LTFAT - Die Zeit-Frequenz Toolbox

Peter Balazs[1], Nicki Holighaus[1]

[1] *Institut für Schallforschung, 1040 Wien, Österreich, Email: {peter.balazs,nicki.holighaus}@oeaw.ac.at*

Einleitung

Die Large Time Frequency Analysis Toolbox (LTFAT) ist eine moderne Octave / Matlab-Toolbox, in der Zeit-Frequenz-Analyse, -Synthese, Koeffizientenmanipulation und Visualisierung implementiert sind. Ziel ihrer Entwicklung ist es, sowohl als Instrument wissenschaftlicher Forschung als auch der Fortbildung zu dienen.

Die LTFAT-Toolbox bietet eine Reihe von Werkzeugen für Zeit-Frequenz-Darstellungen sowie weitere Signalverarbeitungsmethoden. Die wichtigsten Konzepte und Implementierung werden in Beispielen (Demos `demo_`) erklärt. Die zentralen Eigenschaften dieser Toolbox sind: Sie ist Open Source, gut dokumentiert, wird gewartet und ist frei im Netz verfügbar.

Wir geben hier einen Überblick über die Hauptfunktionen von LTFAT, darunter Kurzzeit-Fourier- und Gabor-Transformationen. Wir erwähnen die zugrundeliegende Theorie der Frames. Von diesen Konzepten sowie Filterbänken und zeitinvarianten Filtern berichten wir, wie sie in LTFAT implementiert sind. Wir schließen mit einem geschichtlichen Rückblick und einer Danksagung.

Dieser Artikel präsentiert einen Überblick über LTFAT, für Details sei, unter anderem, auf [24, 22] verwiesen. Die gesamte Toolbox kann über `ltfat.github.io` geladen werden, wo auch die Dokumentation vorliegt.

Frequenz Transformationen

Die Fourier-Analyse [26] erlaubt die Darstellung von Signalen als lineare Kombinationen von Schwingungen. Für reelle Signale f mit L Abtastpunkten, i.e. $f \in \mathbb{R}^L$, ist sie durch

$$\hat{f}(k) = \frac{1}{\sqrt{L}} \sum_{l=0}^{L-1} f(l) e^{-2\pi i k l/L} \text{ für } k = 0, \dots, L-1, \quad (1)$$

definiert. In LTFAT ist sie mit `dft` implementiert, ihre Umkehrung mit `idft`.

Da weder aus dem Zeit-Signal noch aus der Fourier-Transformierten alle relevanten Informationen für zeitveränderliche Signale wie Musik oder Sprache direkt ersichtlich sind, ist häufig eine andere Darstellung notwendig.

Die Kurz-Zeit-Fourier-Transformierte

Der Prototyp der Zeit-Frequenz-Transformation ist die *Kurzzeit-Fourier-Transformierte* (engl. Abk. STFT). Sie ist im Wesentlichen eine Fourier-Transformation von durch Fensterung lokalisierten Signalteilen und ist für $f \in \mathbb{R}^L$ definiert durch

$$\mathrm{STFT}(m,n) = \sum_{l=0}^{L-1} f(l) g(l-n) e^{-2i\pi m l/L}, \quad (2)$$

für $m, n = 0, \dots, L-1$. Die STFT ist ein häufig verwendetes Analysewerkzeug [8]. Die einfachste Wahl des Fensters g, nämlich ein Rechteck-Fenster, erweist sich als in der Praxis wegen seiner schlechten spektralen Lokalisation ungeeignet und glattere Fensterfunktionen werden im Allgemeinen bevorzugt (siehe unten).

Die STFT ist invertierbar: Eine direkte Berechnung ergibt

$$f(l) = \frac{1}{\|g\|^2} \sum_{m,n=0}^{L-1} \mathrm{STFT}(m,n) g(l-n) e^{2i\pi m l/L}. \quad (3)$$

Die STFT ist eine hoch redundante Darstellung des Signals, da für L Abtastpunkte in der Zeit $L \times L$ Punkte in der Zeit-Frequenz-Ebene verwendet werden. Wenn man nicht die volle STFT benützen, sondern eine abgetastete Version verwenden will, gelangt man zur Gabor-Transformation. Insofern wird die STFT in LTFAT als Spezialfall der Gabor-Transformation durch `dgt` berechnet.

Die diskrete Gabor-Transformation

Die STFT-Inversionsformel (3) kann als Entwicklung des Signals f in Bezug auf ein (redundantes) System von elementaren, lokalisierten Wellenformen $g_{m,n}$ interpretiert werden, wobei

$$g_{mn}(t) = e^{\frac{2i\pi m t}{M}} g(t - na), \quad (4)$$

zeit- und frequenzverschobene Kopien des Fensters g sind. Hierbei ist a die Schrittweite in der Zeit, M die Anzahl der Frequenz-Punkte und L die Länge des Signals in Abtastpunkten. (Für die volle STFT ist $a = 1$ und $L = M$.) Dafür wird implizit vorausgesetzt, dass es ganze Zahlen N, b gibt, sodass $L = Mb = Na$. Das kann in der Praxis durch das Hinzufügen von Nullen zum Signal erreicht werden, in LTFAT mittels `postpad`. Mit diesen Parametern werden die Gabor-Koeffizienten in $\mathbb{C}^{M \times N}$ berechnet durch:

$$G_g f(m,n) = \sum_{l=0}^{L-1} f(k) e^{-2\pi i m l/M} g(l - an). \quad (5)$$

Hier ist noch anzumerken, dass Mathematiker die STFT deutlich von der Gabor-Transformation unterscheiden [13, 11], während Techniker den Begriff Kurzzeit-Fourier-Transformation auch für den Gabor-Fall (also $a > 1$ und/oder $b > 1$) verwenden [1].

Für die volle STFT ist die Inversion immer möglich, für Gabor-Systeme muss man Bedingungen dafür überprüfen. Dies wird durch das Frame-Konzept ermöglicht.

Rahmen (Frames)

Eine Familie von Vektoren $\psi_j \in \mathbb{R}^L$ für $j = 0, \ldots, J-1$ heißt *Rahmen* oder *Frame* (engl.) [9], wenn Konstanten $0 < A \leq B < \infty$ existieren, sodass

$$A \|f\|^2 \leq \sum_{j=0}^{J-1} |\langle f, \psi_j \rangle|^2 \leq B \|f\|^2 \quad \forall f \in \mathbb{R}^L. \qquad (6)$$

Die Konstanten A und B werden als untere und obere Frameschranken bezeichnet. Ist Gleichheit, also $A = B$, möglich, so heißt der Frame *eng* oder *tight* (engl.). In endlichdimensionalen Vektorräumen (wie $\mathbb{R}^L$) ist ein Rahmen im wesentlichen eine Familie von Vektoren, die eine Darstellung aller Signale in diesem Raum erlauben. Die Ungleichung auf der rechten Seite ist für $\mathbb{R}^L$ immer erfüllt. Die Frameschranken sind trotzdem relevant, insbesondere für numerische Eigenschaften; sie bestimmen etwa die Konditionszahl der Transformation. Wenn $J > L$, ist der Rahmen redundant. Die Redundanz des Rahmens ist durch den Quotienten $red = \frac{J}{L}$ gegeben. Ein Frame kann eine Basis sein (also $red = 1$), aber interessante Frames sind oft redundant. Damit sind die Koeffizienten nicht mehr eindeutig, eine Rekonstruktion ist aber trotzdem immer möglich.

Rahmen haben das schöne und nützliche Merkmal der Existenz von (im Allgemeinen unendlich vielen) *dualen Rahmen*, die perfekte Rekonstruktion erlauben. Genauer gesagt gibt es für den Rahmen $(\psi_j)_{j=0,\ldots,J-1}$ immer zumindest einen anderen Rahmen $\left(\tilde{\psi}_j\right)_{j=0,\ldots,J-1}$, sodass für alle $f \in \mathbb{R}^L$,

$$f = \sum_{j=0}^{J-1} \langle f, \psi_j \rangle \tilde{\psi}_j = \sum_{j=0}^{J-1} \langle f, \tilde{\psi}_j \rangle \psi_j.$$

Insbesondere existiert für jeden Rahmen der *kanonisch duale Rahmen*, der wie folgt gegeben ist:

Definieren wir den *Frame Operator* $S : \mathbb{R}^L \to \mathbb{R}^L$, durch

$$(Sf)(l) = \sum_{j=0}^{J-1} \langle f, \psi_j \rangle \psi_j(l) \text{ für } l = 0, \ldots L-1, \qquad (7)$$

so kann gezeigt werden, dass S invertierbar ist. Der kanonisch duale Rahmen wird dann definiert durch

$$\tilde{\psi}_j = S^{-1} \psi_j \text{ für } j = 0, \ldots J-1. \qquad (8)$$

Im speziellen Fall von Gabor-Frames ist der kanonische duale Rahmen wieder ein Gabor-Frame und wird durch das *kanonisch dualen Fenster* erzeugt:

$$\tilde{g} = S^{-1} g. \qquad (9)$$

Um eine Transformation mit perfekter Rekonstruktion zu erhalten, in welcher das ursprüngliche und das duale Fenster eine gute Zeit-Frequenz Konzentration haben, ist es notwendig, dass das Gabor-System redundant, also ein Rahmen, ist. Das ist, vereinfacht, der Inhalt des Satzes von Balian und Low [13].

In LTFAT ist die diskrete Gabor-Transformation in der Funktion `dgt` implementiert, wobei alle Parameter sowie das Fenster frei gewählt werden können. Die inverse Gabor-Transformation ist in `idgt` implementiert. Für Audiosignale wird empfohlen die Routine `dgtreal` zu verwenden (dabei wird ausgenützt, dass das Signal reell ist und man nur die positiven Frequenzen berücksichtigen muss). Fenster werden von den Routinen `firwin` generiert, darin sind die klassischen Fenster Hann, Hamming, Blackman, Nuttall und andere [19] enthalten. Es gibt noch andere Fenster wie zum Beispiel das Gauss-Fenster, das optimale Zeit-Frequenz-Konzentrations-Eigenschaften hat, `pgauss`.

Um mit LTFAT zu beginnen, empfiehlt sich die Verwendung des Spektrogramms `sgram`. Diese einfach zu bedienende Funktion zeigt das sogenannte Spektrogramm, die quadrierte Amplitude der Gabor-Transformation $|G_g f(m,n)|^2$ des Signals. Neben einer linearen Farbskala kann das Spektrogramm wahlweise auch mit logarithmischer Farbskala dargestellt werden. Ein einfaches Beispiel findet man in `demo_dgt`, siehe auch Abbildung 1.

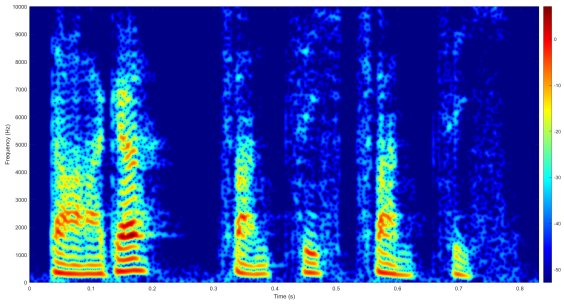

Abbildung 1: Die Abbildung zeigt ein Spektogramm des Sprachsignals „DAGA2020“ von einem Österreicher gesprochen. Dazu wurde `sgram(y(:,1),fs,'dynrange',60,'fmax',10000); colormap(jet)` verwendet.

LTFAT Frames Framework

Der Begriff eines Rahmens passt sehr gut zum Begriff einer *Klasse* im objektorientierten Programmierparadigma. Eine Klasse ist eine Sammlung von Methoden und Variablen, die zusammen eine logische Einheit bilden. Eine Klasse kann von einer anderen abgeleitet werden. Abgeleitete Eigenschaften der ursprünglichen Klasse können direkt verwendet oder erweitert werden. Die Frame-Klasse dient als abstrakte Basisklasse, aus der alle anderen Transformationen abgeleitet werden, darunter auch die `dft` und die `dgt`.

Ein Objekt vom Typ `frame` wird vom Benutzer instanziiert mit der Information, welche Art von Rahmen gewünscht wird, sowie allen zusätzlichen Parametern (wie bei der `dgt` die Fensterfunktion, die Anzahl der zu Frequenzpunkte M, usw.). Grundlegende Informationen zu einem Frame erhält man etwa durch `framebounds`, was die Frame-Schranken berechnet. Die Arbeitspferde des Frame-Frameworks sind die `frana` und `frsyn` Methoden, die die Analyse und Synthese des Rahmens bereit-

stellen. Diese Methoden verwenden einen schnellen Algorithmus, falls für den angegebenen Frame einer verfügbar ist. Die Methode `framedual` berechnet den kanonisch dualen Rahmen (wie in (8)).

Filterbänke

Sehr oft werden zeit-invariante Filter verwendet, also Systeme, die auf einer Faltung mit der sogenannten Impulsantwort beruhen. Diese entsprechen einem Multiplikator auf Frequenz-Ebene (mit der Transferfunktion). Diese Technik wird seit vielen Jahren angewendet und findet ein breites Anwendungsspektrum in allen Bereichen der Signalverarbeitung [19].

Mehrere dieser Filter bilden eine *Filterbank (FB)*, ein klassisches Signalanalyse- und -verarbeitungswerkzeug. Die allgemeine, möglicherweise ungleichmäßige, Struktur bietet die natürliche Umgebung für den Entwurf flexibler, frequenzadaptiver Zeit-Frequenz-Signaldarstellungen [5]. Filterbänke, die die Frame-Bedingung erfüllen, erlauben perfekte Rekonstruktion [6].

In der Toolbox sind Filterbänke mit `filterbank` implementiert. Mit `ifilterbank` wird eine direkte Rekonstruktion erlaubt. In diesem Kontext sind z.B. die constant-Q [14], die nicht-stationäre Gabor [4], die Audlet- [18] sowie die Wavelettransformation [23] implementiert.

Zeitvariante Filter

Eine Verallgemeinerung von linearen zeit-invarianten Filtern sind *zeitvariante Filter* [27]. Dort variiert die Impulsantwort mit der Zeit und eine Multiplikation in der Fourier-Domäne ist somit nicht mehr ausreichend. Eine Möglichkeit, einen zeitvarianten Filter zu konstruieren, ist die Verwendung von *Gabor-Multiplikatoren* [10]. Letztere bestehen aus der Gabor-Transformation, gefolgt durch eine punktweise Multiplikation mit einem festen Zeit-Frequenz-Vektor (genannt das Symbol $\mathbf{m}(m,n)$) und eine Transformation durch die inverse Gabor-Transformation wieder in die Zeitebene:

$$\mathbb{M}_{\mathbf{m}} f = \sum_{m,n} \mathbf{m}(m,n) G_g f(m,n) g_{m,n}. \qquad (10)$$

Diese Operatoren werden schon seit geraumer Zeit implizit in Anwendungen verwendet. Etwa in der Signalverarbeitung werden diese zeitvariante Filtern unter dem Namen *Gabor Filter* [17] benützt. Aktuelle Anwendungen finden sie zum Beispiel in den Bereichen Quellentrennung [28], Systemschätzung [16] oder psychoakustische Modellierung [7].

Die Toolbox enthält eine Routine `gabmul` für die Anwendung von Gabor-Multiplikatoren, deren Analyse- und Synthesefenster, Gitterkonstanten und Symbole, können vom Benutzer angegeben werden, siehe auch `demo_gabmul`. Für eine interaktive Nutzeroberfläche sei auf `mulaclab` hingewiesen (in Abb. 2 veranschaulicht). Solche Multiplikatoren sind für alle Arten von Frames möglich [3] und in `framemul` implementiert.

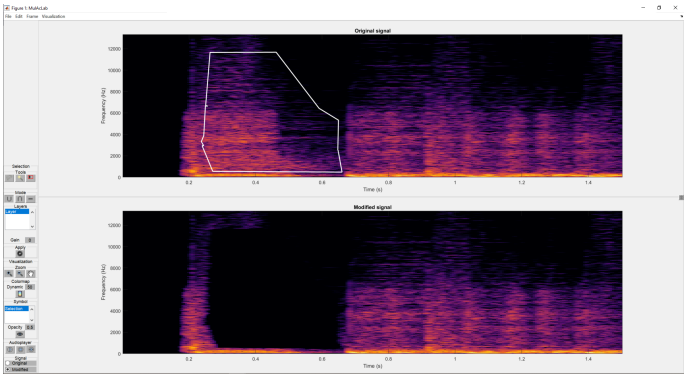

Abbildung 2: Die Abbildung zeigt das GUI von `MULACLAB`, wo versucht wird, die Stimme am Beginn eines Rock-Songs zu löschen(, der geplant war, bei der DAGA 2020 Jam-Session aufgeführt zu werden).

Anwendungen

Entrauschen oder Rauschunterdrückung ist eine der klassischen Aufgaben in der Signalverarbeitung [12]. Ein Beispiel für die Entrauschung von Audiosignalen mithilfe von Hard- und Soft-Thresholding wird in der Funktion `demo_audiodenoise` bereitgestellt.

Die Signalkomprimierung [15] zielt darauf ab, ein Signal mit minimalem Speicherplatz darzustellen, bei maximaler Wiedergabetreue. Ein Beispiel für die lineare und nichtlineare Approximation von Audiosignalen ist in der Funktion `demo_audiocompress` kodiert.

Eine Rekonstruktion, die nur die Amplitudeninformation verwendet („phase retrieval“), wird auch in LTFAT angeboten [21]. Das ist für jeden Frame möglich mittels der Methode `frsynabs`, siehe `demo_frsynabs`. Siehe Abb. 3.

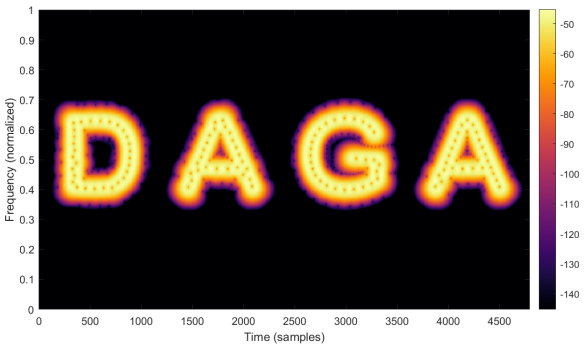

Abbildung 3: Die Zeit-Frequenz-Darstellung eines Audio-Signals, das aus dem Schriftzug „DAGA“ in der Zeit-Frequenz-Ebene mittels `frsynabs` und einem Gabor-System (wie in `demo_frsynabs` und Option `'fgriflim' [20]`) rekonstruiert wurde.

Geschichte

Als der erste Autor vor langer Zeit an seiner Dissertation arbeitete [2], die auch algorithmische Aspekte der Gabor Transformation beinhaltete, verzweifelte er oft an der Nicht-Existenz von passenden Codes von bereits existierenden Algorithmen. Diese waren entweder schwie-

rig aufzufinden, oder kaum dokumentiert. Als er dann zum ersten Mal auf Peter Søndergaard getroffen ist, der LTFAT im Rahmen seiner Dissertation [25] erfunden hat, war er heilfroh über dessen Streben, eine gut dokumentierten Zeit-Frequenz-Toolbox zu entwickeln, mit dem Plan, sie auf längere Zeit zu pflegen. Am Institut für Schallforschung (ISF) wird LTFAT, neben einigen anderen Implementierungen, gewartet und der Forschung zur Verfügung gestellt. Das ISF ist Verfechter der reproduzierbaren Forschung und offener Wissenschaft.

Danksagung

Die Arbeit an diesem Artikel wurde unterstützt durch das FWF-START-Projekt Y 551-N13 (FLAME), sowie das FWF-Projekt I 3067-N30 (MERLIN).

Mit großem Bedauern und vollem Verständis haben wir die Absage der DAGA 2019-Konferenz zur Kenntnis genommen. Unsere besten Wünsche gehen an alle, die die derzeitige Krise nicht so einfach bewältigen können wie viele Wissenschafterinnen und Wissenschater, sowie an Personen der Risikogruppe. Unser Dank gilt insbesondere auch jenen, die sich dafür einsetzen, dass das Leben trotzdem weitergeht.

Literatur

[1] J. Allen. Short term spectral analysis, synthesis, and modification by discrete Fourier transform. *IEEE Trans. Acoust., Speech, Signal Proces.*, ASSP-25(3):235–238, 1977.

[2] P. Balazs. *Regular and Irregular Gabor Multipliers with Application to Psychoacoustic Masking*. Phd thesis, University of Vienna, June 2005.

[3] P. Balazs. Basic definition and properties of Bessel multipliers. *J. Math. Anal. Appl.*, 325(1):571–585, January 2007.

[4] P. Balazs, M. Dörfler, N. Holighaus, F. Jaillet, and G. Velasco. Theory, implementation and applications of nonstationary Gabor frames. *J. Comput. Appl. Math.*, 236(6):1481–1496, 2011.

[5] P. Balazs, M. Dörfler, M. Kowalski, and B. Torrésani. Adapted and adaptive linear time-frequency representations: a synthesis point of view. *IEEE Signal Processing Magazine (special issue: Time-Frequency Analysis and Applications)*, 30(6):20–31, 2013.

[6] P. Balazs, N. Holighaus, T. Necciari, and D. Stoeva. Frame theory for signal processing in psychoacoustics. In R. Balan, J. J. Benedetto, W. Czaja, and K. Okoudjou, editors, *Excursions in Harmonic Analysis Vol. 5,*, pages 225–268. Springer, 2017.

[7] P. Balazs, B. Laback, G. Eckel, and W. A. Deutsch. Time-frequency sparsity by removing perceptually irrelevant components using a simple model of simultaneous masking. *IEEE Transactions on Audio, Speech and Language Processing*, 18(1):34–49, 2010.

[8] R. Carmona, W.-L. Hwang, and B. Torrésani. *Practical Time-Frequency Analysis*. Academic Press San Diego, 1998.

[9] O. Christensen. *An Introduction to Frames and Riesz Bases*. Birkhäuser, 2016.

[10] H. G. Feichtinger and K. Nowak. *A first survey of Gabor multipliers*, chapter 5, pages 99–128. Birkhäuser Boston, 2003.

[11] H. G. Feichtinger and T. Strohmer. *Gabor Analysis and Algorithms - Theory and Applications*. Birkhäuser Boston, 1998.

[12] C. Févotte, B. Torrésani, L. Daudet, and S. J. Godsill. Sparse linear regression with structured priors and application to denoising of musical audio. *IEEE Transactions on Audio, Speech & Language Processing*, 16(1):174–185, 2008.

[13] K. Gröchenig. *Foundations of Time-Frequency Analysis*. Birkhäuser Boston, 2001.

[14] N. Holighaus, M. Dörfler, G. A. Velasco, and T. Grill. A framework for invertible, real-time constant-Q transforms. *IEEE Trans. Audio Speech Lang. Process.*, 21(4):775 –785, 2013.

[15] N. S. Jayant and P. Noll. *Digital coding of waveforms*. Prentice-Hall, 1984.

[16] P. Majdak, P. Balazs, and B. Laback. Multiple exponential sweep method for fast measurement of head related transferfunctions. *Journal of the Audio Engineering Society*, 55(7/8):623–637, July/August 2007.

[17] G. Matz and F. Hlawatsch. *Linear Time-Frequency Filters: On-line Algorithms and Applications*, chapter 6 in 'Application in Time-Frequency Signal Processing', pages 205–271. eds. A. Papandreou-Suppappola, Boca Raton (FL): CRC Press, 2002.

[18] T. Necciari, N. Holighaus, P. Balazs, Z. Průša, P. Majdak, and O. Derrien. Audlet filter banks: A versatile analysis/synthesis framework using auditory frequency scales. *Applied Sciences*, 8(1), 2018.

[19] A. V. Oppenheim and R. Schafer. *Discrete-Time Signal Processing*. Oldenbourg, 3 edition, 1999.

[20] N. Perraudin, P. Balazs, and P. Soendergaard. A fast Griffin-Lim algorithm. In *2013 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz, NY, USA, 2013.*, 2013.

[21] Z. Průša, P. Balazs, and P. L. Søndergaard. A noniterative method for (re)construction of phase from stft magnitude. *IEEE Transactions on Audio, Speech and Language Processing*, 25(5):1154 – 1164, 2017.

[22] Z. Průša, P. L. Søndergaard, N. Holighaus, C. Wiesmeyr, and P. Balazs. The Large Time-Frequency Analysis Toolbox 2.0. In M. Aramaki, O. Derrien, R. Kronland-Martinet, and S. Ystad, editors, *Sound, Music, and Motion*, Lecture Notes in Computer Science, pages 419–442. Springer International Publishing, 2014.

[23] Z. Průša, P. L. Søndergaard, and P. Rajmic. Discrete Wavelet Transforms in the Large Time-Frequency Analysis Toolbox for Matlab/GNU Octave. *ACM Trans. Math. Softw.*, 42(4):32:1–32:23, June 2016.

[24] P. Søndergaard, B. Torrésani, and P. Balazs. The linear time frequency analysis toolbox. *Int. J. Wavelets Multi.*, 10(4):1250032, 2012.

[25] P. L. Søndergaard. *Finite Discrete Gabor Analysis*. PhD thesis, Uinversity of Denmark, 2007.

[26] J. S. Walker. *Fast Fourier Transforms*. CRC Press, 1991.

[27] B. Widrow and S. D. Stearns. *Adaptive Signal Processing*. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1985.

[28] O. Yilmaz and S. Rickard. Blind separation of speech mixtures via time-frequency masking. *IEEE Transactions on Signal Processing*, 52(7):1830–1847, July 2004.