)
20. Wen, Z., Goldfarb, D., Scheinberg, K.: Block coordinate descent methods for semidefinite programming. In: Anjos, M., Lasserre, J. (eds.) Handbook on Semidefinite, Conic and Polynomial Optimization. International Series in Operations Research & Management Science, vol. 166, pp. 533–564. Springer, Boston (2012). https://doi.org/10.1007/978-1-4614-0769-0_19