

**SIGNAL PROCESSING AND GRAPH THEORY TECHNIQUES
FOR SOUND SOURCE SEPARATION**

Submitted in partial fulfillment of the requirements of the
Degree of Doctor of Philosophy

DELIA FANO YELA

School of Electronic Engineering and Computer Science
Centre for Digital Music
Queen Mary University of London

March 2020

Delia Fano Yela: *Signal Processing and Graph Theory Techniques for Sound Source Separation*, © March 2020

SUPERVISORS:

Mark Sandler,

Dan Stowell

EXAMINERS:

Raul Mondragon,

Toon van Waterschoot

STATEMENT OF ORIGINALITY

I, Delia Fano Yela, confirm that the research included within this thesis is my own work or that where it has been carried out in collaboration with, or supported by others, that this is duly acknowledged below and my contribution indicated. Previously published material is also acknowledged below.

I attest that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge break any UK law, infringe any third party's copyright or other Intellectual Property Right, or contain any confidential material.

I accept that the College has the right to use plagiarism detection software to check the electronic version of the thesis.

I confirm that this thesis has not been previously submitted for the award of a degree by this or any other university.

The copyright of this thesis rests with the author and no quotation from it or information derived from it may be published without the prior written consent of the author.

Date:

Signature:



Delia Fano Yela

ABSTRACT

Source separation aims to identify and separate the sources from a given mixture. In music source separation, the sources are typically musical instruments and the given mixture, a recorded track. When there is little or no prior information about the sources or recording conditions, a major goal becomes to target the inherent characteristics of the sources to help with their differentiation and separation. This thesis is concerned with methods for doing so, introducing novel approaches based on signal processing and graph theory techniques.

Kernel Additive Modelling (KAM) is a popular music source separation framework as it is flexible, computationally efficient and requires no training data. The main idea behind KAM is that one can use the inherent repetitions of musical signals to reconstruct a source by defining a proximity kernel. KAM employs robust statistics for the separation, whose success ultimately depends on the kernel ability to identify similar instances of a source in the presence of other overlaying sources. In existing KAM approaches, the kernel design is rather rudimentary and its simplicity is limiting. In this thesis we investigate the current kernel and propose novel extensions boosting its performance without losing interpretability, flexibility or efficiency.

We then explore the inherent graph structure in KAM, leading to the first unsupervised method to optimise the sole parameter in the framework. Following this perspective, we further investigate graph representations, introducing visibility graphs to magnitude spectra. We present a novel visibility graph-based representation with valuable properties for audio. Finally, we propose the first method to compute visibility graphs on-line, broadening the relevance of this thesis to generic time series analysis.

ACKNOWLEDGEMENTS

Raul, Toon, thank you for being my examiners, reading this thesis with so much care, for your excellent feedback and kind words.

Thank you Mark and Dan for supervising my PhD.

Dan, special thank you for picking me up half way, for the graphs, advice and your support. Thank you Sebastian and Derry; the reason I "research" and I "source separate".

To the CS319 crew and its whiteboards. To Adib, Keunwoo, Florian and Alo. Thank you for your research input, for teaching me so much, converting me to python, for the best advice and for your friendship. Alo, thanks for *roosa poni*, the best therapy.

Thank you Giulio for bash, git, chats, home, *moving over, taking a piece of my heart* (Keunwoo, *take it!*) and driving a van so well.

To Marco and Will, thank you for always being available, for sharing, caring and leaving me the best room.

Thank you c4dm-ers for the community, for being so bright and interested, but mainly, for being so much fun. Really. Thank you.

To Alessia, Dave and Sophie for uplifting. To An and Hai for your company and *seeing you tomorrow*. To Ken, Emmanouil and Bob for being keen.

Thank you Peter for being a *highway star* and explaining me every perceptual model I ever had interest in. I hope one day we can finish up the collaboration we started.

Thank you Enzo for collaborating with me and for making time to use the whiteboard.

Thank you June for all the freebies. To Melissa for her *love and kindness* and Edward for keeping an eye out.

Thank you Tijana for providing and promoting me with all the teaching I ever wanted to do.

Thanks to the audio community for being so welcoming, supportive and open. To Antoine and Fabian; thank you for your support, discussions, the source separation hot-line, and your company in conferences.

To my family and their 24/7 availability. Thank you for your support and love.

Finally, thank you Gijs for proof reading (a lot), for debugging, for discussing, advice, patience and for much more than I would ever know how to write.

The work in this thesis was funded by EPSRC grant EP/L019981/1.

Nolite te bastardes carborundorum.

Margaret Atwood, The Handmaid's Tale

CONTENTS

I	FRAMING THE THESIS	16
1	INTRODUCTION	17
1.1	Motivation	17
1.2	Layout	19
1.3	Contributions	19
1.4	Publications	21
2	BACKGROUND	23
2.1	What is source separation and why is it useful?	23
2.2	How to separate sources?	25
2.3	Audio representations for source separation	32
2.4	Tasks in music source separation	35
2.5	Evaluation of source separation methods	37
2.6	Summary	40
II	MEDIAN FILTERING IN SOURCE SEPARATION	42
3	KERNEL ADDITIVE MODELLING : KAM	43
3.1	The framework	43
3.2	KAM in this thesis	47
3.3	Summary	50
4	IMPROVEMENTS AND OBSERVATIONS ON KAM	52
4.1	The dark side of KAM	52
4.2	Temporal Context	55
4.2.1	Empirical evaluation	58
4.3	Shift-invariant KAM	60
4.3.1	Acceleration Extension	64
4.3.2	Empirical evaluation	69
4.4	A machine learning approach to KAM for low SNR	72
4.4.1	Empirical evaluation	79
4.5	How do we pick k?	84
4.6	Summary	86

III BEYOND THE MAGNITUDE DOMAIN : GRAPHS	87
5 EXPLOITING THE GRAPH STRUCTURE WITHIN KAM	88
5.1 Preliminaries: What is a graph?	88
5.2 Properties of the k-NN graph	92
5.3 How to pick k	95
5.3.1 Empirical evaluation	97
5.4 Summary	102
6 INTRODUCING VISIBILITY GRAPHS FOR AUDIO	104
6.1 Visibility graphs	106
6.2 Spectral Visibility graphs	108
6.2.1 Empirical evaluation	112
6.3 How to compute visibility graphs?	116
6.3.1 State of the art	117
6.3.2 Binary Search Tree Codec	118
6.3.3 Time Complexity	123
6.3.4 On-line visibility graphs: merging binary trees .	125
6.3.5 Numerical Experiments	128
6.4 Summary	133
IV TAKE HOME MESSAGE	135
7 FUTURE WORK	136
7.1 Spectral visibility graphs for KAM ?	136
8 CONCLUSION	139
V APPENDIX	143
A COMPLEXITY ANALYSIS FOR SHIFT INVARIANT KAM	144
BIBLIOGRAPHY	146

ACRONYMS AND ABBREVIATIONS

BSS Blind Source Separation

CQT Constant-Q Transform

DNN Deep Neural Networks

DSD₁₀₀ Demixing Secrets Dataset 100 songs

dB Decibel

DC Divide & Conquer

FIR Finite Impulse Response filter

FFT Fast Fourier Transform

GP Gaussian Processes

GB Gigabyte

HMM Hidden Markov Model

Hz Hertz

HV Horizontal Visibility

HVG Horizontal Visibility Graph

ICA Independent Component Analysis

KAM Kernel Addtive Modelling

KBF Kernel Backfitting algorithm

k-NN k Nearest Neighbours

MSS Music Source Separation

MIDI Musical Instrument Digital Interface

MRR Mean Reciprocal Rank

NMF Non-Negative Matrix Factorisation

Nb Number

NSDR Normalised Signal to Distortion ratio

NSIR Normalised Signal to Interference ratio

NV Natural Visibility

NVg Natural Visibility Graph

Prop. Proposed

REPET REpeating Patterns Extraction Technique

RAM Random Access Memory

SSS Sound Source Separation

SNR Signal to Noise ratio

SDR Signal to Distortion ratio

SIR Signal to Interference ratio

SAR Signal to Artifacts ratio

STFT Short time Fourier Transform

SiSEC Signal Separation Evaluation Campaign

secs Seconds

SVg Spectral Visibility Graph

TF Time-Frequency

MATHEMATICAL NOTATION

 $A(G)$ Adjacency matrix of graph G \mathbb{B} Set of binary numbers (i.e. 0, 1) c Temporal context C Maximum temporal context in time frames \mathbb{C} Set of complex numbers D_{KL} Kullback-Liebler divergence $d(v)$ Degree of node v , its value represented as κ $d^+(v)$ Out-degree of node v , its value represented as κ^+ $d^-(v)$ In-degree of node v , its value represented as κ^- d_E Euclidean distance d_C Cosine distance $E(G)$ Finite set of edges of graph G $e(G)$ Size of graph G f Frequency F Total number of frequency bins \mathcal{F} Fourier transform G Graph H Activation matrix of the training data of unwanted source n ,
such that: $\bar{X}_N \approx W_N \cdot H$

H_N Activation matrix of the unwanted source n , such that:

$$\bar{X} \approx W_N \cdot H_N + W_S \cdot H_S$$

H_S Activation matrix of target source s , such that:

$$\bar{X} \approx W_N \cdot H_N + W_S \cdot H_S$$

h Hubness, skewness of the degree distribution

h_{NULL} Hubness of null model

h_{norm} Normalised hubness

ht Height of a binary tree

ht_{\max} Maximum height of a binary tree

ht_{root} Height of the root of a binary tree

$\mathcal{I}(f, t)$ Set of k nearest neighbours of (f, t) , such that $\mathcal{I}(f, t) \in \mathbb{L}^k$

\mathcal{IF} Inverse Fourier transform

I All-ones matrix of size $F \times T$

J Total number of unknown sources

$k\text{-NN}$ k Nearest Neighbours

k Number of nearest neighbours, such that

$$\forall (f, t) \in \mathbb{L}, \quad |\mathcal{I}_j(f, t)| = k$$

K Proposed representation related to SVgs

\mathbb{L} Set of all TF bins (f, t) such that $\mathbb{L} = F \times T$

$\max(\)$ Maximum value retrieving function

$\text{median}(\)$ Median operator

n Unwanted source/interference

N STFT of unwanted source, \bar{N} its magnitude, \hat{N} its estimate

n_{HMM} Unwanted source state vector: 1 if active, 0 if not.

$n(G)$ Order of graph G

n_κ	Number of nodes with degree κ
P	Pruning parameter indicating the number of additional nearest neighbours
$P(\kappa)$	Degree distribution
\vec{p}	Degree distribution vector of a graph
q_f	Quefrency
\mathbb{R}	Set of real numbers
\mathbb{R}_+	Set of positive real numbers
R_1, R_2	Rank values parameters of NMF
s	Target source
S	STFT of target source, \bar{S} its magnitude, \hat{S} its estimate
S_q	CQT of target source, \bar{S}_q its magnitude, \hat{S}_q its estimate
\hat{S}_{NMF}	NMF Source estimate such that $\hat{S}_{NMF} := W_S \cdot H_S$
t	Time, time frame
T	Total number of time frames
T_{HMM}	Threshold symbolising the parameters of a HMM
$V(G)$	Finite set of vertices, or nodes, of graph G
v	Node or vertix
W	Mask used for separation, e.g. $\hat{S} = W \odot X$
W_N	Basis matrix for training data of unwanted source n , such that: $\bar{X}_N \approx W_N \cdot H$
W_S	Basis matrix of target source s , such that: $\bar{X} \approx W_N \cdot H_N + W_S \cdot H_S$
x	Observed mixture of sources $x = \sum_{j=1}^J s_j$, here $x = s + n$

X	STFT of mixture, \bar{X} its magnitude
X_q	CQT of mixture, \bar{X}_q its magnitude
X_s	Specmurt of mixture, \bar{X}_s its magnitude and \bar{X}'_s its reduced version
y	Time series such that $y = g(t)$
Y	STFT of time series y , \bar{Y} its magnitud.
z	Spectrum, \bar{z} its magnitude
α	Quefrency broadband cut-off point
β, λ	Auxiliary variables and parameters
δ	Shift parameter in the kernel measured in frequency bins
Δ	Absolute maximum frequency shift
κ	Value of the degree of a node $d(v)$
$\vec{\kappa}$	Degree vector of a graph
Γ	Number of channels in a mixture x

Part I

FRAMING THE THESIS

This first part presents the motivation and layout of this thesis, as well as the published work relevant to the dissertation. A brief background on the field and fundamental concepts of source separation will also be outlined.

INTRODUCTION

In recent decades, digital music signal processing has revolutionised the way we create, consume and produce music. The digital era opened a door for music analysis and processing that was unimaginable before, introducing new exciting ways of understanding and working with musical sounds. A new field of research emerged; how can we make machines understand music as we do?

Now we are able to automatically generate a playlist tailored to your music taste, to use a mobile phone to listen and recognise songs, to automatically mix a multitrack recording, to up-mix old records into surround sound, to automatically transcribe music and much more. None of these applications would be possible without the extensive research on the large variety of digital music signal processing core tasks.

Sound source separation is one of the core tasks in many audio applications [19, 92, 109] such as de-noising or up-mixing [42, 137]. In the common blind source separation scenario, the goal is to extract a source from a given mixture of sources, with little or no information about the sources. In this thesis we will explore and expand some existing methods for music source separation and introduce a new perspective through graph theory tools.

1.1 MOTIVATION

The current state-of-the-art in music source separation methods employ machine learning techniques, typically variants of Non-Negative Matrix Factorisation (NMF) [76] and in recent years, based on Deep Neural Networks (DNNs) [127]. DNNs have both objectively and

audibly drastically improved separation results, shadowing model-based approaches unable to compete in separation performance [125]. Even though one could question the flexibility of these approaches, as they heavily rely on the quality and diversity of their training data, they are beyond doubt raising the standard for source separation applications regardless of the major interpretability cost from such a black-box approach. Led by impressive results, the sound source separation community is converging towards DNN-based solutions, accompanied by a change in paradigm.

Despite this overwhelming trend, modelling continues to play a key role in the community as it essentially brings the interpretability which DNN approaches lack. Model-based methods provide us with valuable insights on the behaviour and interaction of musical signals, widening the creative space of reflection, mother to future ideas. Here we argue that modelling and machine learning respond to different, yet all relevant, questions towards a shared goal; in short DNNs care about performance, and models about behaviour knowledge. Therefore, these two approaches are not exclusive, but in fact, complementary.

In this thesis we will focus on interpretable model-based methods for source separation and on their computational efficiency. The majority of the proposed methods in this thesis are computationally inexpensive and could be implemented to run real-time. Further, we provide alternatives for those methods which are heavier to compute. The low computational cost of the methods presented brings an added advantage of low requirements. None of the methods presented in this thesis require particular machine specifications nor large data-sets, and are therefore accessible to a wide range of researchers. In addition, source code is freely available online¹. We appoint efficiency and accessibility as the driving forces of this thesis.

¹ Source code available at <https://github.com/delialia>

1.2 LAYOUT

The thesis is divided into four parts. This first one aims to introduce the reader into the field of music source separation with a literature review in Chapter 2 giving enough grounding and pointers for the rest of the thesis.

The second part is mainly concerned with the so-called Kernel Additive Modelling (KAM) framework for music source separation which is fully defined in Chapter 3. In Chapter 4 we explore KAM's limitations and propose different extensions under problematic scenarios were KAM is likely to fail.

In the third part of the thesis we shift in perspective by introducing graph theory concepts to our framework in Chapter 5. We then take a step further and explore the potential of graph representation for audio applications by introducing visibility graphs to spectra in Chapter 6. In addition, we present the first algorithm capable of computing visibility graphs on-line while maintaining the state-of-the-art efficiency, relevant to any time series and so, widening the outreach of this thesis.

Finally we briefly outline in Chapter 7 some of the possible research lines derived from this thesis and conclude in Chapter 8 with the key aspects of this dissertation.

1.3 CONTRIBUTIONS

The contributions of this thesis expand and improve kernel additive modelling for music source separation [82]. We focus on the popular kernel choice of k nearest neighbours (k-NN) commonly applied to vocal separation tasks [41, 108]. We expand its use, for the first time, to interference reduction applications, where the music signal is overlaid by a transient burst-like unwanted sound, like a cough in a live recording.

The popular k-NN kernel relies on the following strong assumptions:

- The target source is energetically dominant
- The target source repeats in time and frequency

Such assumptions are often violated in music signals and so, in this thesis we address such limitations by:

- introducing a temporal context in the kernel, taking some musical structure into account
- differentiating the kernel search space from the processing space
- proposing a shift-invariant kernel capable of identifying similar spectral content even under frequency shifts
- integrating machine learning in the framework to overcome low signal-to-interference ratio by:
 - locating the interference activation in the recording
 - incorporating an initial estimate of the clean music signal as a search space

In addition we address the influence and optimisation of the sole parameter of the framework for the first time by:

- exploiting and defining the graph structure in KAM
- proposing a method to automatically optimise k in a vocal separation task
- discussing the influence and importance of such parameter

We then further explore the translation to graph domain of music signals, offering a novel perspective by:

- introducing visibility graphs to audio spectra
- proposing a novel graph-based representation for audio analysis: the spectral visibility graph degree

- demonstrating its utility to measure robust similarity between harmonic signals

In addition, we propose a novel method to compute visibility graphs efficiently that is, for the first time, capable of assimilating incoming data on-line by:

- using an encoder/decoder approach, defining :
 - an on-line adjustable binary search tree encoder for time series
 - a corresponding decoder for visibility graphs

Our proposed method for computation of visibility graphs offers an on-line computation solution at no additional computation time cost, and allows to employ visibility graphs in the analysis of large-scale time series and for the on-line assimilation of new data.

1.4 PUBLICATIONS

Most of the main contributions of this thesis have been peer-reviewed and published. Therefore, some of the ideas and figures in this thesis are also discussed in published work, as follows:

- In Chapter 4: Section 4.2 [4], Section 4.3 [5] and Section 4.4 [3].
- In Chapter 5 : Section 5.3 [1].
- In Chapter 6 : Section 7.1 [2] and Section 6.3 [6].

- [1] Delia Fano Yela, Dan Stowell, and Mark Sandler. "Does k Matter? k-NN Hubness Analysis for Kernel Additive Modelling Vocal Separation". In: *International Conference on Latent Variable Analysis and Signal Separation*. Springer. 2018, pp. 280–289. URL: https://link.springer.com/chapter/10.1007/978-3-319-93764-9_27.
- [2] Delia Fano Yela, Dan Stowell, and Mark Sandler. "Spectral Visibility Graphs: Application to Similarity of Harmonic Signals".

- In: *Proceedings of the European Signal Processing Conference (EUSIPCO)*. 2019. URL: <https://arxiv.org/pdf/1903.01976.pdf>.
- [3] Delia Fano Yela, Sebastian Ewert, Derry Fitzgerald, and Mark B. Sandler. "Interference Reduction in Music Recordings Combining Kernel Additive Modelling and Non-Negative Matrix Factorization". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. New Orleans, USA, 2017, pp. 51–55. URL: <https://ieeexplore.ieee.org/document/7952116>.
 - [4] Delia Fano Yela, Sebastian Ewert, Derry Fitzgerald, and Mark Sandler. "On the Importance of Temporal Context in Proximity Kernels: A Vocal Separation Case Study". In: *Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio*. 2017. URL: <http://www.aes.org/e-lib/browse.cfm?elib=18752>.
 - [5] Delia Fano Yela, Sebastian Ewert, Ken O'Hanlon, and Mark B. Sandler. "Shift-Invariant Kernel Additive Modelling for Audio Source Separation". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. Calgary, AB, 2018, pp. 616–620. URL: <https://ieeexplore.ieee.org/document/8461801>.
 - [6] Delia Fano Yela, Florian Thalmann, Vincenzo Nicosia, Dan Stowell, and Mark Sandler. "Online visibility graphs: Encoding visibility in a binary search tree". In: *Physical Review Research (forthcoming)* (2020).

2

BACKGROUND

In this chapter the field of source separation is introduced, alongside the main concepts and standards in the community. It is to serve as an introduction for the newcomers and as a pointer for those with curiosity. For ease of reading, the mathematical notations and definitions of the relevant state-of-the-art will be described in detail later in the dissertation suiting the scientific content.

2.1 WHAT IS SOURCE SEPARATION AND WHY IS IT USEFUL?

Source separation is the discipline aiming to extract individual sources from a given time series mixture of different sources. Being able to estimate the contribution of a target source at a certain time has proven to be decisive in numerous fields including medicine, finance, telecommunications and engineering [21]. The nature of the sources and mixture will vary depending on which field the problem is defined. For example, source separation techniques are used to monitor the vibrations in rotors to control and prevent fault functioning, like rotating out of axis, that could have deep consequences [62]. Source separation has also been widely used as a denoising tool, notably to remove electroencephalographic artifacts [63, 130] or to enhance speech signals [107, 133], as well as an analysis tool in, for example, natural language processing applications [138].

In *sound source separation* (SSS), the sources are sound objects (i.e. anything producing a sound, from musical instruments to environmental sounds) and the mixture is typically a recording of such sound objects. As a peek of the diversity of applications within the vast SSS field, some have found its utility in the analysis of environmental and animal sounds, popular in the emerging field of acoustic scenes and

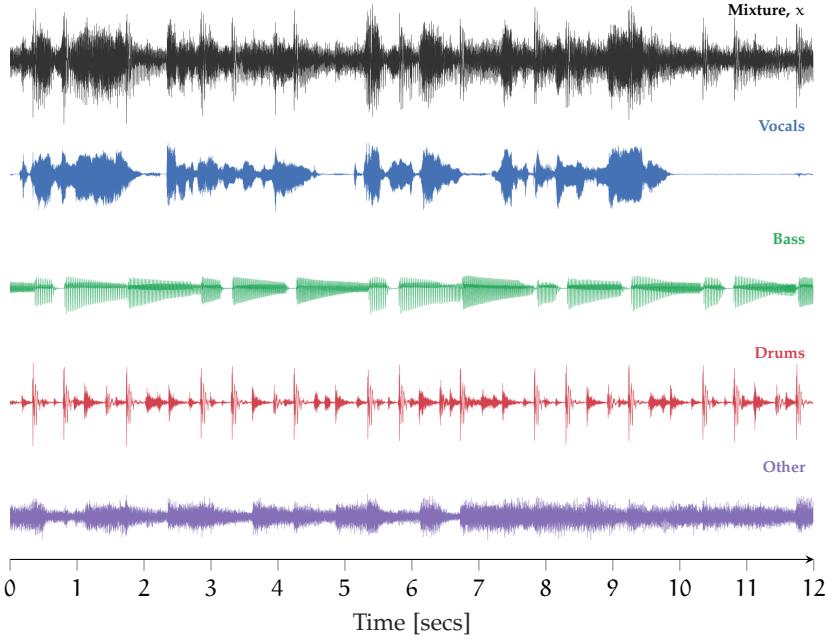


Figure 1: Illustration of a mixture of sources and its source components waveforms in the time domain

events detection and classification [128], as well as in bioacoustic tasks such as separating chimpanzee’s drummings from vocalizations [56]. However, speech applications have been the driving force of SSS research in the last decades, leading telecommunications technologies [51, 133].

Even though much of the sound source separation research has indeed been driven by speech related tasks, the interest in *music source separation* (MSS) has considerably increased in the last decade taking the music industry to a new level of applications possibilities in which the different sound objects can be manipulated individually. Having the ability to remove an instrument from a musical recording has, for example, helped students to practice in the context of an ensemble by playing along the remaining instrument of the recording [16]. Further, MSS has also found successful applications in automatic music transcription systems as transcribing instruments in isolation benefits the overall performance [7, 99] in the same way it benefits instrument classification tasks [6]. MSS has also been recently employed to assist automatic mixing, as it allows to adjust individual instruments [91] and found a real-world application in the upmixing of mono recordings to stereo [42].

In music source separation, the sound sources are the ones typically found on a music recording set-up, such as musical instruments or audience noise. Figure 1 illustrates a given time series mixture and the different sound components in it. Within this thesis, the sources are taken to be in a specific sound field (i.e. acoustic conditions do not change over time) and only the measurement of the effect of the ensemble will be observable. Such scenario, where there is no information about the individual sources (such as their placement or number) other than their joint effect, is known as a *blind source separation* (BSS) problem.

2.2 HOW TO SEPARATE SOURCES?

The diverse methods to obtain the estimate of the isolated unknown sources depend on different factors, such as the assumptions on the sources (statistically independent, uncorrelated temporally ...), the measurement conditions (number of microphones, spacing between them ...), the processing domain (time, frequency ...) or the number of measurements available compared to the number of sources (determined, overdetermined or underdetermined problem). This thesis will focus on the *underdetermined* scenario in which the number of observable mixtures is less than the number of sources.

Unlike the typical speech enhancement scenario where strong assumptions can safely be made on the observable mixture, such as the independence of the source signals, musical sources are highly correlated and often non-linearly mixed [21]. Therefore, in a music source separation scenario, most of the assumptions of the well established speech separation methods such as independent component analysis (ICA) [60] are violated and hence not suitable for music signals [103]. In response, multiple music specific source separation algorithms have been proposed in recent decades [19].

Following the description in [19] illustrated in Figure 2, a typical music source separation (MSS) workflow starts by transforming the input mixture signal x to the time-frequency (TF) domain. Most of MSS research has focused on the short-time Fourier transform (STFT),

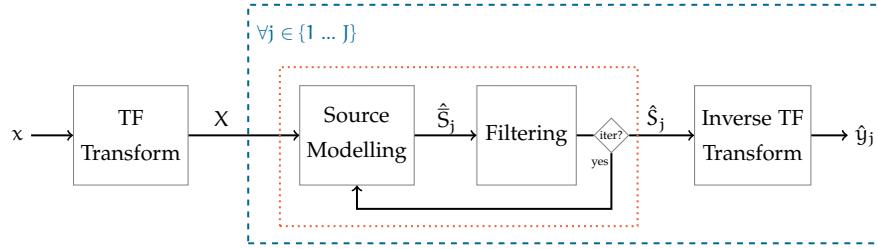


Figure 2: Illustration of the common music source separation workflow as described in [19].

$X \in \mathbb{C}^{F \times T}$, where T is the total number of time frames and F the number of frequency bins. It is common to discard the phase information by employing the magnitude spectrogram $\bar{X} \in \mathbb{R}_+^{F \times T}$, as it will be further discussed in section 2.3.

In the next step referred to as *source modelling* in [19], the TF representation is used to model the individual sources in the mixture by either estimating a model of their spectrograms or of their locations in the sound field. Given the input mixture x constituted by J musical sources s_j with $j \in 1 \dots J$, the estimates of the magnitude spectrogram of the sources $\hat{S}_j \in \mathbb{R}_+^{F \times T}$ are obtained at this stage.

Such estimates are then used in the *filtering* stage to retrieve the separated music source signals, usually, through soft-masking [19]. For every source j , the complex mixture spectrogram X is weighted by a mask. Commonly a generalised Wiener filter [30] recovers the source $\hat{s}_j \in \mathbb{C}^{F \times T}$ through the following element-wise multiplication:

$$\hat{s}_j = \frac{\hat{S}_j}{\sum_{i=1}^J \hat{S}_i} \odot X \quad \forall j \in \{1 \dots J\} \quad (1)$$

when the mixture is assumed to be a linear sum of independent sources.

To improve the source estimates one can iterate over the source modelling and filtering stages, using the output source estimates as input "mixtures" of the next iteration [19, 82, 84]. Ultimately, the time domain source estimate signals \hat{s}_j are obtained through the inverse TF transform, usually the inverse short-time Fourier transform. The TF transformation and filtering stages are often similar amongst MSS methods, and is the source modelling stage that differs.

In order to isolate the musical sources with little or no information about them or the sonic environment involved in the recording, some methods make use of the multichannel information such as difference in phase or panning of sources [46, 77, 96, 101]. In this case the mixture x can be described as a set of Γ time series where Γ is the total number of channels (e.g. $\Gamma = 2$ for stereo) and so the STFT of the input mixture can be viewed as a tensor $X \in \mathbb{C}^{F \times T \times \Gamma}$. However, when only one channel is actually available, one can no longer exploit spatial information. In addition, such information is often unreliable due to the use of various non-linear sound effects yielding artificial sound scenes which cannot physically be reproduced and are difficult to model [109]. Therefore, this thesis focuses on *single-channel source separation* methods.

In this context, the goal becomes to find characteristics helping with the definition, identification and separation of the individual sources, which can be either modelled explicitly or learned from data. *Model based* approaches exploit characteristics of the sources for their separation by explicitly including them in the model. For example, music signals are likely to follow a melody, so some methods include the score information (given or issued from a transcription stage) in their pipeline to inform the separation [35, 36, 49, 57]. Following the same logic, one could assume the musical source of interest to be harmonic and then use a pitch tracker to aid with the separation. Some have proposed to use sinusoidal modelling to implement an analysis-synthesis approach where the source of interest is estimated and synthesised using the pitch information from the initial transcription stage [117, 118]. Others propose to take a comb filter approach by expanding the pitch information with an a priori harmonic expectation of the target source [15].

It is not uncommon to involve the user in the process to gain knowledge of the target source. For example, some have involved the user as a precise way to select relevant information for the algorithm, such as where is the target source present [44] or which is the fundamental frequency of the target source [31]. Other approaches rely on the user to hum [122] or play [40] the target source to assist with the separation. Even though these approaches have successfully included in-

formation about the sources in the form of prior knowledge, they all depend on a preliminary step which does not guarantee an improvement in the separation performance as they often rely on the user knowledge of the proposed representation [120] or on assumptions that are not always met (e.g. pitch trackers often assume the source of interest to be the loudest harmonic sound in the mixture).

As an alternative, other model-based approaches target directly the inherent characteristics of the source of interest. Such characteristics can include various acoustical or perceptual aspects, including the typical behaviour of a source in time such as vibrato [28, 29], continuity in activity [8, 134], repetitiveness of patterns [41, 108], or spectral characteristics such as broadband vs harmonic energy distribution [24, 39, 45]. Such approaches are quite popular in the field as they tend to be quite inexpensive in terms of both computation and knowledge required. This methodology ease comes at performance cost as the models of these methods usually rely on core assumptions about the sources that could be violated by the signal under study.

On the other hand, *machine learning* methods aim to avoid such limitation by simply not making any strong assumptions on the sources. Instead, a large and representative database of examples is needed to "learn" the model and its vast number of parameters. Currently, most state-of-the-art methods lie within this context and are based on either Non-Negative Matrix Factorisation (NMF) [100, 113, 123] or Deep Neural Networks (DNN) [52, 58].

Based on generalised Wiener filtering, NMF aims to model a time-frequency representation of the mixture $\bar{X} \in \mathbb{R}_+^{F \times T}$ as a product of two non-negative matrices $W \in \mathbb{R}_+^{F \times R}$ and $H \in \mathbb{R}_+^{R \times T}$, such that $\bar{X} \approx WH$, where *rank R* is the parameter of the method. The goal is to minimise the error of reconstruction defined by a cost function of choice (e.g. Frobenius norm $\|X - WH\|_F$) [75, 76].

The basic idea in the context of MSS is that, after convergence, the first matrix W should encode the essence of the different sources and the second matrix H their activation in the mixture. It is so that, the columns of W are often interpreted as "templates" capturing the spectral properties of the individual sound sources in the signal; the rows

of H are often referred to as the corresponding "activations", encoding *when* and *how* strong each template is active in the input signal [121].

In a complex musical scenario, the sources are hard to model with a fixed number of templates and will often share spectral properties, and thus applying the original NMF approach [76] was found to rarely yield useful results [43]. Therefore, various extensions were proposed integrating various constraints on the parameter estimation process. Examples include sparsity and temporal continuity constraints [134] or harmonicity constraints [8]. Further, various types of side information have been used, such as user-assisted annotations [120] and musical score information [36].

Alternatively, one of the most widely used and successful approaches is to employ training data. Supervised and semi-supervised NMF methods "learn" the spectral templates of the target source and use those to determine the activation of such in the mixture [20]. This way, one can avoid relying on specific assumptions about the statistical independence of the sources [1]. As a major drawback of this approach, however, the quality of the separation result heavily depends on the assumption that the acoustical conditions in the training material and in the recording to be processed are similar. The more this assumption is violated, the more artefacts are to be expected.

To overcome NMF drawbacks, recent methods propose DNNs as means to learn the relation between the time-frequency representation of the mixture and the source of interest [52, 53, 78]. In short, DNNs can be understood as a combination of non-linear transformations learnt from the dataset of examples reserved for training. DNNs are typically trained on the magnitude spectrogram of the mixture to predict either a time-frequency mask describing the energy distribution of a source relative to the other sources [95] or the source spectrogram directly [58, 96]. Most of the methods differ either in the network architecture or its training fashion.

Even though machine learning methods have different trade-offs with respect to run-time, separation quality and adaptability to new acoustic conditions, DNNs have, beyond doubt, remarkably improved

separation performance, becoming the new standard in the field [125, 127]. Such a success has attracted multitude of researchers displaying an overwhelming increase of interest in the last years (comparison between [2] and [125]). However in order to achieve outstanding performance improvements, learning-based approaches must be used in settings where large amounts of training material are available: otherwise their flexibility and adaptability could be questioned as their methods are typically trained for specific combinations of instruments or instruments groups [96]. Therefore it comes as no surprise that such approaches have been often promoted by large corporations with access to the large audio datasets required for a flexible, adaptable and successful training. The best performing algorithm of the latest source separation evaluation campaign [125] is DNN based and used an additional extensive private dataset [52]. However, the recently released "reference implementation for music source separation" *Open-Unmix* [127] is trained on the publicly available MUSDB18 dataset of only 100 songs and claims to match such state-of-the-art; maybe opening the door to the new standard in music source separation as the authors did release all their pre-trained model and source code with supporting material as open software for the benefit of the community. Nonetheless, the multitude of adjustable parameters is often viewed as an unavoidable shortcoming leading to a daunting tuning task reinforcing the lack of interpretability of DNN-based systems.

Consequently, despite a measurable difference in performance, interest in "learning-free" methods remains high. A focus on explicitly modelling concepts increases the interpretability of methods, which opens more angles for including prior knowledge, which might help with understanding how machine learning methods operate and can lead to a high generalisation capacity across datasets. Therefore, instead of training a model, one could just target the inherent properties of the sources directly by taking advantage of their differences in the separation process. In this context, a number of methods exploit these differences by relying on a similarity measure that will accentuate the target source in the mixture [39, 41, 106, 108]. Such methods are based on median filtering and can be considered instances of the general framework known as *Kernel Additive Modelling* (KAM) [82].

KAM is a flexible time-frequency domain framework able to separate sound sources at a low computational cost. The main idea behind KAM relies on the assumed repetition of sound events in musical signals, by defining a proximity kernel function which detects these repetitions and so identifies similar spectral bins for the sources we want to keep while ignoring the energy associated with other sources. Since some of the kernel bins might be overlaid by other sounds as well, or are not exact repetitions, KAM employs order statistics to identify the commonalities between the bins while neglecting the outliers. In other words, KAM reconstructs the magnitude for a given source by analysing the values at the locations where the target source is likely to assume similar values.

From a modelling point of view, the core idea of KAM is related to the more widely known Gaussian Processes (GP) [80]. In both cases, one assumes that individual entries correlate with others in a known way, and so if we can observe the value of one entry, we can make a statement about the value of related entries. This means that for many signals we can estimate the value of a single sample by looking at the value of neighbouring (or similar) samples. For example, similarly to a low-pass FIR filter [92], we can take the average of the values of neighbouring samples to reconstruct a low frequency signal corrupted by white noise. KAM and GP take a step further by enabling much more general notions of similarity or neighbourhood.

KAM differs from GP in several aspects. The GP framework is formulated as an inference problem, where usually a covariance matrix defined by a kernel function is updated based on the observed data. The relationship between random variables, as well as the noise, must be Gaussian and the kernel function is restricted to lead to positive semi-definite covariance matrices. The inference process involves the inversion of such matrix and renders the method computationally expensive, yet highly adaptable and expressive. In KAM, inference does not exist in such form as the kernel depends on the actual observations themselves, introducing the use of outlier resistant methods from robust statistics which allows for non-Gaussian relationships and modelling of non-Gaussian noise [82]. In addition, such kernel,

while limiting its expressiveness, drastically improves the computational performance.

The high interpretability and flexibility of KAM explains its increasing popularity in the field, offering a quick blind source separation approach to a multitude of applications [39, 82, 107, 108]. However, in existing KAM approaches, the proximity kernel design is often rather rudimentary, offering a natural starting point for this thesis, which starts by investigating different kernel designs under problematic scenarios where KAM is likely to fail in Chapter 4. KAM framework will therefore be introduced in more detail in the following Chapter 3.

2.3 AUDIO REPRESENTATIONS FOR SOURCE SEPARATION

Most methods for audio source separation use a time-frequency (TF) representation (2D) of the given time domain waveform (1D), illustrating the spectrum of frequencies over time. In the TF domain, the individual characteristics of the sources become more apparent as there is less overlap between sources than in its associated time domain waveform, and so better suited for a source separation task (compare Figure 1 with Figure 3).

The short-time Fourier transform (STFT), represented in Figure 3, is the most commonly used transformation to the TF domain because it is efficient and because the resulting complex spectrogram is linear and invertible. This means that a sum of sources in the time domain (t) corresponds to a sum of the STFT of the sources, and that any modification in such domain will modify the time domain waveform after inversion. Therefore, given a linear mixture $x(t)$ of sources $s_j(t)$, such that

$$x(t) = \sum_{j=1}^J s_j(t) \quad (2)$$

their corresponding STFTs X and $S_j \in \mathbb{C}^{F \times T}$ also hold such relation

$$X = \sum_{j=1}^J S_j. \quad (3)$$

In practice, most audio source separation techniques actually operate on the magnitude spectrogram and assume the phase of the sources to be equal to that of the mixture and disregard the phase information in the estimation process (with some exceptions [17]). The ease of operating on the real versus the complex domain comes, however, at an audible cost. Therefore, research for alternatives on how to reconstruct or work with the phase of a STFT spectrogram has been developed.

The reference approach to reconstruct the phase from a modified magnitude STFT spectrogram is the Griffin-Lim algorithm [54]. This iterative approach aiming to find the closest consistent STFT spectrogram (that of a real signal) to a given magnitude spectrogram is however slow and thus a faster alternative was proposed [72, 73]. Instead of using consistency, others propose to use frequency and time coherence of partials to reconstruct the phase [86] or to work on the complex domain directly [23] or, of late, thanks to the recent advances in deep learning, to avoid TF representations entirely by staying in the time domain with end-to-end architectures [124].

Despite these short-comings, the magnitude STFT spectrogram remains the usual TF representation of choice in music source separation methods [92]. Further, in a linear mixing problem (i.e. the time waveforms of the sources add to the given mixture waveform expressed in Equation (2.3)), it is common to assume that the magnitude STFT spectrograms of the sources add up to the given magnitude STFT spectrogram of the mixture [19], such that

$$\bar{X} \approx \sum_{j=1}^J \bar{S}_j. \quad (4)$$

Even though this is not strictly the case, as it is for the complex spectrograms, the ease this assumption introduces to model-based approaches is often worth the degradation of audio quality it generates.

Alternative TF representations have been presented with a logarithmic frequency resolution, notably the constant Q transform (CQT) [12], represented in Figure 4. This representation is advantageous for music applications as a log-frequency spectrogram can be set to

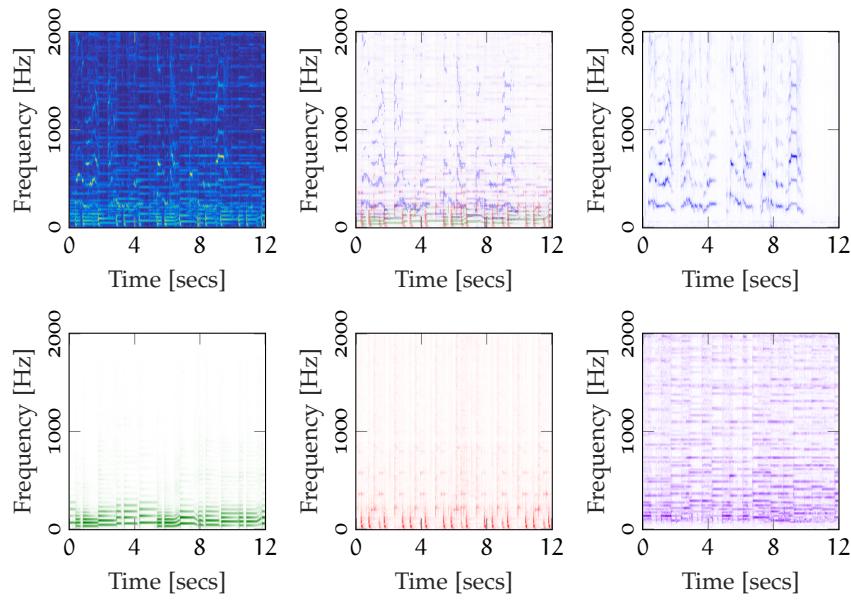


Figure 3: Magnitude short-time Fourier transform (STFT) of a given mixture and a representation of each source contribution color coded as in Figure 1.

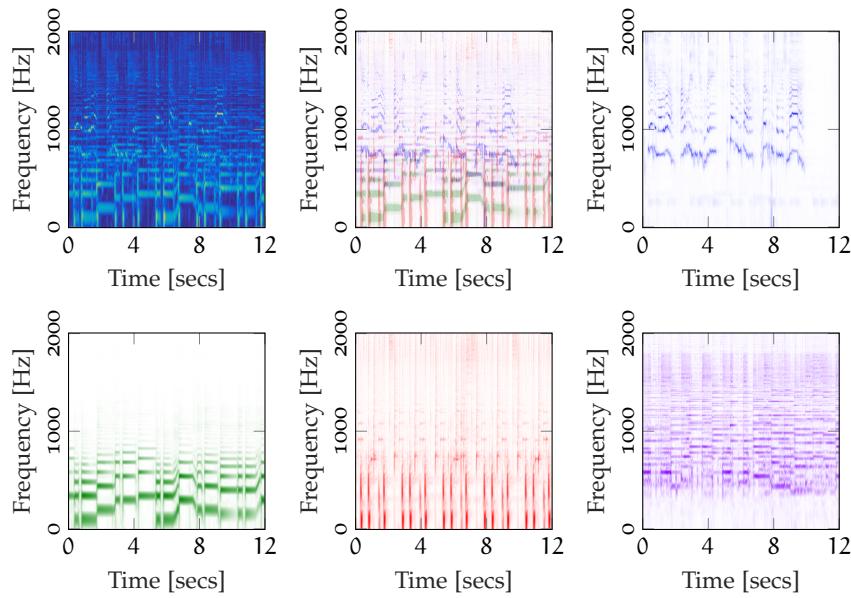


Figure 4: Magnitude constant Q transform (CQT) of a given mixture and a representation of each source contribution color coded as in Figure 1.

match the equal tempered western music scale, where the ratio between adjacent note's frequency is constant.

In the STFT, the frequency distance and resolution is constant and dictated by setting a fixed window size. In the CQT, the length of the window varies with frequency, as what is now fixed is the ratio between the centre frequency to bandwidth, known as Q . In other words, in a CQT the ratio between the centre frequency and resolution is constant, whereas in the STFT the resolution is constant.

Unfortunately, the CQT is computationally expensive, in particular where the initial minimum frequency is low as it requires a large window size, and it was at first not invertible [22], although recent advances have shown otherwise [114]. It remains, however, a less popular TF representation choice compared to the STFT.

Even though in TF representations the difference between sources is more apparent than in the time domain signal, the two dimensional discrete Fourier transform (2D-DFT) domain has been found more appropriate to model non-stationary sound objects (e.g. those exhibiting vibrato) [126] and to distinguish periodic from non-periodic patterns applied to singing voice extraction [115]. In particular, the 2D-DFT gives an excellent alternative representation for unison mixtures where the sources clearly overlap in the standard STFT representation [102, 126].

2.4 TASKS IN MUSIC SOURCE SEPARATION

Music source separation algorithms aim to isolate a target source from the mixture. However, due to the challenging nature of the problem, what defines the target source is remarkably relaxed. Instead of targeting the entire sound produced by a particular object, like a flute or drum, music source separation algorithms target the most prominent characteristics of those sound objects, like its harmonic or percussive nature. Musical sound objects are generally divided into three categories: harmonic, percussive or vocal; culminating in two

main separation tasks: harmonic/percussive separation and vocal or lead/accompaniment separation.

Harmonic sources present a stationary behaviour over time, displaying horizontal lines in the TF domain (see bass in green in Figures 3 and 4). Such harmonic sound is mainly composed by a fundamental frequency (usually assumed to be the one with greater power) and its harmonic components whose frequencies are multiples of the fundamental. Further, the harmonics and fundamental frequencies are said to have "common fate" as their trajectories in time are often aligned [11]. Such phenomenon has been exploited in the past to aid with the separation of unison mixtures [102, 126].

On the contrary, percussive sources are considered to be transient broadband events appearing as vertical ridges in the TF domain (see drums in green in Figures 3 and 4). Percussive music sources are also assumed to be repetitive and even periodic as they usually mark the beat in the given song mixture [39].

In the same way, the accompaniment music tends to be more repetitive than the lead in a standard popular song, and so some methods have exploited such difference to extract vocals from the mixture [41, 106, 108]. Vocals are considered to be a particular non-stationary and sparse harmonic source (see vocals in blue in Figure 3 and 4). Such characteristics are hard to model and so most methods target the background music and treat the vocals as residual.

In this thesis a novel music source separation task is presented, including the audience noise interferences as a category of sound objects found in a live musical recording: interference reduction. Here, the interferences are taken to be transient loud burst-like sound events, such as coughs or door slams. Unlike the percussive sources, these do not tend to repeat in time and less so in a periodic manner. In addition, such events often overpower other sound sources, masking the musical content in most frequency bands for a short period of time. Their impact and removal will be further discussed in Sections 4.3.2 and 4.4.1.

In practice, the sound sources in a music recording will most certainly share characteristics amongst them. For example, a clarinet and a flute are both harmonic instruments. Therefore, one target source is often estimated as an ensemble of akin sound sources. What is more, individual sound sources often present aspects corresponding in different categories. For example, both the guitar and piano have a clear percussive attack which is often dismissed if the source is considered fully harmonic. Some methods have proposed to overcome this using a cascade approach where the mixture is separated into harmonic, percussive and residue components iteratively. The sources are then estimated recombining the different aspects of them [28, 84].

2.5 EVALUATION OF SOURCE SEPARATION METHODS

Most methods in music source separation are expected to result in musically pleasing signals and what ultimately defines "musically pleasing" relies on human perception. Therefore the most rigorous practice to evaluate source separation methods is to perform listening tests where the separation quality is judged by humans. However, this is deeply time consuming and finding enough relevant participants is not a trivial task, leaving space for objective metrics to give us an indication on the success of the separation.

The currently widely adopted measures to quantify the quality of separation, given the clean target s_j (i.e. ground truth) and estimated sources \hat{s}_j , are the ones from the blind source separation evaluation (BSS Eval) Matlab toolkit, notably the signal-to-distortion ratio (SDR), signal-to-interference ratio (SIR) and signal-to-artifacts ratio (SAR) [132]. These metrics essentially quantify distortions between the target and estimated source signals.

More precisely, the estimated signal \hat{s}_j is decomposed into a true source part $s_{j\text{target}}$ (i.e. target source) and some error terms corresponding to the bleeding of other unwanted sources $e_{j\text{interf}}$, plus additive noise $e_{j\text{noise}}$ and algorithm artifacts $e_{j\text{artif}}$ [132], such that

$$\hat{s}_j = s_{j\text{target}} + e_{j\text{interf}} + e_{j\text{noise}} + e_{j\text{artif}}. \quad (5)$$

The SDR represents the ratio of the true source and all the error terms,

$$\text{SDR}_j := 10 \log_{10} \frac{\|s_{j_{\text{target}}}\|^2}{\|e_{j_{\text{interf}}} + e_{j_{\text{noise}}} + e_{j_{\text{artif}}}\|^2} \quad (6)$$

the SIR the ratio between the true source and bleeding,

$$\text{SIR}_j := 10 \log_{10} \frac{\|s_{j_{\text{target}}}\|^2}{\|e_{j_{\text{interf}}}\|^2} \quad (7)$$

and the SAR the ratio between the target source with additive noise and bleeding errors and the error of the algorithm's artifacts

$$\text{SAR}_j := 10 \log_{10} \frac{\|s_{j_{\text{target}}} + e_{j_{\text{interf}}} + e_{j_{\text{noise}}}\|^2}{\|e_{j_{\text{artif}}}\|^2}. \quad (8)$$

Since the SDR ultimately measures the overall error energy contribution in the estimate, it is common to use it as a sole quantitative indicator of the separation performance [125] and it will be used for that purpose throughout this thesis.

None of the metrics above are consistently bounded as their values depend on the maximum energy of the estimated source signal, and so, higher values imply better separation. Therefore, oracle performance in the form of ideal binary mask or ideal ratio mask are often included in the evaluation as a best-case scenario to help with the interpretation of the results [125].

Another approach to put the BSS Eval results into perspective is to compute the worst case scenario by using the unprocessed mixture as the source estimate (i.e. $\hat{s}_j = x$). In this way, one can compute the gain in separation performance compared to not doing anything, resulting in the normalised NSDR, NSIR and NSAR metrics. Therefore, given the SDR_j of the source estimate and that of the mixture itself SDR_x , we can define the normalised SDR metric measured in dBs as

$$\text{NSDR}_j := \text{SDR}_j - \text{SDR}_x. \quad (9)$$

The NSIR and NSAR are defined following the same logic.

Alternatively, source separation methods can be compared to each other. Such an approach would not be straight forward would the source code of state-of-the-art methods not be available: fortunately,

with the increase of open source research one can often compare the performances of methods [90]. The *signal separation evaluation campaign* (SiSEC) [125] has played a major role in making fair comparisons between methods possible. SiSEC provides a platform in the form of open-source software, where source separation methods can be plugged-in, that will automatically load, process and report performance on the freely available music separation database MUSDB100. Not only such software provides for the most fair MSS methods comparison practice up to date, but it also contains the current official Python BSS Eval toolbox as well as the implementation for three oracle separation methods. In addition, SiSEC campaign provides multi-track datasets that contain the isolated target source signals necessary to compute the evaluation metrics. Moreover, the organisers of SiSEC have recently released the previously mentioned *Open-Unmix* platform¹, serving as a reference method to promote collaborative work within the community [125].

In this thesis we use the public *Demixing Secrets Dataset* DSD100 dataset², the main constituent of the recently released MUSDB100, used in SiSEC 2016 [83]. The DSD100 is a multi-track dataset of 100 full length songs of different styles, containing a mixture track along side the isolated drums, bass, vocals and "others" stems. The mixture is the sum of all the isolated signals, which are all encoded at 44100Hz. All tracks are stereophonic but in this thesis we will down-sample to monophonic signals by taking the mean from both channels at a given time.

Regarding the BSS Eval metrics, it is important to keep in mind that they are merely proxies for perceptual quality and that they are therefore not always aligned with human auditory assessment. In fact, there have been several studies questioning its precision and correlation with human perception altogether [18, 34, 55], and alternatives have been proposed such as the Perceptual Evaluation methods for Audio Source Separation (PEASS) toolkit [131]. Moreover, a recent study seems to suggest the community has been using a misleading SDR definition sensitive to energy scale variations and so they propose to use the scale-invariant definition of SDR [74].

¹ More information available at <http://sigsep.github.io/open-unmix>

² Available at <https://sigsep.github.io/datasets/dsd100.html>

In addition, the standard global SDR computation has some known flaws handling silent or near silent segments. More precisely, to compute the SDR following the SiSEC procedure, the track is divided into non-overlapping segments that are then individually processed. The segment-wise SDR are then averaged to obtain the overall track SDR, which will then be averaged with the SDR of the other tracks in the dataset to obtain the global SDR indicating the overall separation performance across the full dataset. If a segment is silent its SDR will be undefined and the current solution is to ignore such value when averaging. However, if any segment is near silent, its SDR value is not ignored but it will be exceptionally low, biasing the overall average. Some have suggested to use the median instead of average value as a work-around to this issue [124], but, up to this date, there is no consensus on what is the best alternative practice to calculate global BSS Eval metrics.

In this thesis, we will follow the common practice until this date, and use the SDR and its normalised version as evaluation objective metrics.

2.6 SUMMARY

This thesis is interested in blind source separation methods for an underdetermined scenario of musical source separation tasks given a monophonic audio recording. In particular, most of the dissertation will revolve around the family of source separation methods based on median filtering, known as the Kernel Additive Modelling (KAM) framework.

Now that basic source separation concepts have been established and a brief background of the relevant field has been outlined, the mathematical notation will be introduced in the following Part ii accompanied by an increment in depth of the explanations and definitions. Part ii focus on the KAM framework, defined in Chapter 3, as well as its limitations and proposed extensions described in Chapter 4. In order to ease the reading of this thesis, the relevant state-of-the-art methods will be fully defined only when they are required in, or com-

pared to, the proposed methods. For further literature reading on the topic of music sound source separation, we recommend the following references as a starting point: [19, 36, 92, 96, 109, 127].

Part II

MEDIAN FILTERING IN SOURCE SEPARATION

The family of source separation methods based on median filtering, known as the Kernel Additive Modelling (KAM) framework, is introduced, as well as the adopted mathematical notation. Further analysis and discussion will expose the framework's limitations, followed by proposed solutions to alleviate them.

3

KERNEL ADDITIVE MODELLING : KAM

In this chapter we introduce the main framework concerning this dissertation alongside the mathematical notation adopted for the rest of the document. We will focus on the family of methods performing music source separation through median filtering, which fall under the name of Kernel Additive Modelling (KAM). The relevant popular sub family of methods concerning this dissertation will be further defined.

3.1 THE FRAMEWORK

Our ability to distinguish between sound sources in a mixture has been shown to often rely on local features of the sources such as repetitiveness, continuity and common fate [11]. In consequence, a number of SSS techniques propose to use simple local models of sound sources, such as self-similarity of percussive instruments across a small number of frequency bins [39] or periodic self-similarity in time of the backing track in popular music [41, 106, 108]. By defining a local model of the target source characteristics one can regard the other sources as outliers to that model and use robust statistics to remove them. These SSS techniques are based on median filtering and can all be considered instances of the kernel additive modelling (KAM) framework.

The KAM framework takes the idea of using simple local models and generalises it by assuming that a source at some location can be estimated by its values at other similar locations, defined by a proximity kernel [82]. Assuming that several sources overlap in a specific bin in a time-frequency representation, the idea is to reconstruct the magnitude of a given source in that bin by analysing the values in

other bins, in which the target source is likely to assume similar values. This approach is ultimately exploiting the repetitive nature of music to separate sources from a given mixture.

In order to select time-frequency bins with similar occurrences of the target source, a source-specific function is defined known as the proximity kernel. Such function ought to capture the inherent properties of the target source (or sources), such as the broadband nature of percussive sounds or the sparsity of vocals in contrast to the musical accompaniment. A major advantage of the flexibility in the proximity kernel design is its adaptability to different types of sources, rendering the whole KAM framework relatively rich, both in possible application scenarios and theory.

Formally, KAM can be divided into three core steps: modelling of the source signals, definition of the local model and separation step. These steps translate into a joint minimisation of three cost functions: source cost function, model cost function and separation cost function. Starting from a single observation, this joint minimisation is performed iteratively using the new source estimate as an input until a stopping criterion is met. In the KAM literature, this is referred to as the kernel backfitting algorithm (KBF).

KBF could be regarded as a de-noising algorithm that, given some prior knowledge, improves the estimates of sources through a procedural source-specific operation. The previously mentioned SSS techniques based on median filtering can be regarded as instances of KAM using only one iteration of the KBF [39, 41, 106–108]. The methods presented in this thesis extending the KAM framework will similarly be defined for one iteration of the KBF algorithm but are, however, just as valid in the full framework.

In KAM applications for audio source separation the observation mixture x is generally taken as a sum of J unknown sources $\{s_j\}_{j=1,\dots,J}$, and in the monophonic case it can be written as:

$$x = \sum_{j=1}^J s_j \quad (10)$$

All samples from each source are then assumed to be independent from each other and to be Gaussian distributed. In this way, the source cost function is minimised by the observed value itself. Note that the assumed independence between samples in each source does not mean they are not related.

In the following, let $X, S_j \in \mathbb{C}^{F \times T}$ be time-frequency representations of x and s_j respectively, where F is the number of frequency bands and T the total number of time frames. $\bar{X}, \bar{S}_j \in \mathbb{R}_+^{F \times T}$ are the corresponding magnitudes.

The next step in KAM is to define a local model for each source characterised by the proximity kernel of choice. Regardless of its name, the "local" model does not imply proximity in location, but proximity in shape determined by the kernel. For example, we could choose the kernel function to assign proximity to every time-frequency bin separated in time by a certain period τ , and so the local model will include $(f, t + \tau)$ as proximate to the TF bin (f, t) and not $(f, t + 1)$.

A popular kernel choice is the uniform k nearest neighbours function which assigns a positive proximity value to the k most similar locations. More precisely, let \mathbb{L} be the set of all TF bins (f, t) such that $\mathbb{L} = F \times T$. The k -NN kernel function will specify for every TF bin $(f, t) \in \mathbb{L}$ a set of k neighbour bins $\mathcal{I}_j(f, t) \in \mathbb{L}^k$ containing the k most similar magnitude occurrences of the target source j present in the TF bin (f, t) , such that

$$\forall (f', t') \in \mathcal{I}_j(f, t), \quad \bar{S}_j(f', t') \approx \bar{S}_j(f, t) \quad (11)$$

$$\forall (f, t) \in \mathbb{L}, \quad |\mathcal{I}_j(f, t)| = k \quad (12)$$

where $\bar{S}_j(f', t')$ is the magnitude corresponding to source j of the k neighbours in the kernel and $\bar{S}_j(f, t)$ is the magnitude occurrence of the target source j in the (f, t) TF bin.

Essentially, the kernel indicates the similarity between time-frequency bins of the target source \bar{S}_j . Since the target source is the constituent of interest in the observable mixture \bar{X} , knowing which bins contain similar target source contributions is the key to identify and separate it from the other unwanted sources.

In the usual case where the target source is overlaid by other sources in \bar{X} , one can now use the similarity information given by the kernel function to identify which TF bins in \bar{X} have a common target source contribution and restore the overlaid bin and produce the estimate $\hat{\bar{S}}_j$. To this end, this estimation problem in KAM is expressed as a minimization of a model cost function \mathcal{L} , which can be stated for a single channel as follows:

$$\hat{\bar{S}}_j(f, t) = \operatorname{argmin}_{\lambda \in \mathbb{R}} \sum_{(f', t') \in \mathcal{I}(f, t)} \mathcal{L}(\bar{X}(f', t'), \lambda). \quad (13)$$

Depending on the choice of \mathcal{L} the information in the bins indexed by \mathcal{I}_j is merged in different ways. The choice should take into account that, while all these bins are similar in \bar{S}_j , there might be considerable differences between them in \bar{X} due to the overlaying unknown sources.

A popular choice is the absolute deviation $\mathcal{L}(a, b) := |a - b|$ as it is known to be robust against outliers, since it expresses, from a probabilistic point of view, that we expect some larger deviations in the difference a and b , and so the distance should not be Gaussian distributed (the loss would increase quadratically otherwise). In this case we can further express the estimation problem in KAM as:

$$\hat{\bar{S}}_j(f, t) = \operatorname{argmin}_{\bar{S}_j(f, t) \in \mathbb{R}} \sum_{(f', t') \in \mathcal{I}_j(f, t)} |\bar{X}(f', t') - \bar{S}_j(f, t)|. \quad (14)$$

The solution to the above problem employs operators from robust statistics (order statistics), which enable unbiased parameter estimation in the presence of up to 50% outliers. With this choice of \mathcal{L} , the solution of the estimation problem (Equation (13)) is:

$$\hat{\bar{S}}_j(f, t) = \operatorname{median}(\bar{X}(f', t') \mid (f', t') \in \mathcal{I}(f, t)), \quad (15)$$

$\hat{\bar{S}}_j$ being the magnitude estimate of the source of interest s_j . The derivation of this solution is detailed below, following the generic approach described in [3].

For simplicity, let

$$\mathcal{L}(\bar{S}_j(f, t)) = \sum_{(f', t') \in \mathcal{I}(f, t)} |\bar{X}(f', t') - \bar{S}_j(f, t)|$$

and be expressed as follows :

$$\mathcal{L}(\beta) = \sum_{i=1}^k |\bar{X}_i - \beta|$$

Assume the observed data \bar{X}_i to be arranged in ascending order of magnitude such that $\bar{X}_1 \leq \bar{X}_2 \leq \dots \leq \bar{X}_k$. For values of β in the range $\bar{X}_p < \beta \leq \bar{X}_{p+1} \forall p = 1, 2, \dots, k$ we can write

$$\mathcal{L}(\beta) = \sum_{i=1}^p (\beta - \bar{X}_i) + \sum_{i=p+1}^k (\bar{X}_i - \beta) = \left(\sum_{i=p+1}^k \bar{X}_i - \sum_{i=1}^p \bar{X}_i \right) - (k - 2p)\beta$$

$\mathcal{L}(\beta)$ is clearly linear and continuous, with a slope of $(k - 2p)$ increasing by 2 for each ascending p value. If k is odd, there is an integer m such that the slope over the intervals $(\bar{X}_{m-1}, \bar{X}_m]$ and $(\bar{X}_m, \bar{X}_{m+1}]$ are negative and positive respectively. These two conditions are met if $m = \frac{k+1}{2}$. However, if k is even, there is an integer m for which the slope over $(\bar{X}_m, \bar{X}_{m+1}]$ is zero, which is possible for $m = \frac{k}{2}$.

Thus a solution to the estimation problem is:

$$\hat{\beta} = \underset{\beta}{\operatorname{argmin}} \sum_{i=1}^k |\bar{X}_i - \beta| =$$

$$= \begin{cases} \bar{X}_{\frac{k+1}{2}} & k \text{ odd} \\ (\bar{X}_{\frac{k}{2}}, \bar{X}_{\frac{k+1}{2}}] & k \text{ even} \end{cases} = \text{median}(\bar{X}_1, \bar{X}_2, \dots, \bar{X}_K)$$

Therefore, in order to achieve the magnitude estimate of the target source \hat{S}_j , every TF bin is replaced by the median value of the kernel bins. In order to perform the actual separation, the complex estimate \hat{S}_j is needed so it can consequently be inverted back to a time domain signal. This is typically done using a soft-masking approach [19, 92, 125].

3.2 KAM IN THIS THESIS

Since KAM is a considerably broad framework as a whole, we will, from now on, focus on a subset summarised in the table 1, that was

also used in a similar form in the REPET (REpeating Pattern Extraction Technique) family of methods mostly applied to singing voice removal [108]. However, most of the insights this thesis provides on this reduced subset are as valid for the whole framework.

Therefore, for ease of understanding and applicability, let x be the signal to be processed with

$$x(t) = s(t) + n(t) \quad (16)$$

where s and n are the clean music (i.e. target source) and the interference signal (i.e. unwanted source), respectively. Further, let $X, S \in \mathbb{C}^{F \times T}$ be the spectrograms of x and s .

The interference signal n is usually considered to be sparse compared to the music signal s assumed to be repetitive. The vocal component in a given mixture is also often assumed to fluctuate and to be sparse in comparison to the musical accompaniment which is expected to be repetitive and powerful [109]. Therefore, in a vocal/accompaniment separation scenario the vocals would correspond to n and the accompaniment to s .

In the case of an interference reduction task where the goal is to reduce the impact of short burst-like sound events such as coughs, s will represent all the musical component in the given signal and n the interference itself. Both vocal/accompaniment and interference reduction will be considered applications in this thesis. It is however important to notice that the definition of the framework is the same for both those scenarios.

In the following, we exploit that spectral frames in S typically occur several times in similar form, either because note constellations are repeated over time (as is common in music) or because notes are being held for a while. The unwanted source on the other hand may or may not be repetitive and thus we do not make any assumptions on it. Therefore, we will model only s in KAM without considering the interference n as an actual sound source but just as noise with an unknown distribution.

Since s only consists of a single channel, we can eliminate many unnecessary elements in KAM (multi-channel and iterative re-estimation

	KAM framework	KAM in this thesis
Applications	Source Separation	Audio Source Separation:
Data type	multitrack	monophonic
KBF iterations	~ 5	1
Nb source models	variable	1
kernel function	variable	frame-wise k-NN
model cost function	variable	$ a - b \Rightarrow$ median
separation function	variable	soft masking

Table 1: Summary of the KAM subset framework concerning this thesis.

extensions, compare [82]), resulting in a very simple representation. As in the case of REPET and other vocal separation methods based on median filtering within the KAM framework [41, 106, 108], we use a *frame-wise*, k-nearest neighbours (k-NN) kernel function based on the Euclidean distance, i.e. (f, \tilde{t}) is in $\mathcal{I}(f, t)$ if frame \tilde{t} is among the k most similar frames. Therefore, we can now rewrite the KAM optimization problem over the model cost function \mathcal{L} as follows:

$$\hat{\bar{S}}(f, t) = \operatorname{argmin}_{\lambda \in \mathbb{R}} \sum_{(f, \tilde{t}) \in \mathcal{I}(f, t)} \mathcal{L}(\bar{X}(f, \tilde{t}), \lambda) \quad (17)$$

Here we choose the common cost function $\mathcal{L}(a, b) := |a - b|$ which basically models our belief regarding how good or bad a specific choice for $\bar{S}(f, t)$ is, considering that we call all elements in $\mathcal{I}(f, t)$ similar to it. As the derivation above shows, this choice leads to the use of robust statistics in the form of the median, which as an operator is invariant against outliers (breakdown point is 50%) and thus allows robust parameter estimation in the presence of noise. More precisely, the solution to the problem in Equation (17) is:

$$\hat{\bar{S}}(f, t) := \operatorname{median}(\bar{X}(f, \tilde{t}) | (f, \tilde{t}) \in \mathcal{I}(f, t)). \quad (18)$$

To perform the actual separation, we employ soft masking (similar to Wiener filtering). In an interference reduction application, we are only interested in yielding an estimate $\hat{\bar{S}}$ for the magnitude spectrogram of the music. Therefore, we define an estimate of the unwanted

source as $\hat{N} = \max(\bar{X} - \hat{S}, 0)$ so we can obtain an estimate \hat{S} for the complex music spectrogram via

$$\hat{S} = \frac{\hat{S}}{\hat{N} + \hat{S}} \odot X \quad (19)$$

where \odot represents element-wise multiplication, preceded by an element-wise division.

In the case of vocal/accompaniment separation, similarly to [41], we use a more sophisticated mask to extract both accompaniment and vocals from the mixture. More precisely, we measure the distance between the mixture \bar{X} and the accompaniment estimate \hat{S} after a logarithmic compression (with the logarithm leading to a perceptually more meaningful distance [41]) and employ this distance in a Gaussian radial basis function to obtain a mask $W \in \mathbb{R}^{F \times T}$:

$$W(f, t) = \exp \left(-\frac{(\log(\bar{X}(f, t)) - \log(\hat{S}(f, t)))^2}{2\lambda^2} \right) \quad (20)$$

for all TF bins (f, t) in $F \times T$ and where λ is a parameter to additionally compress the log-distances non-linearly. Here we set $\lambda=1$.

The complex spectrograms for the accompaniment $\hat{S} \in \mathbb{C}^{F \times T}$ and vocals $\hat{N} \in \mathbb{C}^{F \times T}$ can then be estimated by applying the soft masks W and $(1 - W)$ to the original mixture spectrogram X respectively:

$$\hat{S} = W \odot X \quad (21)$$

$$\hat{N} = (1 - W) \odot X \quad (22)$$

3.3 SUMMARY

The Kernel Additive Modelling (KAM) framework comprises the popular and computationally efficient family of methods for source separation based on median filtering. Such a flexible framework, applied in multiple music source separation tasks [39, 41, 82, 106, 108], requires no training data as it targets the inherent properties of the sources directly by taking advantage of their differences in the separation process.

The basic idea behind KAM is that one can reconstruct the magnitude for a given source by analysing the values at the locations where the source is likely to assume similar values, ultimately relying on the assumed repetition of sound events in musical signals. The success of the separation will depend on the ability to identify similar sound events to the source of interest in the presence of overlaying sources. The source similarity is determined by a source-specific kernel function, which often corresponds to a k nearest neighbours (kNN) search based on the Euclidean distance.

From now on, we will focus on the popular KAM subset of methods using the frame-wise k-NN kernel function for monophonic signals [41, 106, 108]. We will take the input mixture to be a linear sum of two sound sources, the target (s) and the unwanted (n) source. The aim will be to reconstruct the target source, reducing the impact of the unwanted source in the mixture. We will assume the interference to be sparse regarding the target source, and of unknown distribution. Therefore we will only focus on modelling the target source. In addition, we will only consider one iteration of the kernel backfitting algorithm (KBF) as it is independent from the modelling stage of interest to this thesis and therefore it would only improve the separation results further acting as a post-processing de-noising algorithm.

In the following Chapter 4, we will discuss further the flexibility of the KAM framework and we will be particularly interested in its limitations in Section 4.1. The rest of the chapter (Sections 4.2 to 4.4) provides extensions to the framework to overcome such limitations, comprising one of the major contributions of this thesis. The proposed methods will be initially outlined and then formally introduced within the framework, to be further evaluated in a music separation task. We will consider two main music separation tasks: vocal/accompaniment separation and burst-like interference reduction. The latter being introduced for the first time in this thesis.

4

IMPROVEMENTS AND OBSERVATIONS ON KAM

In this chapter we unveil and discuss some limitations of the popular kernel choice in KAM presented in section 3.2, developing a sense of the kernel behaviour at work. We further propose several extensions to the framework to alleviate such limitations, offering a view on how signal processing techniques can help overcome non trivial cases, such as loud noise in music recordings.

4.1 THE DARK SIDE OF KAM

As fully described in the previous Chapter 3, to apply a KAM-based method to a source separation problem, one needs to design a corresponding kernel that identifies similar spectral bins for the target source while ignoring the energy associated with other sources. The success of separation in KAM ultimately depends on the ability of the kernel to identify frames with similar target source energy in the presence of overlaying sources.

In existing KAM approaches, the kernel design is often rather rudimentary. In particular to this thesis, the frame-wise k-NN kernel is a simple function based on Euclidean distance finding the k most similar frames to a given current frame. Even though a such kernel can exploit some of the source's regularities, its simplicity leads to further drawbacks.

To illustrate this, let's consider a recording containing two instrumental solo sections for a piano and a guitar. Depending on the recording conditions, the sustain part for both instruments can have a similar energy distribution in frequency direction (playing the same musical pitch), as it can be observed in Figure 5 where the frames

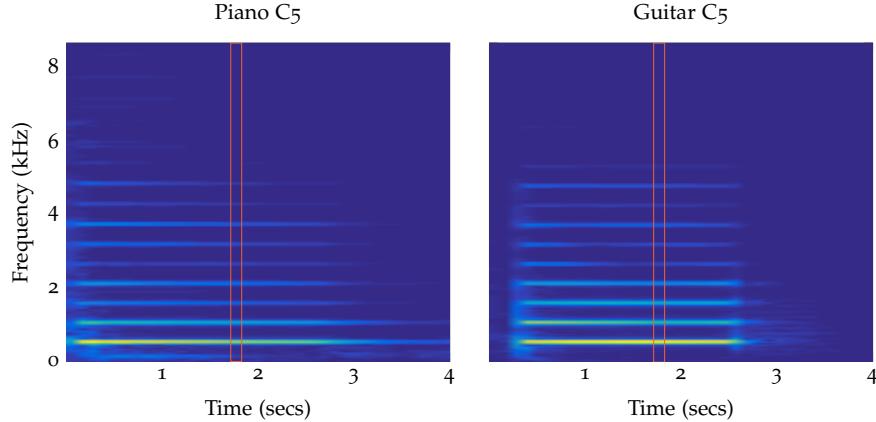


Figure 5: MIDI C5 spectrogram of two different synthesised instruments, piano and guitar, illustrating the importance of the temporal context when defining a similarity kernel, that could mistakenly relate the two frames in red if the surrounding is not taken into account.

in the sustain part of the note (like the one in red) of the piano and guitar could be considered similar. As a consequence, a frame-wise kernel based on the Euclidean distance sometimes fails to identify the intended dissimilarity between frames and can confuse a guitar frame with a piano frame. Such issues are even more pronounced if an instrument has variable timbre, for example due to the use of effects. This mix-up can lead to an unexpected energy distribution for an instrument in the separation result.

Using only a single frame, such issues are difficult to resolve. However, by taking the temporal context of a frame into account, we obtain more information about which frames are actually similar to each other for a given target source. For example, using a larger temporal context, the similarity measure might take a frame containing the onset into account, which can be very discriminative for an instrument, as it can be seen in Figure 5 where the difference between these sources is clearly visible in the 4 seconds window presented. Also, the temporal context might even be large enough to pick up some basic information about the musical context and, assuming the different instruments play different note patterns, we can use this low-level musical context as additional guidance to find similar frames for a given instrument. Based on this simple idea, Section 4.2 below shows how to introduce a temporal context to existing kernels and the benefits of it.

Furthermore, the notion of similarity derived from such a simple kernel can be quite limited as it fundamentally relies on the following assumptions. Firstly, the target source is considered to be closer to stationary than the overlaying source, considered to be sparse. For example, in the case of vocal separation outlined in Section 3.2, this means that we expect many time frames containing the same, or similar, accompaniment but not many with the same voice content.

Secondly, this kernel choice assumes the target source to repeat in time at the same frequency, meaning the position of partials and other objects must be the same within the frames selected by the kernel. While this might be a valid assumption for full-length pop songs, it might be wrong if the recording is short, the source is consistently overlaid with the same interference in each repetition or for sources with highly variable pitch.

One way to alleviate such restrictions is to increase the sound material available for the sound reconstruction by including frames with notes of the target source with different pitch but similar frequency constellation. Section 4.3 below details how to do so by extending the KAM framework in the form of a shift-invariant kernel.

Finally, in the KAM framework it is implicitly assumed that the energy in each time frame of the given mixture is dominated by the target source. So, in a vocal separation example, the energy contribution of the accompaniment is assumed higher than that of the vocal source, in the same way any audio interference is inherently assumed to be less powerful than the target music. This means that if the signal-to-interference (i.e. signal-to-vocal) ratio is low, the kernel function will fail to find similar frames.

In such case where the target source is masked by the other source, KAM would need an additional initial step to discriminate between the sources and yield a preliminary signal model to input the framework. Section 4.4 explains how to achieve so and how to design an adaptive, interference-resilient kernel by combining NMF with the KAM framework.

4.2 TEMPORAL CONTEXT

In order to improve the similarity notion between frames we can take advantage of the information in neighbouring frames by introducing a temporal context in the kernel function. Basically, given a frame we aim to find similar frames for, we simply include the preceding and succeeding frames in the similarity function underlying our kernel. Effectively, that means we measure similarity based on entire groups of frames instead of single frames. The size of the temporal context should be chosen large enough to give some rough indication of local musical patterns.

Following the notation of the previous Chapter 3, a time frame (f, \tilde{t}) is now in $\mathcal{I}(f, t)$ not only if $\bar{S}(f, \tilde{t}) \approx \bar{S}(f, t)$, but also for its neighbouring frames such that:

$$\forall (f, \tilde{t}) \in \mathcal{I}(f, t), \quad \bar{S}(f, \tilde{t} + c) \approx \bar{S}(f, t + c) \quad \forall c \in [-C, C] \quad (23)$$

where C specifies the temporal context. Therefore, instead of comparing frames t and \tilde{t} with a simple squared Euclidean distance

$$\sum_f (\bar{X}(f, t) - \bar{X}(f, \tilde{t}))^2 \quad (24)$$

we employ

$$\sum_f \sum_{c=-C}^C (\bar{X}(f, t + c) - \bar{X}(f, \tilde{t} + c))^2 \quad (25)$$

as frame distance in the k-NN search. We maintain the Euclidean distance in the kernel as we care for both the magnitude value and location of frequency bins in the group of frames.

In Figure 6 both of these distances are compared for a given frame t , with $t = 1000$ in the left and $t = 1120$ in the right half of the figure. The top row shows the frame-wise distance (equation (24)) values between t and all other frames, and the middle row shows the temporal context distance (equation (25)) values between a 1s segment centred in t and all 1s segments centred around all other frames. The spectrogram of the mixture to be processed is shown in each half of the figure and the two frames are indicated by a vertical red line. The yellow and magenta lines in Figure 6 indicate the most similar frame

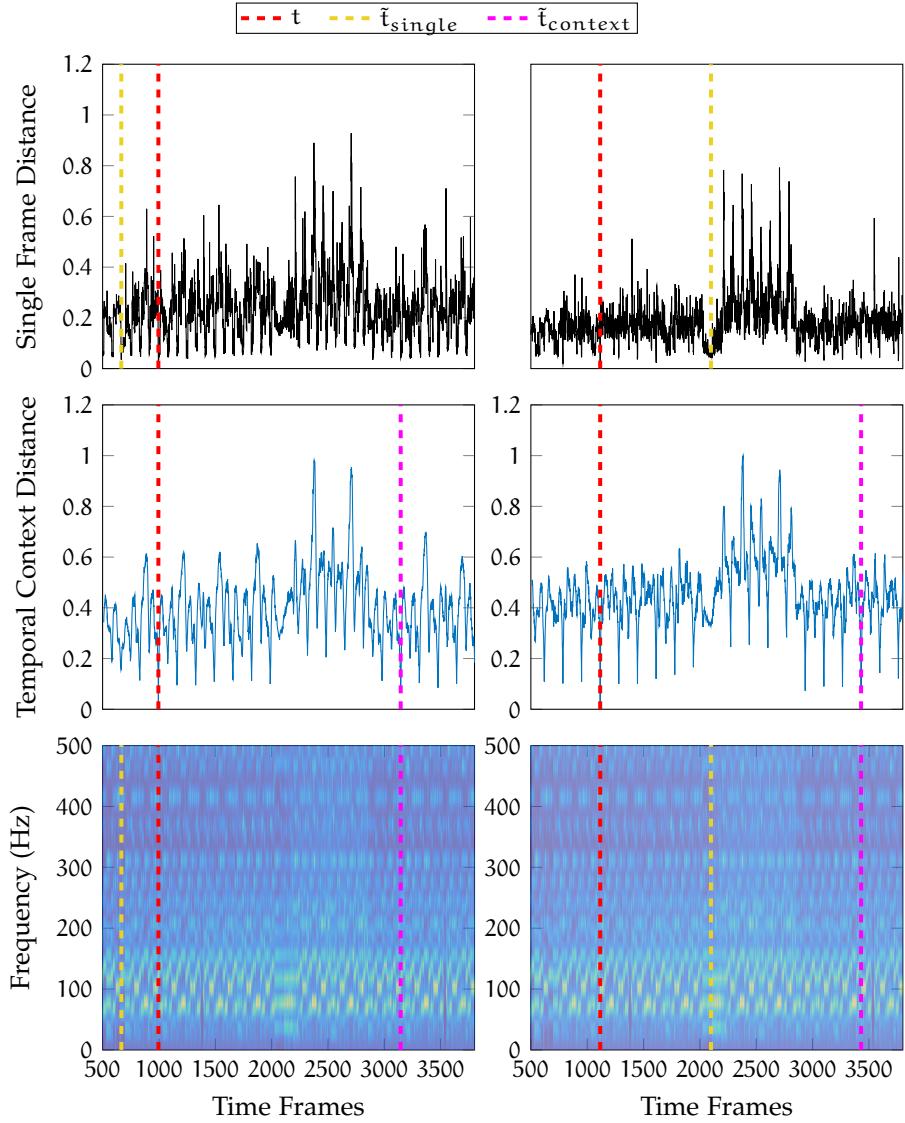


Figure 6: Comparison between frame-wise (top row, equation (24)) and temporal context distance (middle row, equation (25)) for a given frame t (in red) in a 2.5 minutes segment of a sample song mixture magnitude spectrogram (bottom row). The closest nearest neighbours, $\tilde{t}_{\text{single}}$ and $\tilde{t}_{\text{context}}$, are highlighted for both the frame-wise (yellow) and temporal context (magenta) kernels respectively.

found using the single frame and temporal context distance, respectively.

As we can see, the single frame distance (top row Figure 6) is much more noisy compared to the one with temporal context (middle row Figure 6). Also, peaks indicating a low distance (i.e. high similarity) are much clearer for the curve using a temporal context; this is particularly visible in the right example of Figure 6 where many spurious peaks can be found in the single frame distance (top row). Such overall change in qualitative behaviour also influences which frames are selected as the most similar frames, as the distance values between similar (i.e. "clear" peaks) and not-similar frames using a temporal context are different enough to avoid confusion in the nearest frame selection, unlike the distance values obtained with a single frame kernel.

The nearest neighbour selection for the example frame highlighted in red ($t = 1120$) on the right half of Figure 6 exhibits this behaviour. As we can notice, the nearest frame selected (in yellow) via the "noisy" single frame distance (first row in black) is in a completely different section of the song with different frequency components, and so introducing unwanted noise in the filtering stage. On the other hand, the temporal context distance of that section presents rightly much higher values (second row in blue), ensuring a better selection of actually akin frames with similar frequency components, like the frame selected as nearest neighbour (in magenta). In some cases both distances present a low value for a frame that makes sense musically, as can be seen in the left side example of Figure 6 where both selected frames (in yellow and magenta) happen at the end of a similar note passage. Even though this is not always guaranteed, the introduction of a temporal context helps towards having a musically sound nearest neighbour, further stabilising the kernel.

Re-using the previous example of a mixture containing a guitar and piano solo illustrated in Figure 5, if the current frame is in the guitar solo, when looking for similar frames with the basic kernel, one may find confusion with the piano solo. However, if we introduce a temporal context in the kernel, the newly created group of frames centred around this frame might actually span a few notes. Therefore, when

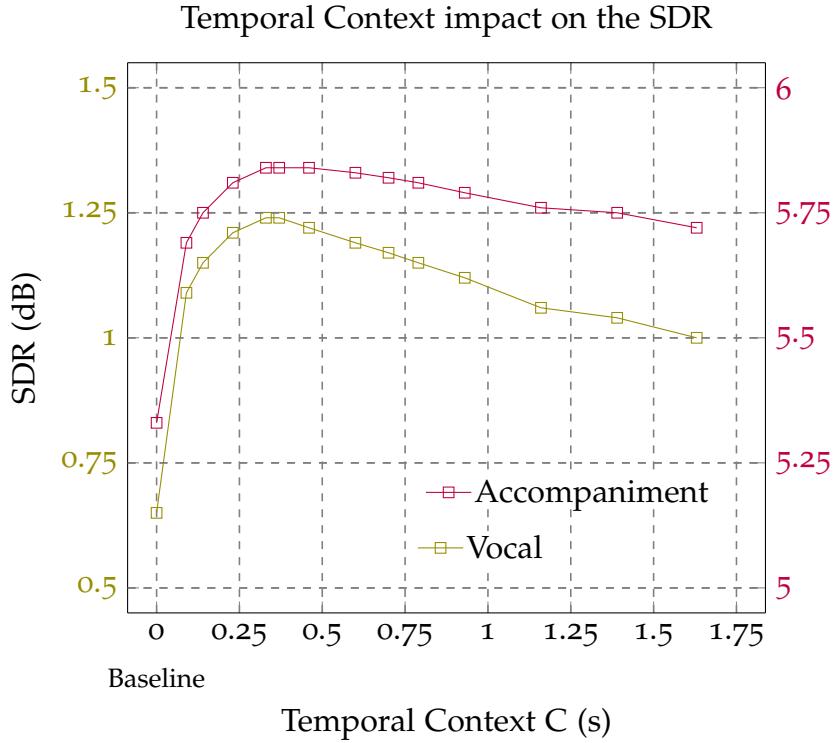


Figure 7: SDR results for the proposed extension with different temporal contexts for the DSD100 dataset.

looking for similar segments, we can take this local note constellation to some degree into account, which potentially aids in differentiating between similar timbres. In particular, we would expect the guitar to not be mistaken for the piano any more.

4.2.1 Empirical evaluation

To quantitatively compare the frame-wise kernel with and without temporal context, we now focus on a vocal separation task. We consider the first 30 seconds of each song in the Demixing Secrets Dataset 100 (DSD100) from the 2016 Signal Separation Evaluation Campaign (SiSEC) [83] and the Signal to Distortion Ratio (SDR) BSS Eval toolbox 3.0 to assess the separation quality [132], both described in Section 2.5.

For this evaluation, the baseline is the instance of KAM for vocal separation implemented using an FFT size of 4096 and a hopsize

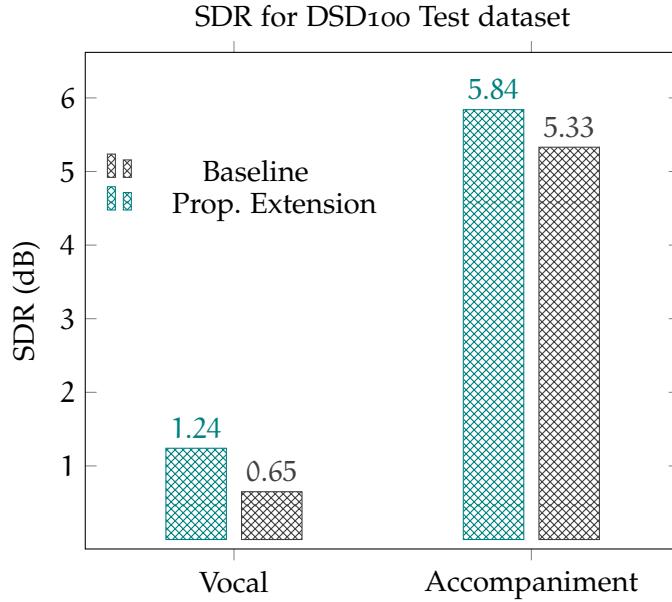


Figure 8: SDR results for the proposed extension (left bars in green) and the baseline method (right bars in black) tested on the DSD100 dataset using C selected by parameter sweep.

of 2048 samples as in [41]. The proposed method extends this baseline by introducing a temporal context in the proximity kernel as described in Section 4.2. The number of frames C specifying the temporal context is the parameter of this extension.

In principle, this parameter could be adapted for each frame based on musical knowledge, for example, based on segmentation information or pitch tracking data, which would render the method more flexible and adjustable to musical changes in the signal. However, we chose to use a fixed setting for the context C to set a benchmark of its overall influence as we expect an informed dynamic method to only improve the performance. Therefore, to find a suitable value, we can conduct a simple parameter sweep, as the one shown in Figure 7.

Figure 7 shows the averaged SDR values for both vocal and accompaniment separation using the proposed extension for different context values. We can observe an overall trend in Figure 7 shared by both vocal and accompaniment separation, where the biggest difference in SDR value is between a zero radius (the baseline method) and the other values taking a temporal context into account. In addition, we see that the highest SDR values are achieved for a temporal

context of around 1 second (C values between 0.25 and 0.6 seconds), which can be considered wide enough to capture some simple musical patterns. If C is increased, the musical information within the temporal context grows and we observe a slight decrease of the SDR. For this reason, C is fixed to 0.372s for this experiment.

Using this fixed value for C , Figure 8 shows the SDR values comparing the frame-wise baseline kernel with the extension proposed in this thesis introducing a temporal context in the kernel. On the SiSEC dataset, the proposed extension consistently outperforms the baseline and improves the results by about 0.5dB SDR on average for both vocal and accompaniment separation, representing, in particular, a substantial improvement of the vocal estimate. Given its simplicity, this is quite considerable.

Overall, the results show the advantage of introducing a temporal context in the similarity search. The presented method is simple and unsupervised, requires no prior training (other than the potential optimised choice of the temporal context C), and temporally stabilises the source estimates, improving the separation performance over the baseline frame-wise k-NN kernel function.

4.3 SHIFT-INVARIANT KAM

KAM framework relies on the repetitive nature of music. This means that we expect to find enough frames with the same (or similar) target source overlaid by different constellation of frequencies from the other sources in order for the separation to succeed. In particular, the proportion of identified nearest neighbours with overlaying sources at the same position needs to be less than half or it would surpass the 50% of outliers breakdown point of the median operator.

To illustrate this, in Figure 9 every box represents a time frame, the target source is represented by a snake (box E) and the overlaying sources by the coloured circles. The four boxes (A,B,C,D) represent the k-NN and the goal is to reconstruct the snake by taking the median value amongst them (box F). Overlaying sources that appear in

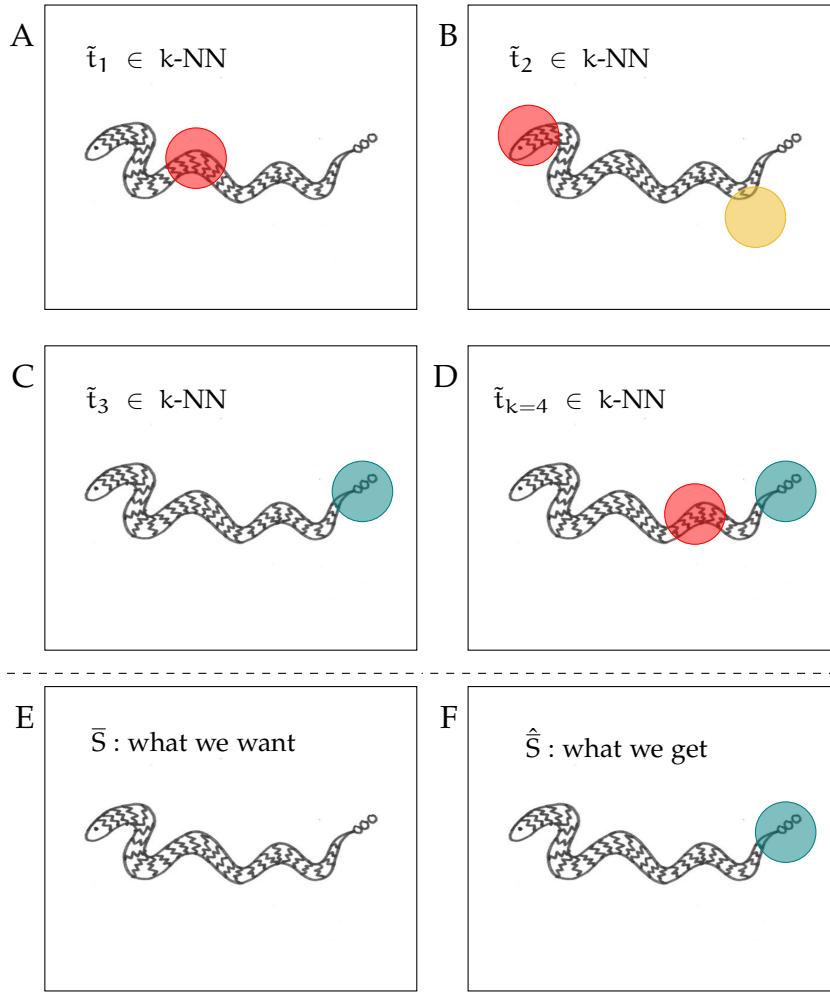


Figure 9: Illustration of the different cases of overlaying sources within the kernel function. Every box symbolises a time frame, the snake represents the target source and the circles the overlaying sources. Overlaying sources present in less than half of the frames in yellow, present in more than half of the frames in different positions in red and in the same position in green. Being the latter one the problematic for KAM as the median operator is only robust to up to 50% of outliers in the same position.

less than half of the frames (yellow circle in B) can be considered as outliers and will be therefore removed by the median operator.

The issue is not as straight-forward for interferences that appear in more than half of the nearest neighbours. If the overlaying source is present in more than half of the frames but in different locations, like the red circles in Figure 9, it can still be considered as outliers and so the median filter will be able to restore the target source. For example,

even though the snake's head is overlaid by the red source in box B of Figure 9, the median value amongst all boxes will correspond to that of the snake's head alone as none of the other boxes have an interference at that location, and so the head of the snake could be restored. However, the snake's tail has a different fate since it is overlaid by the same green interference in 3 out of the 4 boxes, and so its contribution will remain when taking the median value amongst all boxes. This scenario corresponds to overlaying sources that appear in more than half of the k-NN at the same location and is one of the cause of failure in the separation stage.

A way to avoid such unwanted case, could be to pick a k large enough to ensure all overlaying sources appear as actual outliers given the target source is the most repetitive one or that there are enough repetitions to gather a large enough pool of close frames. However this is rarely the case in music signals, as sources are correlated and it is common to find different sources repeating at the same time. In addition, even if the target source was distinctively much more repetitive than the other sources, it would still be a problem if the recording is short or the target source varies highly in pitch (e.g. vibrato).

On the other hand, it is clear that what all the target source appearances have in common is that they are all issued from the same source. This implies that their frequency pattern will somehow be related regardless of the pitch. Therefore we propose to use such relation to increase the pool of potential nearest neighbours candidates to reduce the percentage of repetitions of unwanted sources in the kernel.

KAM implementations typically use a standard linear scale time-frequency representation as it is both memory and computationally inexpensive as mentioned in Section 2.3. In such a representation, the spacing between harmonics will depend on the fundamental frequency. However, using a logarithmic frequency scale, the location of every harmonic with respect to the fundamental frequency will be constant [12].

Taking f_0 as the fundamental frequency of a signal, the frequency of the n^{th} harmonic will be located at $n \times f_0$ in a linear scale but would appear at $\log f_0 + \log n$ in a logarithmic frequency scale. In particular, within a certain frequency range, pitch shifts simply correspond to shifts in log-frequency representations.

In consequence, here we propose an extension to the KAM framework in the form of a shift-invariant kernel using a logarithmic frequency axis. This kernel extends the k-NN function by comparing not only the original frames but also all frequency shifted versions. In other words, we expect it to identify notes of the same source differing in pitch as being similar and to reconstruct a unique musical event from them despite the shift, drastically increasing the sound material available for the sound reconstruction.

Following the notation of the previous Chapter 3, let $X_q, S_q \in \mathbb{C}^{F \times T}$ be the Constant-Q transforms (CQTs) of x and s , a log-frequency representation with a perfect reconstruction property [114], and \bar{X}_q, \bar{S}_q the corresponding magnitudes. The goal now is to locate not only patterns repeated in time but also their shifted versions. In order to do so, we introduce a shift δ in the kernel function measured in frequency bins.

To this end, let \bar{X}_q^δ be a frequency shifted version of \bar{X}_q such that

$$\bar{X}_q^\delta(f, t) := \bar{X}_q(f + \delta, t) \quad (26)$$

We define a new shift-invariant kernel \mathcal{I}_s as follows: for a given (f, t) , we have $(\tilde{f}, \tilde{t}) \in \mathcal{I}_s(f, t)$ if $|\delta| < \Delta$ for $\delta := \tilde{f} - f$ and $\bar{X}_q^\delta(:, \tilde{t})$ is among the k closest frames for frame $\bar{X}_q(:, t)$ across all $\delta \in \{-\Delta, \dots, \Delta\}$, where Δ denotes the absolute maximum shift measured in number of frequency bins. Here, we used the slicing notation $:$ to denote all elements in an index dimension.

This means that two time frames can now be considered as neighbours if they display a similar harmonic pattern at different frequency locations. In other words, the proposed kernel function \mathcal{I}_s can be seen as a shift-invariant version of the frame-wise k-NN baseline kernel \mathcal{I} defined in Section 3.2. The estimation problem remains essentially the

same (compare to Equation (17)), just that the variability in frequency is now explicit:

$$\hat{S}(f, t) = \operatorname{argmin}_{\lambda \in \mathbb{R}} \sum_{(\tilde{f}, \tilde{t}) \in \mathcal{I}_s(f, t)} \mathcal{L}(\bar{X}_q(\tilde{f}, \tilde{t}), \lambda). \quad (27)$$

For the same model cost function as above in Section 3.2, we get the solution:

$$\hat{S}(f, t) := \operatorname{median}(\bar{X}_q(\tilde{f}, \tilde{t}) | (\tilde{f}, \tilde{t}) \in \mathcal{I}_s(f, t)). \quad (28)$$

As a result of this extension, we can now recover a note played only once by using notes different in pitch played by the same instrument, as seen in the third row of Figure 10.

In practice, the implementation of this approach can be split into two main steps: similarity measure (Fig 10 B1) and frequency alignment (Fig 10 B2). In particular, every frame in the mixture has to be shifted in frequency direction and compared to the remaining frames, $2 \cdot \Delta$ times. Computing the Euclidean distances in every step is in $O(T^2 \cdot F)$. Altogether, with Δ typically being dependent F , the complexity of this approach is considerable: $O(T^2(F^2 + \log T))$. Note that Δ depends on the chosen frequency resolution, commonly, at least of half tone for Western music.

In practice, even after limiting Δ to a reasonable frequency range (e.g. 24 semitones), a basic implementation of this approach turns out to be computationally quite expensive in comparison to the baseline of overall complexity of $O(T^2(F + \log T))$ as usually $T > F$. Further detail on this complexity derivation can be found in appendix A.

4.3.1 Acceleration Extension

Under runtime constraints, the method above forces the user to trade-off separation performance for better running time. For example, one may set the Δ to cover only half an octave, at the risk of not finding similar events. In order to accelerate the shift-invariant kernel computation while preserving the increase in separation quality, we propose to use a different time-frequency representation to allow a quicker shift-invariant search.

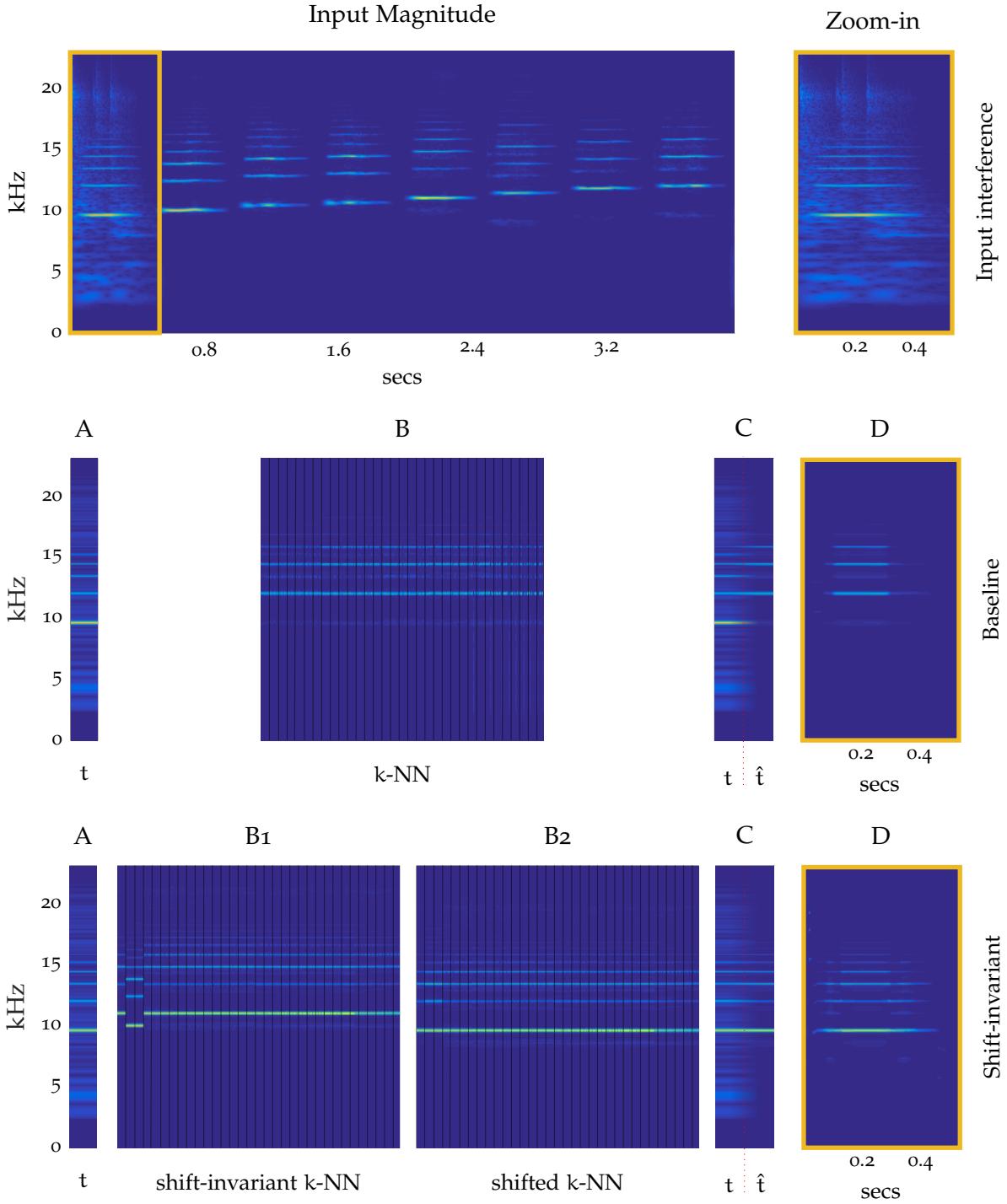


Figure 10: Comparison of the baseline (second row) and the basic version of our proposed shift-invariant method (third row) for an example frame (A) from the input magnitude frames overlaid by an interference framed by a yellow box (Zoom-in). The k closest frames found for the current frame (A) by the baseline (B) and the proposed method, before (B₁) and after the shifting operation (B₂). The plots (C) contain the current frame (t) next to the estimated output frame (\hat{t}) for each method. The complete estimation of the harmonic source for the frames containing interference is shown in D for both methods.

The idea is to employ a representation that captures the harmonic patterns in each frame, while being invariant against their exact frequency location. More precisely, given the magnitude CQT \bar{X}_q , we perform our search on the magnitude spectrum calculated on each frame $\bar{X}_q(:, t)$. This transform is related to cepstral analysis [104] but has more recently been called *specmurt* analysis [111] when applied to a log-frequency linear-magnitude representation (as in our case). In such domain, the frequency of frequencies are referred to "quefrency". We can define the specmurt \bar{X}_s , with \mathcal{F} being the Fourier transform, such that:

$$\bar{X}_s(:, t) := |\mathcal{F}(\bar{X}_q(:, t))| \quad \forall t \in [1, T] \quad (29)$$

The specmurt is invariant against the frequency location of the patterns, since this is encoded in the phase (which we ignore). Hence, we no longer need to shift its frames in the frequency direction to find the nearest neighbours as in Equation (4.3). Now we can simply employ a frame-wise k-NN kernel function (as defined in Section 3.2) on this domain and be shift-invariant.

Using the specmurt domain brings various advantages. First, eliminating the specmurt-phase by using the magnitude value, we eliminate pitch information and keep only the "pattern" information as shown in Figure 11. Second, certain spectral characteristics are represented more compactly. For example, a broadband sound in the time-frequency domain, like the interference overlapping the first note in Figure 11, will correspond to "low-quefrency" components in the specmurt domain. This way, percussive components can more easily be ignored in the similarity search (if needed) and provides an interesting new angle to design source specific kernels by applying different weightings to the specmurt coefficients. Further, we can exploit the symmetry of the Fourier transform to eliminate half of the specmurt components, reducing the run time further. Therefore we can define the employed section of the specmurt $\bar{X}'_s \in \mathbb{R}^{(F/2-\alpha) \times T}$ using a parameter α to set the quefrency (q_f) cut-off point of the broadband information as follows:

$$\bar{X}'_s(q_f, :) = \bar{X}_s(q_f, :) \quad \text{for } q_f \in [\alpha, F/2] \quad (30)$$

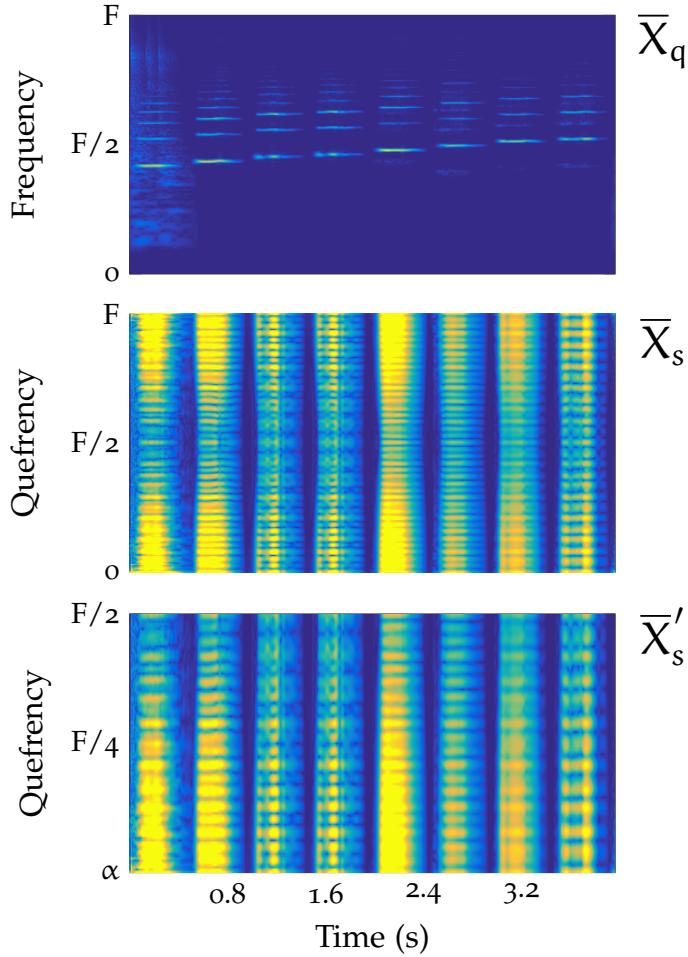


Figure 11: From top to bottom, illustration of the magnitude CQT (\bar{X}_q), magnitude specmurt (\bar{X}_s) and the employed section of the magnitude specmurt (\bar{X}'_s) in the proposed acceleration, of a given mixture of a musical instrument corrupted by a broadband interference on the first note.

Overall, instead of $O(T^2(F^2 + \log T))$ operations for the shifts and Euclidean distances as before, we transform \bar{X} to specmurt and only have to perform one set of Euclidean distances (comparable to the baseline that does not support shift invariance), resulting in only $O(T^2(F + \log T))$ operations for these steps.

Nonetheless, while this approach enables a rapid shift-invariant selection of frames, it does not tell us the shift we need to apply to a CQT frame such that it is indeed similar to a given one. Given an input frame, a first idea is to apply all possible shifts to the k frames found as similar on the specmurt domain. While this is a considerable

speed-up over the plain approach described in Section 4.3, it is still rather slow. Therefore, we will accelerate this step next, again using the Fourier transform, which was explored in [111] in a related form in the context of source-filter modelling.

To this end, we assume we know from the specmurt shift-invariant kernel that frames t and \tilde{t} are similar. For notational purposes, we will use the shorthands $Y := \bar{X}_q(:, t)$ and $Z := \bar{X}_q(:, \tilde{t})$. That means, Y and Z differ mostly by a shift in frequency, which we need to identify. We can express this as $Y = H * Z$ and solve for H , where $*$ denotes a convolution. In case $Y = Z$, $H(0) = 1$ and there is no shift. If all entries in Z are shifted by 1 compared to Y , we obtain $H(1) = 1$. That means, to obtain the correct shift between Y and Z we only need to compute a deconvolution between them and the Fourier transform can again accelerate this step. As detailed in [111], a fast deconvolution can be calculated via

$$H = \mathcal{F} \left(\frac{\mathcal{I}\mathcal{F}(Y)}{\mathcal{I}\mathcal{F}(Z)} \right), \quad (31)$$

where $\mathcal{I}\mathcal{F}$ denotes the inverse Fourier transform.

Assuming that frames t and \tilde{t} are indeed similar, the H we obtain this way will typically be very sparse and essentially have a strong peak at exactly one position, which indicates the shift we need to apply to frame \tilde{t} . Once we have the optimal shift for all k close frames we can continue as in the baseline method. Combining the two acceleration methods, the computational complexity is $O(T^2(F + \log T)) + O(T \cdot F \log F)$. Further detail on this complexity is available in appendix A

Even though measuring similarity based on the magnitude specmurt considerably reduces the computational complexity, it does not assure the frames found to be similar are the most similar. Discarding the phase in the kernel function renders the method shift-invariant but it also eliminates the unitary property of the Fourier Transform, i.e. Parseval's theorem does not hold anymore and thus Euclidean distances can be different. Therefore, when measuring the Euclidean distance between two frames in the magnitude specmurt, a large distance certainly indicates dissimilarity but a small distance does not assure a close match in the time-frequency domain (for example, ma-

ajor and minor chords can get confused). In practice, this means that kernel function will find k close frames but not necessarily the k most similar ones.

To overcome this drawback while maintaining the complexity reduction, we here propose to use the acceleration technique as a pruning method. Instead of selecting k -NN in the kernel function, we select a larger fixed value ($k + P$) to increase the pool of close frames, where P is the number of additional *pruning* time frames. We then perform the specmurt analysis described above to find their optimal shift. At this point, one can retrieve these $(k + P)$ frames in the time-frequency representation and shift them by their corresponding amount. This means, we now have a narrowed down shifted version of the input magnitude, and so we can apply the baseline method to select the k -NN from the $(k + P)$ frames presented. The overall complexity remains the same.

4.3.2 Empirical evaluation

In order to evaluate the proposed shift-invariant kernel influence on the overall framework, we here present a simple and well defined experiment that will validate the potential of the proposed method. The given mixtures correspond to a single musical instrument corrupted once by a burst-like interference (e.g. the input magnitude shown in Figure 10) and the task is to remove the interference, or reduce its impact. The interferences are those that typically occur in live or studio recording scenarios: cough, chair drag sound, door slam and sound of object being dropped. We retrieved example recordings of each from freesound¹.

Since we are mainly interested in finding out how the different kernels behave on recordings where the musical source is not repeated in time, we have created a synthetic dataset where the repeated and not repeated passages are known. In this way we are able to compare the proposed method against the baseline in both cases. We created five different melodies (monophonic) and five different chord pro-

¹ <https://www.freesound.org/>

gressions, to simulate short studio takes, and synthesized these with 12 different instruments using the high quality Native Instruments Komplete Ultimate suite. We then created test recordings by overlaying the recordings with the interferences at 12 dB SNR, placing the interferences at two different locations: on a repeated musical segment and on a not repeated one, resulting on 960 tracks between 5 and 10 seconds each. While a more realistic dataset might better indicate the performance of the methods, we chose this setup to investigate exactly those cases where the individual methods might differ the most.

To quantitatively compare the separation quality of our proposed extension to the baseline, we used the BSS Eval toolbox 3.0 [132] to calculate the Signal to Distortion Ratio (SDR). We used the CQT implementation described in [114], setting the parameters to 24 bins per octave, gamma value of 20, minimum frequency of 27.5Hz and the maximum frequency being half of the sampling frequency (44.1kHz). For all methods, we hold the parameter k of the k -NN kernel function fixed at 300 frames. Note however, as it will be further discussed in Section 4.5, that k can and should be adjusted to the level of repetitiveness in the input recordings – the higher the repetitiveness, the more all methods benefit from higher k . For our proposed method, we fixed the number of shifts Δ to 48 (covering 4 octaves in total). In the acceleration+pruning method, the parameter P is set to be $2k$. We ignored the first coefficient in the specmurt representation as we expect it to mainly capture the broadband components. In addition, we assume the location of the interference in the mixture is known and thus we only process the frames affected and measure the SDR on those segments. The kernel function for all methods is applied to the remainder of the frames.

The results with respect to the normalised SDR (NSDR) are given in Table 2, for both melody and chord progressions, on repeated and not repeated musical segments. As expected, the KAM baseline behaves poorly when there is no repetition, especially for melodies, which resembles the common scenario in popular songs where the source of interest is consistently repeating on the same pattern of unwanted sources. The NSDR value for the baseline for the non-repeated chords

	Melody		Chords	
	Repeated	Not repeated	Repeated	Not repeated
Baseline	3.31	-2.40	4.11	1.26
Prop. 1	4.61	3.87	4.11	2.11
Prop. 2	5.06	4.22	4.03	1.09
Prop. 3	5.23	4.36	4.52	2.10

Table 2: NSDR values for the baseline, the basic shift-invariant proposed method (Prop. 1) and the acceleration technique without pruning (Prop.2) and with pruning from an initial pool of twice the amount of k frames (Prop. 3). Parameters: $k = 300$, SNR = 12dB, $\Delta = 48$. See Section 2.5 for the interpretation of NSDR.

shows that, even though the chord is not repeated, some of its notes might, which can already be exploited by the method.

However, the basic shift-invariant method (Prop. 01) clearly outperforms the baseline in those not repeated cases demonstrating standard KAM’s limitations in such cases. In addition, it matches or improves the performance of the baseline on repeated segments, which suggests the proposed kernel function benefits from the shifting operation presenting the overlaying unwanted sources as clear outliers (affected by a shift in frequency). The basic shift-invariant method remains computationally expensive. However, the proposed methods based on specmurt analysis with (Prop. 03) and without pruning (Prop. 02) are effective in the melody scenario by even improving upon Prop. 01’s separation performance. This can be explained by the fact that the accelerated variants can find arbitrary shifts, while the shift in Prop. 01 is limited to reduce the computational time. In the chord progressions scenario, the low results of Prop. 02 confirms limitations in using the specmurt domain and justifies its use as a pre-selector for the pruning method Prop.03.

The results clearly demonstrate the inability of the baseline kernel to reconstruct non-repeated musical events and confirms the efficacy of the proposed shift-invariant kernel for such cases. Moreover, even

for repeated segments, the increase of the pool of similar frames led to improvements over standard KAM.

4.4 A MACHINE LEARNING APPROACH TO KAM FOR LOW SNR

Even though the extensions presented above already partially alleviate KAM framework's limitations, they still rely on the assumption that the energy in frames is dominated by the target source. Therefore, while KAM is free of the need for suitable training data, it might fail to find similar frames if the signal-to-noise ratio is low. In particular, in the presence of sudden and loud interferences, existing KAM approaches are likely to fail.

To overcome such frailty we propose to combine the strengths of two algorithmic families by adopting a machine learning approach to inform KAM on the target source when it is overlaid by a more powerful unwanted source. The idea is to include a preliminary step to "learn" about the unwanted powerful source so we can then identify it and reduce its impact. By doing so, one can then safely use KAM assuming the target source to be dominant.

To improve the baseline frame-wise k-NN search in KAM and make the kernel function more invariant against the overpowering signal, we propose to build a first initial signal model using training data. Most of the state-of-the-art machine learning methods are either based on NMF variants or DNNs, as previously discussed in Section 2.2. While supervised NMF might not be as precise as a DNN approach to yield a high quality signal model necessary for source separation, it might be discriminative enough to obtain an initial signal model for the music at a lower computational cost. Therefore we propose to employ NMF to create a preliminary signal model which can later be used to design an adaptive, interference-resilient kernel for KAM.

More precisely, we let the user provide keywords to describe the unwanted source (e.g. "cough") and retrieve corresponding recordings (i.e. training data) from the publicly available freesound² archive.

² <https://www.freesound.org/>

Concatenating these recordings into a single file, we compute its magnitude spectrogram $\bar{X}_N \in \mathbb{R}^{F \times T_N}$ as well as an NMF factorization into a basis (or feature) matrix $W_N \in \mathbb{R}_+^{F \times R_1}$ and an activation matrix $H \in \mathbb{R}_+^{R_1 \times T_N}$ such that

$$\bar{X}_N \approx W_N \cdot H \quad (32)$$

where the only parameter is the NMF rank R_1 . To visualise the influence of such parameter, in Figure 12 we can find four learnt coughs dictionaries W_N for different NMF rank values R_1 . If the rank is too high, e.g. the dictionary down right with 50 spectral basis, one can easily see that the signal is overfitted and the abundance of detail dilutes the original broadband nature of the interference. On the other hand, if the rank is too small, e.g. dictionary on the top left, there is no room for an actual spectral characterisation and everything is condensed into the reduced number of spectral bases, that end up containing so much information it could correspond to any sound. In consequence, for this particular example, either a rank 10 or rank 20 would suit better the application.

The NMF factorisation itself is accomplished using the well-known Lee-Seung NMF updates for the generalized Kullback-Leibler divergence D_{KL} [76]; i.e. we minimize $D_{KL}(\bar{X}_N, W_N \cdot H)$ over non-negative matrices W_N and H iteratively following:

$$\begin{aligned} H &\leftarrow H \odot \frac{W_N^\top \cdot \mathcal{R}_N}{W_N^\top \cdot I} \quad \text{and} \\ W_N &\leftarrow W_N \odot \frac{\mathcal{R}_N \cdot H^\top}{I \cdot H^\top} \quad \text{with} \end{aligned} \quad (33)$$

$$\mathcal{R}_N := \frac{\bar{X}_N}{W_N \cdot H}$$

where I is the all-one matrix of size $(F \times T)$. After convergence, either set by a cost threshold or a maximum of iterations, the columns of W_N contain templates reflecting the spectral properties of the unwanted overpowering interference signal.

Now we can employ NMF to model our input spectrogram $\bar{X} \in \mathbb{R}^{F \times T}$ using a combination of interference templates, $W_N \in \mathbb{R}_+^{F \times R_1}$,

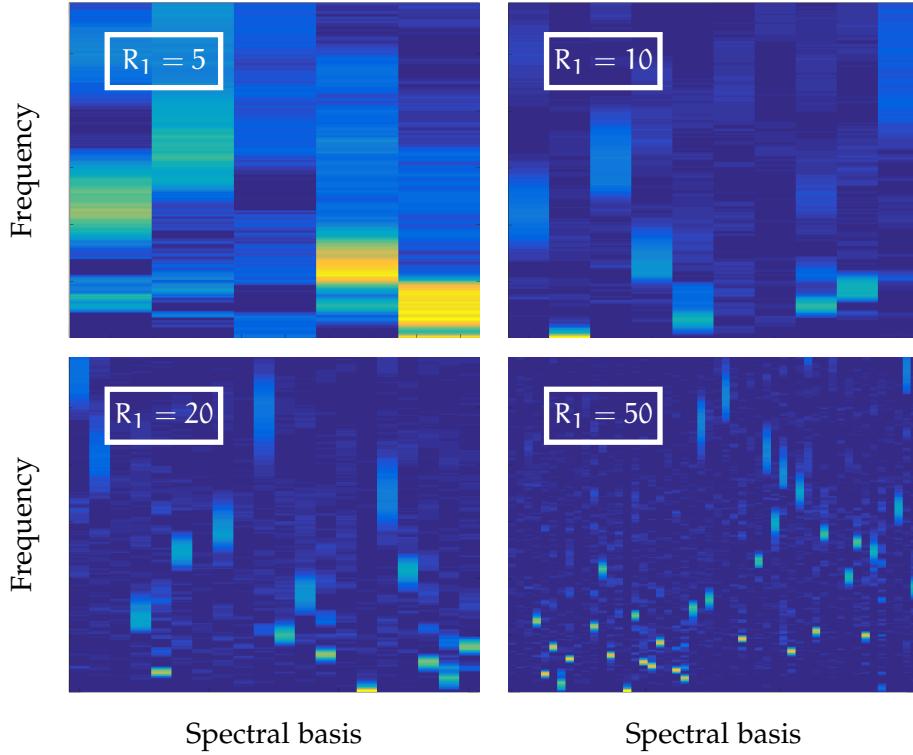


Figure 12: NMF learnt coughs dictionaries W_N (Equation 4.4) for different rank R_1 values, to visualise the trade-off between them, going from not being discriminating enough ($R_1 = 5$) to overfitting ($R_1 = 50$).

and music templates, $W_S \in \mathbb{R}_+^{F \times R_2}$, introducing another rank parameter R_2 , such that:

$$\bar{X} \approx W_N \cdot H_N + W_S \cdot H_S \quad (34)$$

where the interference templates W_N can be kept fixed and we only need to learn the target templates W_S , often referred to as *semi-supervised NMF*.

More precisely, we now minimize the function $D_{KL}(\bar{X}, W_N \cdot H_N + W_S \cdot H_S)$ over H_N , W_S and H_S (i.e. we fix W_N). In this case, the update rules are:

$$\begin{aligned}
H_N &\leftarrow H_N \odot \frac{W_N^\top \cdot \mathcal{R}}{W_N^\top \cdot I}, \\
H_S &\leftarrow H_S \odot \frac{W_S^\top \cdot \mathcal{R}}{W_S^\top \cdot I} \quad \text{and} \\
W_S &\leftarrow W_S \odot \frac{\mathcal{R} \cdot H_S^\top}{I \cdot H_S^\top} \quad \text{with} \\
\mathcal{R} &:= \frac{\bar{X}}{W_N \cdot H_N + W_S \cdot H_S}
\end{aligned} \tag{35}$$

After convergence, the rows of H_N capture the activations of the unwanted source templates, while $W_S \cdot H_S$ yields an approximation of the magnitude spectrogram of the target source. Using these two interpretations, we employ these results for two different purposes.

First, we use H_N to identify where the unwanted source is and in this way, in contrast to existing KAM approaches, we can filter the signal only where needed. To this end, we sum the values in H_N in each frame to obtain a single curve indicating the unwanted source activity, as higher values indicate activation of the learnt templates in W_N and so presence of the unwanted source (or interference). In order to be robust against the possible leftover activations of other sources in H_N , one could assume higher activation values to be those corresponding to our unwanted source and use a simple threshold to determine the unwanted source activity. However, such method could lead to false negatives if the unwanted source is at some point less powerful, for example, in the decay part of the sound. To alleviate such issue, one can implement a smoother thresholding by assuming that, if the unwanted source is likely to be active at some point in time, it is also likely that it is active in the following time step, and so incorporating previous states to the decision of the present one.

To this purpose, instead of using a fixed energy threshold to determine whether the unwanted source is active or not, we propose to decode its activity using a hidden Markov model (HMM) as a flex-

ible and dynamic "smart" threshold inspired by its use in different audio applications [59, 94, 110]. The HMM smooths the normalised energy H_N taking into account the surrounding frames of those who had energy above a certain threshold, to make a decision using previous states information. The HMM smoothing implemented has two parameters that will vary the probability of a energy value to correspond to the unwanted source state (1) or not (0). The first parameter corresponds to a threshold to which each energy value is referred to, and the second parameter is the cost of changing state representing the probability of a state to remain as it is. For ease of visualisation, here we represent both parameters by a set token "smart" threshold TH_{HMM} . The result is a binary frame-wise state vector $n_{HMM} \in \mathbb{B}^{1 \times T}$ (1 if the unwanted source is present, 0 otherwise):

$$n_{HMM} = \begin{cases} 1 & \text{if } \frac{\sum_f H_N}{\max(\sum_f H_N)} > TH_{HMM} \\ 0 & \text{otherwise} \end{cases} \quad (36)$$

The parameters of the HMM, detection threshold and cost of changing state, should be adjusted to favour recall over precision in the detection, because in this situation, it is preferred to process a frame with little or no interference (i.e. false positive) than missing an interference frame entirely (i.e. false negative), leaving it untouched in the final estimate of the target source.

Additionally, using the activations for the free templates H_S , we can reconstruct an initial rough estimate for the music, where the unwanted source is strongly reduced as most of the corresponding energy is already captured by the interference templates. Based on this initial model, we identify for each frame affected by the interference a list of similar frames, which are then used within the KAM framework to produce the final output. In other words, having an initial target source model not only serves to improve the query for similar frames, but also helps with its reconstruction offering an alternative to nearest neighbours overlaid by the unwanted source. In this way, the pool of nearest neighbours is increased, overcasting the unwanted source as an outlier and ultimately improving the separation as previously discussed in Section 4.3.

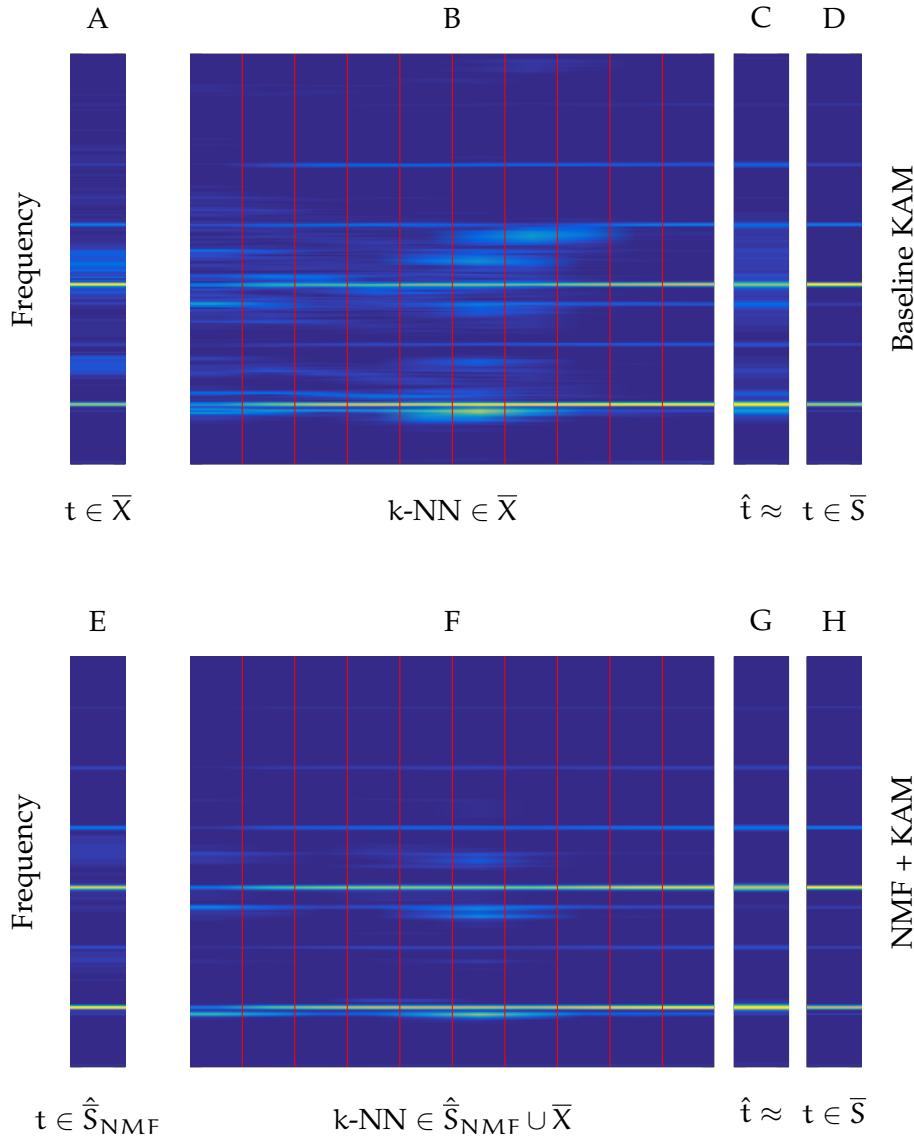


Figure 13: Individual steps in our proposed method, (A)-(D) using standard KAM, (E)-(H) using our proposed extension: (A,E) current frame used for similarity search, (B,F) first 10 closest frames found, (C,G) estimated frame and (D,H) ideal clean frame.

More precisely, we can improve the k-NN search in KAM kernel by replacing the input spectrogram \bar{X} containing the interference with the NMF approximation for the target source $\hat{\bar{S}}_{\text{NMF}} := W_S \cdot H_S$. If the interference is sparse enough, one can use the k-NN frames without interference (informed by n_{HMM}) of the input magnitude \bar{X} and only the NMF alternative $\hat{\bar{S}}_{\text{NMF}}$ for those k-NN overlaid by the unwanted source, for the reconstruction of the target source through median filtering. In other words, the frame wise k-NN search takes place on

the NMF target source approximation, such that the frame \tilde{t} is in $\mathcal{I}(f, t)$ if \tilde{t} is amongst the k most similar frames to t , where similarity is measured as:

$$\sum_f (\hat{S}_{NMF}(f, t) - \hat{S}_{NMF}(f, \tilde{t}))^2 \quad (37)$$

However, the k -NN will be selected either from the input magnitude or the NMF approximation depending if they are overlaid by the unwanted target source or not, reconstructing the magnitude target source as follows:

$$\hat{S}(f, t) := \text{median} \left(\begin{cases} \bar{X}(f, \tilde{t}) & \text{if } n_{HMM}(\tilde{t}) = 0 \\ \hat{S}_{NMF}(f, \tilde{t}) & \text{if } n_{HMM}(\tilde{t}) = 1 \end{cases} \mid (f, \tilde{t}) \in \mathcal{I}(f, t) \right) \quad (38)$$

In practice, this just means that the pool of good nearest neighbour candidates increases, as the space of time frames doubles going from $t \in \bar{X}$ to $t \in \bar{X} \cup \hat{S}_{NMF}$, and as mentioned above, this benefits the median operator.

The resulting improvement is clearly visible in Figure 13. Replacing the \bar{X} -frame (Figure 13 A) with the corresponding \hat{S}_{NMF} -frame (Figure 13 E) in the similarity search, we see that the frames selected as nearest neighbours (Figure 13 F) are much closer to the actual target (Figure 13 D = Figure 13 H). The median filter can then remove remaining noise robustly, bringing the result (Figure 13 G) much closer to the target (Figure 13 H).

Since the semi-supervised NMF is here providing two services, detection of the unwanted source and creation of a preliminary target source model, the choice of the rank R_2 determining the number of free templates to learn W_S has some trade-offs to be considered. The higher the rank R_2 , the more spectral bases are available for the target source definition but also, the higher the chances are of overfitting such source and redefining the interference. In other words, if the number of spectral basis for the target source estimation in W_S is too high, some will be used to over-define the unwanted source, cause the interference estimate to lose information and therefore affect its detection. On the other hand, if the number of spectral bases in W_S is low, those intended for the interference W_N could be activated to characterise the unwanted source. In consequence, the rank chosen

should be a good compromise between allowing enough spectral basis to characterise the target source and avoiding over-representing the unwanted source in the target estimate (false positives are preferred over false negative activations).

An alternative to alleviate the semi-supervised NMF rank trade-off is to do two rounds with two different ranks. One with a larger rank focusing on yielding a good target source model and another with a smaller rank favouring false positives on the interference detection step. Even though this alternative does add computation time, it is a quite simple solution to guarantee the best rank parameter choice for each case.

A further problem we observed is that the kernel \mathcal{J} was often changing considerably between frames in the sense that often $(f, \tilde{t}) \in \mathcal{J}(f, t)$ would not imply $(f, \tilde{t} + 1) \in \mathcal{J}(f, t + 1)$. Without this property, however, we observed a slight pitch jitter in the magnitude across frames after median filtering, which was audible in the final time domain signal. To further temporally stabilize the kernel function, we incorporate a temporal context into the similarity search following Section 4.2. Further, we found that filtering $\hat{\mathbf{S}}_{\text{NMF}}$ slightly in frequency direction before the k-NN search using a small Gaussian kernel additionally improved the results, as it makes the similarity search invariant to small changes in the fundamental frequency of harmonic sounds.

4.4.1 *Empirical evaluation*

Similarly to Section 4.3.2, here we evaluate the proposed method for a non-stationary burst-like interference reduction task. We chose interferences that typically occur in a live or studio scenario including cough sounds, door slams, sounds of objects of different material being dropped, chair-drag sounds as well as audience screams as seen in Table 3. Similar to Section 4.3.2 and [32], we retrieved recordings of interferences from *freesound*. This way, the method does not rely on the availability of non-public training data and is easily extended to other types of interferences. However, this also implies that the quality and number of training samples can vary, and thus explains why,

in our case, each interference has a different amount of training data, ranging from 10 scream samples to 40 coughs tracks (see Table 3). The separation quality is expected to improve as the number of tracks in the training data increases.

The music dataset contains 58 instrumental mono stems from the freely available multitrack MedleyDB dataset [9], covering 23 different instruments ranging from guitar, violin, piano over to bass, trombone or flute as seen on Table 3. The choice for pitched instruments without vocals can be explained by the choice of interferences used in this study.

Since separating sources of the same nature is exceptionally challenging and all of the interference sounds present a dominant broadband aspect, it would be ambitious and out of scope for this evaluation to attempt the separation of such interferences and broadband instruments. Therefore, we choose pitched instrumental recordings as our target sources in order to evaluate the approach presented on a well defined problem that would not introduce added difficulties unrelated to the methodology under test. In addition and for similar reasons, vocal recordings have not been considered in this study as they represent a major separation challenge on their own.

We created test recordings by making artificial linear mixes of the instrumental mono stems and the test interference recordings independent of the training data and of each other (other acoustic conditions). In order to achieve a controlled mix of instrumental and interference levels, all tracks were normalised to a specific RMS energy. Then three interferences are added to the music at different SNR, measured on the segment where the interference is active. The final mix is a 30s long monaural recording with three different sounds of the same kind interfering at different times at a certain SNR.

We evaluated the proposed method on the resulting 290 mixtures (58 instrumental stems times 5 types of interferences), measuring the separation performance using the BSS Eval toolbox [132], obtaining a SDR and SIR for each mixture separation. To indicate the improvement over the raw music-interference mix, we employ the normalized SDR/SIR (NSDR/NSIR) as in [81]. This way, we can account for the

Music	Interferences		Test	Training tracks
		Test		
Banjo	1	Coughs	3	41
Basoon	1	Door Slams	3	15
Bass	1	Screams (male and female)	3	10
Double Bass	4	Chair sounds	3	10
Cello	2	Object drop sound	3	10
Clarinet	5			
Dizi	2			
Erhu	2			
Flute	4			
French Horn	2			
Guitar Acoustic	5			
Guzheng	2			
Mandolin	1			
Oboe	3			
Piano	4			
Saxophone	5			
Trombone	1			
Trumpet	5			
Viola	2			
Violin	2			
Yangqin	2			
Zhongruan	1			

Table 3: Summary of the dataset tracks used for this evaluation.

	NSDR			NSIR		
	0dB	-3dB	-6dB	0dB	-3dB	-6dB
Prop.	6.78	4.76	2.52	16.79	15.30	13.69
NMF	4.76	3.16	1.13	13.15	14.40	15.62

Table 4: Comparison of our method with supervised NMF for different SNR values.

fact that a separation at a low SNR is more difficult than at a high SNR, making results for different SNRs more comparable.

Here we have chosen supervised-NMF to represent the current state-of-the-art method (following Equations (34) and (34)) to quantitatively compare its separation performance to the proposed method. In order to obtain a competitive baseline, we use the same learned dictionary for both methods and we also optimise the NMF rank with a parameter sweep. Tables 4 and 5 show the overall results, averaged across all NSDR/ NSIR values of every mixture, for our proposed method as well as for the semi-supervised NMF approach. Comparing the results, our proposed method yields a higher separation quality than the NMF-based method not only for a 0dB SNR mixture, but also for mixtures where the interference is 3dB and 6dB above the instrumental RMS energy. Overall, we obtain an improvement between 1.4 and 2.0dB, which from a relative point of view is quite considerable.

In order to measure the influence of the individual components of our proposed method, Table 5 shows results separately for several variations of our method. In addition, to provide another angle on the results and focus on the positions where the interferences actually happen, we evaluated the separation performance by averaging across the three segments in the mix where the interference is active in Table 5, and so the resulting NSDR scores are not directly comparable to Table 4.

Starting with a baseline frame-wise k-NN kernel KAM approach, as described in [41], *Variant V1* adds the NMF interference detection step introduced in Table 5. The high NSDR shows the interference

	NSDR	NSIR
V1: Standard KAM + NMF Interference Detection	7.09	13.62
V2: V1 + NMF-based Kernel Similarity + Temporal Context	7.92	15.48
V3: V2 + Adaptive Frame Selection + Smoothing (Proposed Method)	8.84	14.53

Table 5: Influence of individual KAM extensions on the separation result (interference at 0dB SNR; separation evaluated on the segments affected by an interference).

was successfully identified and reduced. *Variant V2* further adds the improved similarity measure of our proposed method, where similarity is measured based on a rough NMF estimate of the signal. Additionally, the frame-wise similarity search used in standard KAM (and *Variant V1*) is modified to account for the local temporal context in V2 as introduced in Section 4.2. The higher NSDR shows that the temporal context stabilizes not only the kernel but also the results. In this context, it is important to remark that our test signal are only 30 seconds long – for longer signals with additional repetitions of musical patterns, we would expect even higher improvements in NSDR. Overall, both extensions improve the capability of our method to better identify and select similar frames and thus to increase the performance of the median filtering step.

Variant V3 is an extension of *Variant V2* incorporating the smoothing filter and the adaptive frame selection, which replaces frames in the median filter in which an inference was detected with the corresponding frames from the NMF estimate. As shown in Table 5, both extensions further improve the NSDR over variant *Variant V2*. However, the NSIR values are sometimes lower – in our experiments, we found this to be a side effect of the smoothing filter, which slightly blurs the spectrum, leading to a tendency of leaving more residual energy in the output. However, overall, these results show that each of our proposed extensions measurably improves the separation quality.

4.5 HOW DO WE PICK k?

A successful separation of the target source relies largely on the interference actually being outliers within the selection of the k closest frames as discussed in Section 4.3.

We want to make sure that the k -NN frames have a similar contribution of the target source with no or different overlaying unwanted interference (refer back to Figure 9). However, there are also frames matching both wanted and unwanted sources which will then be very likely to be selected as near neighbours. Those frames are unhelpful for the median filtering but since the breakdown point of the median operator is of 50% of outliers (vocals), the method is robust to the unwanted repetitions up to a point. This robustness is closely related to the number of nearest neighbours we choose, i.e. the parameter k .

There seems to be little or no indication on a method to find the optimal parameter k in the literature [41, 82, 106]. In [106] the authors introduce three other parameters to set boundaries for the choice of k . However, no indication was found on how to actually fix any of those parameters, including k .

To our knowledge, there are currently two broad approaches to setting k : perceptual assessment or evaluation metric optimisation (later also referred to as *parameter sweep*). In the first approach one simply listens to the estimates for different k values and adjusts the parameter to the best sounding setting. This is the preferred method to set k when there is a reduced number of songs to be processed.

The second approach relies on a metric, typically the Signal to Distortion Ratio (SDR), comparing the estimated sound sources with the ground truth. One will then set k to obtain the best metric result. In practice, this means a parameter sweep for different k values (similar to the temporal context optimisation shown in Figure 7), for which no indication was found on how to pick. Essentially, one makes an educated choice of k values and picks the one yielding the best metric performance.

Since the evaluation metrics are often costly to compute, the number of k values one can try out is reduced, which comes with the risk of missing the optimal k value. In addition, for the same reason it is common practice to perform the parameter sweep over the entire testing dataset and pick the k value maximising the *mean* SDR, instead of finding the optimal value for each track, disregarding individual (potentially crucial) characteristics such as the song length.

However, when dealing with large datasets, perceptual assessment of the results can be very time consuming. Therefore, the second method involving a *parameter sweep* is more popular even though the common metric used for the optimisation (i.e. SDR) is known to be a proxy for perceptual quality and its precision has been criticised [18].

Overall one could argue that the parameter sweep approach has a number of disadvantages, primarily linked to the optimisation through a performance metric. Firstly, the separation performance metrics usually require to have ground truth separate tracks available, which is not always possible in an application scenario. Further, the commonly used separation performance metrics are computationally expensive [132], limiting the parameter sweep to a reduced number of values in a time constraint situation. In addition, optimising k using an overall performance metric does not assure the best value for all songs in the dataset. Moreover, fixing the k sweep values leaves no room to inform the optimisation with the track's individual properties.

Ideally we would like to be able to automatically pick k in an unsupervised way for each track separately, taking into account the nature of the song and thus finding a tailored value for k assuring a successful separation. We would also like to do this without having to perform multiple runs of source separation and discarding all but one of them. In the following part of the thesis we will present how to do so by taking a different perspective : graphs.

4.6 SUMMARY

The success of separation in KAM heavily depends on the ability of the kernel to identify similar frames in the presence of overlaying sources. Using just a squared Euclidean distance between entire frames, this notion of similarity, however, can be quite limited. Firstly, this kernel choice assumes that the target source repeats in both time and frequency, meaning the position of partials and other objects cannot change. Secondly, it assumes the energy in the time frames of the given mixture to be dominated by the target source.

In this chapter we have investigated the common scenarios where these conditions fail to be true and consequently we proposed several extensions to the KAM framework to improve its flexibility under such circumstances, summarised in Table 6. We introduce a temporal context in the kernel function which temporally stabilises the target source estimate improving the separation performance in Section 4.2. We then present in Section 4.3 a shift-invariant kernel function introducing a degree of freedom to the similarity search in the frequency direction. Finally we incorporate a machine learning approach to overcome a low SNR scenario where the target source is overpowered by an unwanted interference, by combining for the first time NMF and KAM. We close this chapter by pointing out the lack of discussion in the literature about the sole parameter of the KAM framework, k , and we point towards the following Part of the thesis where we will expand on this subject and present a new perspective on it based on graphs.

KAM limitations	Prop. Extensions	Application	Introduced Parameters
similarity metric	→ temporal context in kernel	Vocal separation	C
not repeating source	→ Shift-Invariant KAM	Interference reduction	Δ, α, P
low SNR conditions	→ NMF + KAM	Interference reduction	R_1, R_2, TH_{HMM}
how to set k ?	→ ?	?	?

Table 6: Summary of KAM limitations and the extensions proposed in this thesis

Part III

BEYOND THE MAGNITUDE DOMAIN : GRAPHS

The required graph theory concepts are introduced to then explore the graph structure within KAM and propose an automatic parameter optimisation method. A novel representation for audio is introduced based on visibility graphs, a powerful tool for time series analysis that can now, for the first time, be computed on-line with a proposed method.

5

EXPLOITING THE GRAPH STRUCTURE WITHIN KAM

From the discussion in the previous Chapter 4, we can state that the performance of KAM fundamentally depends on the nearest neighbours. The size of the set (i.e. the number of selected nearest neighbours) is determined by the parameter k and there appears to be a gap in the literature on its influence and optimisation, even though one could expect the choice of k to be crucial for a successful separation as discussed in Section 4.5. Here we investigate the influence of the parameter k in a vocal separation task and we further propose a novel method for its automatic optimisation, based on consideration of the proximity graph, which is lightweight and needs no prior training.

We start off by introducing the notion of graphs in Section 5.1 and the concepts and definitions from graph theory necessary to understand the following sections. We will then explore the graph structure within KAM in Section 5.2 and propose a novel computationally inexpensive method to optimise the parameter k based on graph theory statistics in Section 5.3. We will further analyse and discuss the impact of this parameter through an experimental evaluation and validate the proposed method in such scenario in Section 5.3.1.

5.1 PRELIMINARIES: WHAT IS A GRAPH?

NOTE: Here we will go through some definitions necessary for the following sections, serving as a brief overview of some graph theory notions. For a full and detailed explanation of graph theory concepts please refer to [4, 71, 136].

A graph G is a triple consisting of a non-empty finite set $V(G)$ of elements called *nodes* (or *vertices*), a finite set $E(G)$ of *edges* (or *arcs*) and a relation associating each edge with its *endpoints* (i.e. two nodes not necessarily distinct). One can represent an edge as a pair of nodes and regard the nodes as the fundamental element to form a graph. The graph is said to be *finite* if both the node and edge sets are finite, and *null* if they are both empty. In a finite graph G , the total number of nodes is referred to as the *order* of the graph $n(G)$ and the total number of edges as the *size* of the graph $e(G)$. A subset G' of G such that $V(G') \subseteq V(G)$ and $E(G') \subseteq E(G)$ is called a *subgraph* and G is said to contain G' . Usually we illustrate a node as a circle (or point) and an edge as a curve connecting two nodes, as showed in Figure 14.

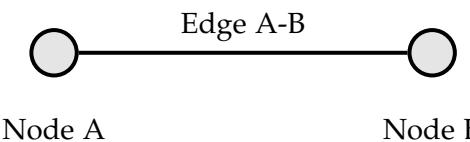


Figure 14: Simple graph composed by two nodes and an edge joining them.

Here we will focus on *simple* graphs, which contain at most one edge between two distinct nodes. This means nodes cannot be connected to themselves (i.e. loops) and that there are no multiple edges between two nodes. Two nodes connected by an edge are said to be *neighbours* or *adjacent*. Therefore, we can define the *adjacency matrix* $A(G)$ of the simple graph G with a node set $V(G) = v_1, v_2, \dots, v_N$ of size $n(G)$, as a $n(G) \times n(G)$ matrix where each entry a_{ij} contains the number of edges in G with endpoints v_i, v_j . In a simple graph this matrix will be binary, as the nodes are either connected by one edge (i.e. $a_{ij} = 1$) or not connected (i.e. $a_{ij} = 0$), with 0s in the diagonal as there are no loops. The adjacency matrix is also symmetric.

A *path* is a simple graph whose nodes can be ordered so that only consecutive nodes are connected. If the number of nodes and edges is equal and only sequential nodes are connected, the path is called a *cycle*. In other words, a cycle is a path that begins and ends on the same node. The red and green edges in Figure 15 are paths examples, but only the red path in Figure 15.c is a cycle. A path or a cycle is said to be *Hamiltonian* if all the nodes in the graph are included once

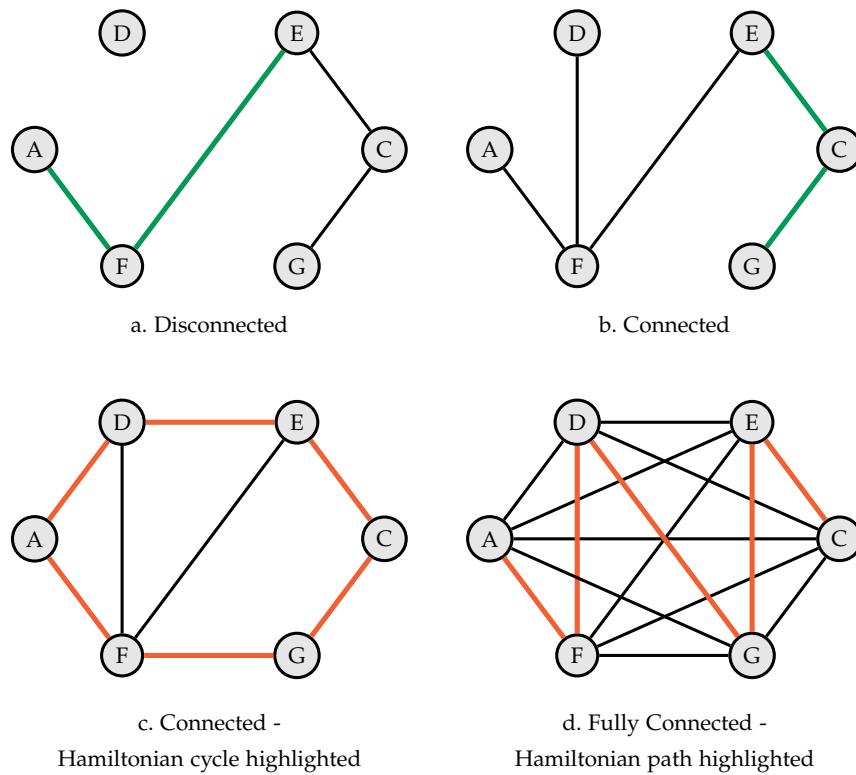


Figure 15: Types of graphs classified by their connectivity. Hamiltonian paths examples in red.

and only once. Both paths in red in Figure 15 are Hamiltonian and therefore, graphs in Figure 15.c and d are said to be Hamiltonian too.

If all the nodes are connected to each other like in Figure 15.d, the graph is said to be *fully connected* or *complete* and are always Hamiltonian. However, if the graph is said to be just *connected*, it means there is a path from any node to any other node in the graph but it does not guarantee a direct connection between nodes as in a complete graph, as shown in Figure 15.b and c. If the graph is not connected it is simply said to be *disconnected* (Figure 15.a).

We can further classify graphs depending on the nature of their edges. If the set of edges is a set of ordered pairs of distinct nodes, the graph is called a *digraph* and it is said to be *directed*. On the other hand, if the set of edges is simply a set of unordered pairs of nodes, the graph is said to be *undirected*. As seen in Figure 16, unlike an undirected graph (Figure 16.a) where the edges are simply represented by straight lines, the edges of a directed graph (Figure 16.b) are repre-

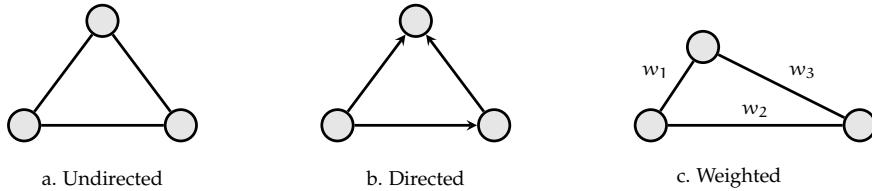


Figure 16: Types of graphs classified by the types of edges.

sented by arrows with a *tail* and a *head* to indicate the order. We say the edge leaves the node at the tail and enters the node at the head. In a directed graph the connection between nodes is therefore not necessarily symmetric, whereas the connections of undirected graphs are symmetric and can be regarded as a special class of directed graphs (i.e. symmetric digraphs).

If the connections between nodes are not all equal, weight values are usually allocated to the edges and the graph is referred to as *weighted* (e.g. Figure 16.c). Such weights are often used to record transition probabilities, and in the case of Markov chains, the probabilities of all the edges leaving a node sum up to 1.

The set of edges incident to a node v is referred to as the *degree* of such node $d(v)$ and its value is represented as κ ¹. In Figure 17, there is a sample graph with 7 nodes. The degree of node F is highlighted and equal to 3 as it is connected to three nodes (A,B and G) and therefore has 3 incident edges. The degree of a node can be deduced by summing the entries of the adjacency matrix in either the row or column for that node. As it can be seen in In Figure 17, by summing the columns one can obtain a degree vector $\vec{\kappa}$ containing the degree value for every node in the graph, and therefore retrieve the value 3 for node F highlighted in green. For directed graphs, one can define the *out-degree* $d^+(v)$ as the number of edges with tail in node v and the *in-degree* $d^-(v)$ as the number of edges with head in node v . The degree in such a case will be the sum of the out and in degree.

The κ -*occurrence* of the graph captures the number of nodes n_κ with a degree κ and can be represented by the histogram of $\vec{\kappa}$ as shown

¹ The standard notation for the degree of a graph is the letter k . However, in this dissertation k refers to the number of nearest neighbours in the kernel. Therefore, we have opted to use the greek letter kappa κ to refer to the degree of a graph.

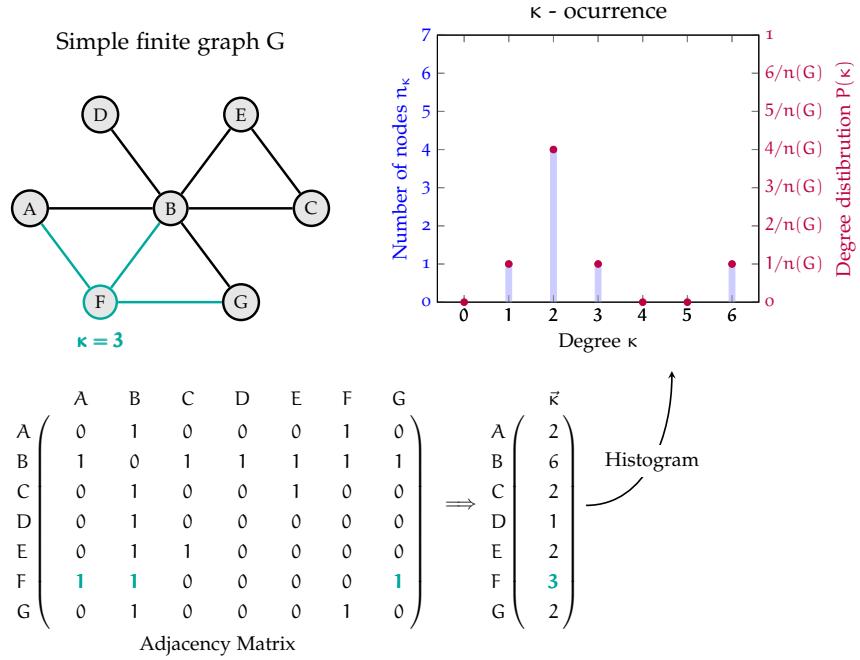


Figure 17: Visualisation of an a graph example G , its adjacency matrix, degree κ , degree vector $\vec{\kappa}$, κ -occurrence and degree distribution.

in Figure 17. The degree distribution can be understood as the normalised κ -occurrence by the order of the graph $n(G)$, representing the probability of a node to have a degree κ in such a graph (red dots in Figure 17).

5.2 PROPERTIES OF THE k-NN GRAPH

We can exploit the graph structure within KAM by defining a proximity k -NN graph, where the nearest neighbour relationships are represented as a directed graph D . In such a graph, the nodes represent the time frames of the given mixture magnitude spectrogram $\bar{X} \in \mathbb{R}_+^{F \times T}$, so the order of the graph will always be the total number of time frames T . Every frame has k nearest neighbours and so each node has k edges leading outward to its nearest neighbours nodes. The size of the graph will therefore always be $k \times T$.

In Figure 18 we find an example of a k -NN graph for $k = 3$ of order 7. Note that if frame i is a neighbour of frame j , the reverse is not necessarily true. For example, the nearest neighbours of node

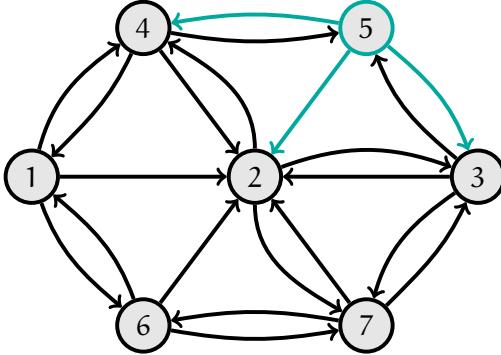


Figure 18: Illustration of a k-NN graph for $k = 3$, where every node represents a time frame $t \forall t \in [1, T]$ of the mixture magnitude spectrogram $\bar{X} \in \mathbb{R}_+^{F \times T}$. In this example the total number of frames T of the supposed time-frequency representation encoded in this k-NN graph would be 7 (i.e. the total number of nodes).

5 in Figure 18 are the time frames 2, 3 and 4; however, 5 is only a nearest neighbour to 3 and 4 and not for 2. At extreme settings, if $k = 0$ (no nearest neighbours) then the graph has no arcs and thus no useful structure, while if $k = T$ the graph is fully connected (all nodes have the same degree $\kappa = T$) and likewise exhibits no useful structure. What are desirable characteristics for a k-NN graph to be used in KAM?

Unlike many problems defined on a graph, in KAM we do not wish our graph to take on a simple structure such as well-separated clusters: instead, we want all frames to have connections to frames which are similar according to the current source kernel, but dissimilar in terms of the other sources as discussed in detail in Section 4.3. It is not clear how these structural considerations can be best quantified numerically, though such structure would have some impact on summary statistics considered in graph theory.

Consider a vocal separation scenario where the target source is the accompaniment music. A set of frames containing a background musical phrase which is repeated often is expected to form a densely connected component in the graph. On the other hand, the frames containing sparsely-present and variable vocal energy would be expected to have arcs pointing to that densely connected component but few arcs pointing back out to them. Therefore, the number of incoming arcs (i.e. in-degree) would be unevenly distributed across the

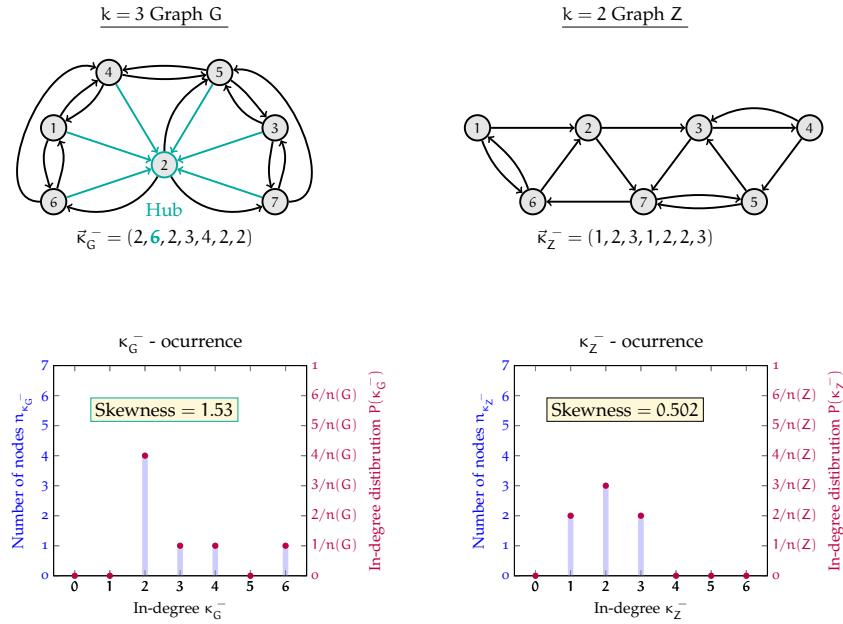


Figure 19: Illustration of how hubness relates to skewness. Two example k-NN graphs G (left) and Z (right) for $k = 3$ and $k = 2$ respectively of order 7. Graph G node 2 is a hub, therefore the hubness of G is greater than that of graph Z which contains no hub, as it can be seen from the shape of the κ^- -occurrence distribution and its skewness value.

nodes, directly as a result of the observed signal properties which one assumes in KAM.

One way to analyse such properties in graph theory is the concept of *hubs*, which are nodes with an unusually high in-degree [105]. For example, node 2 in Figure 18 represents a hub in the given graph as its in-degree, $d^-(2) = 6$, is considerably higher than the rest of nodes. This has been of particular influence in social network theory as researchers studied effects such as "small world" phenomena, which can have important effects such as the speed at which news or illness spreads through a social network [64, 135].

For a given graph, one can define summary statistics which reflect the general presence of hubs. One referred to as the *hubness* is simply the skewness of the κ -occurrence statistics. In this case, the skewness of the distribution of the in-degrees κ^- of nodes in the graph. Here, the κ^- of a frame corresponds to the number of times that frame is amongst the k nearest neighbours, and the hubness is therefore the

skewness of the distribution of all frames' k-occurrence. Figure 19 shows two k-NN graphs examples, G (left) and Z (right), and their corresponding κ^- -occurrence distributions on the second row along side the skewness value for each one of them.

In a k-NN graph we assign a fixed number of edges, and so the average in-degree is always k; however if the graph contains strong hubs, as graph G in Figure 19, then the skewness of the in-degree will be high. This can be seen in the second row of Figure 19, where the skewness of G is clearly higher than that of Z, indicating higher hubness.

5.3 HOW TO PICK k

For the family of methods of KAM of study in this thesis (refer to Section 3.2), in a vocal separation application, it is clear that a graph with relatively *high* hubness should typically be one which has appropriate structure. As discussed in Section 4.5, we typically have very little *a priori* guidance over what value of k to choose, so it is advantageous that, for each track separately, we can iterate over a selection of possible k, inspect graph statistics such as hubness for the graphs thus produced, and select k which produces the optimal statistics. Therefore, we here propose to select the k producing the maximum hubness of the associated k-NN graph.

However, in a situation where we vary k, the hubness h will vary even in the null case of a randomly-constructed graph. (This can be seen in the extreme cases: for $k = 0$ or $k = T$ the graph is symmetric and the hubness is 0, whereas for other k it can be non-zero.) A standard null model can be generated by selecting k neighbours for each frame purely at random. The distribution of k-occurrences in this null model follows a binomial distribution with parameters T and k/T , leading to an expression for the expected hubness as:

$$h_{\text{null}} = (1 - 2k/T)/\sqrt{k(1 - k/T)} \quad (39)$$

We can thus define a normalised hubness h_{norm} statistic as the ‘excess’ hubness, i.e. the raw observed hubness minus the hubness expected under the null model,

$$h_{\text{norm}} = h - h_{\text{null}} \quad (40)$$

which should then be less biased than the raw hubness in selecting k . We can think of the normalised hubness as a *de-noised* version of the raw hubness.

However, the above null model is one of the simplest random graphs and we found the scale of its hubness statistic to differ from that of the raw hubness, larger than in the simple null model. In practice, graphs constructed from high-dimensional similarity measures do not behave strictly in that fashion, and it is an ongoing research topic to model how k -NN graphs behave in general [105]. Here we simply rescale both raw and null model hubness through max normalisation, taking the maxima across the sweep of k settings.

Using the maximum hubness as a metric to choose k has numerous advantages:

1. It does not require any ground truth information
2. k is optimised per track as a pre-processing step before the separation actually takes place
3. It is quick to compute so we can sweep through a lot of different k values, so we can have a finer optimisation
4. The hubness has been demonstrated to have perceptual relevance for song similarity in music recommendation [48], suggesting that it reflects properties of the nearest neighbour graph that have impact on its applied use. However, it has not been used for frame selection in KAM and so that is to be explored here.

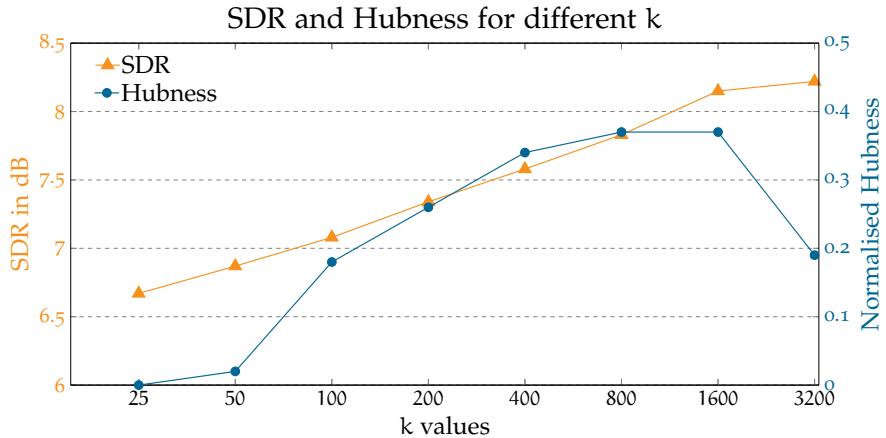


Figure 20: Median SDR and hubness across all songs in Test DSD100 for different fixed k values

5.3.1 Empirical evaluation

To evaluate the proposed method, we quantitatively compare it against the standard parameter sweep for setting k described in Section 4.5 for a vocal separation task. We chose to follow the vocal separation method [41] described in Chapter 4 with FFT size of 4096 and hop size of 1024 samples, as it represents a baseline instance of the larger KAM framework.

To encourage reproducibility, we use the publicly available Test Demixing Secrets Dataset (DSD100) [83], containing 50 full length songs of diverse genres sampled at 44.1 kHz. Since the kernel implemented relies on musical repetition, we evaluated our proposed method on full length songs to ensure as much sound material as possible for KAM's source reconstruction. However, the literature only offers some indication on k values for 30 second segments. We therefore use a broad range of fix k values for the standard parameter sweep (refer to Section 4.5), letting $k \in \{0, 25, 50, 100, 200, 400, 800, 1600, 3200\}$, and a finer percent increase sweep for the computationally inexpensive proposed method taking the song length into account, letting $k \in \{(0.001, 0.011, 0.021, 0.031, \dots, 0.45) \times T\}$ where T is the total number of time frames in the song.

Following common practice in the field, we employ the Signal to Distortion Ratio (SDR) in the BSS Eval toolbox 3.0 [132] as the quan-

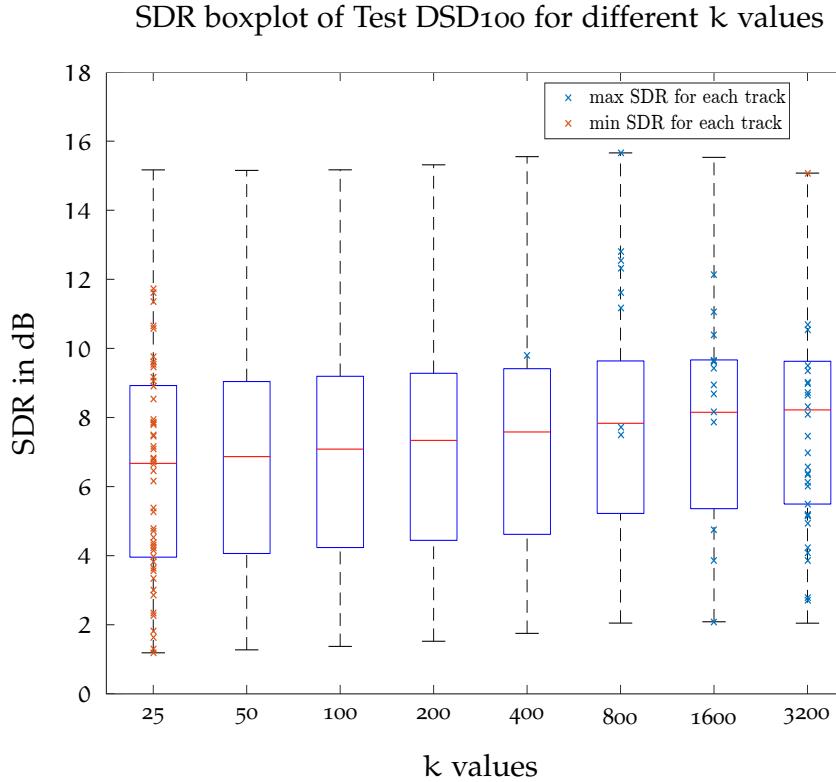


Figure 21: SDR Boxplot of every song in the Test DSD100 dataset for different k values. The maximum and minimum SDR obtained for each song are marked in blue and orange respectively, showing a general trend of higher separation performance with increasing k value.

titative indicator of the separation performance. Therefore, we would expect to observe a positive correlation between SDR and hubness for different k values. Due to the diversity of styles in the dataset, one could also expect an improvement in the overall separation performance (and so SDR) by using a tailored k for each song following the proposed method.

Note that with the proposed method we can afford checking for more k values and for every track, as computing a parameter sweep maximising the hubness is more efficient than one maximising the SDR. This is mainly because the hubness analysis is done in the time-frequency domain unlike the SDR, which requires a time domain estimate and ground truth. Therefore, in the proposed method one only needs to calculate the similarity within frames once and then produce the necessary k-NN graphs for hubness analysis without having to re-

SDR boxplot of k values for Test DSD100

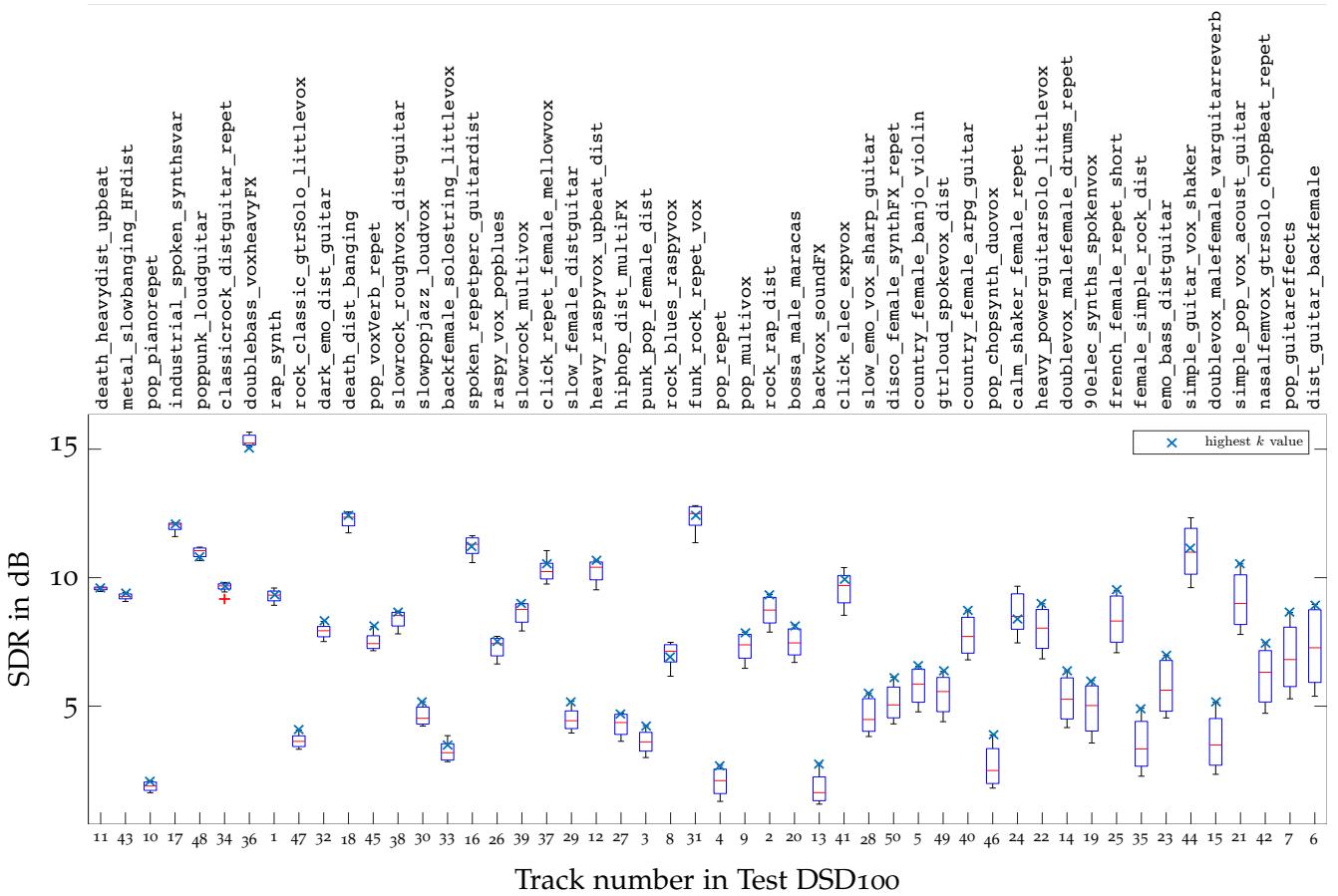


Figure 22: SDR Boxplot of different k value for each song in the Test DSD100 dataset, briefly informally described in the top axis and sorted in ascending variance order. The SDR obtained with the maximum k value of 3200 for each song is marked in blue showing the different behaviour between songs.

calculate the similarity, nor perform the separation through masking, nor convert to time domain for every k. In addition, one can implement a standard hubness measure in $O(T \log(T))$, where the current SDR implementations² appear to be at least quadratic in the length of the signal, adding significant computation cost to the parameter optimisation process.

According to the standard method to fix k, one would pick the value with a higher overall SDR, here (Fig. 20) is the highest k of 3200

² Example SDR implementation can be found here: <https://github.com/sigsep/sigsep-mus-eval>

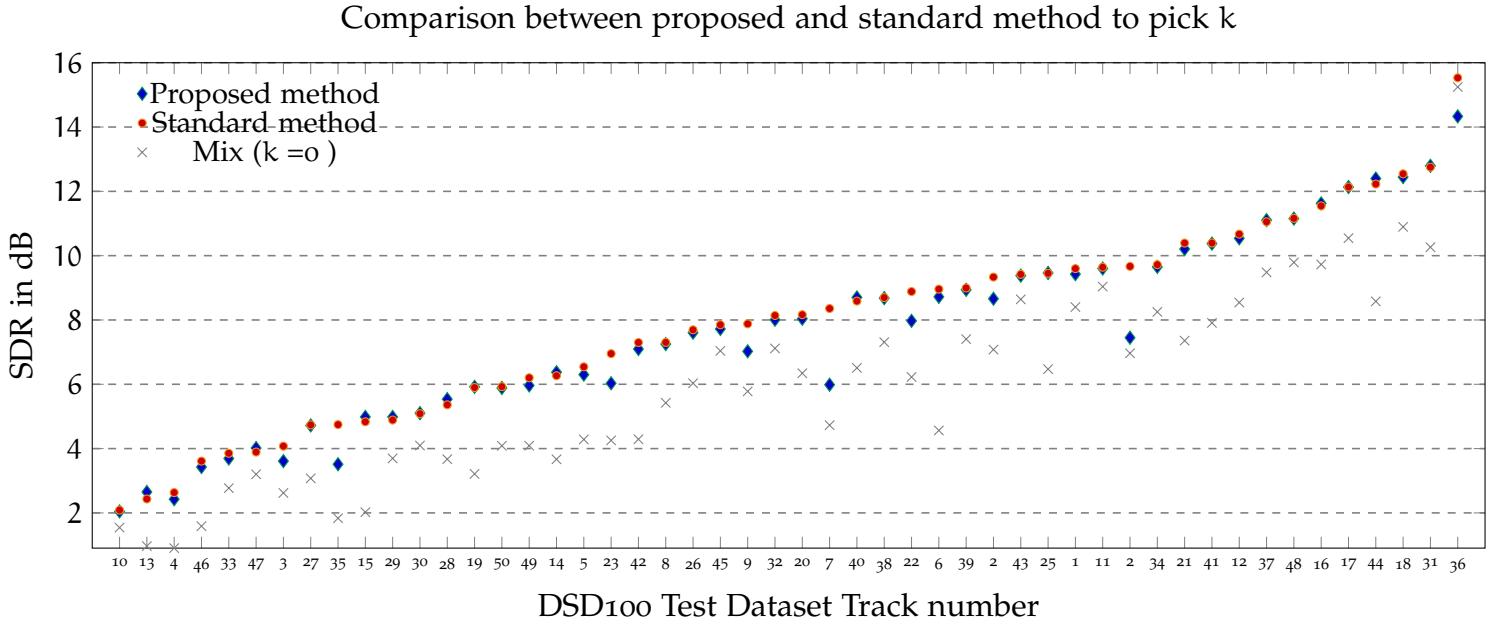


Figure 23: SDR values for each song in the Test DSD100 dataset sorted in ascending order, using the optimal k issued from the standard and proposed method, in comparison to the SDR of the raw mixture (i.e. $k=0$).

frames. Alternatively, the positive correlation between the hubness and SDR seen in Fig. 20 suggests the hubness to indicate the optimal k value for a successful separation.

Moreover, the similarity between boxplots in Fig. 21 for different k values suggests there might not be an unique k that maximises the SDR of every song in the dataset. However, the markers in Fig. 21 show differently as most of the songs obtained a higher SDR with the highest k value. This behaviour comes as a surprise taking into account the dataset's disparity. Most tracks were expected to peak in SDR for lower k values than 3200 frames, which seems to be so many frames that it should generally surpass the 50% of outliers breaking point of the median operator. The abundance of highly repetitive songs could potentially explain how such a large k could be successful, although the literature indicates the SDR may not be a reliable metric of the actual separation performance [18].

Fig. 22 offers a different perspective on the individual song behaviour which should shed some light on the above dilemma. As expected, very repetitive songs such as track 45, 4 or 50, achieve a

higher SDR with highest k values. However, it is also the case for unconventional pop songs such as 43 or 17, where the variance in SDR is extremely low (less than 0.05). For such cases the separation may not have been successful, but Fig. 23 shows otherwise as the median SDR is above the mixture's SDR (equivalent $k = 0$). Further, the overall SDR variance is surprisingly low, with a median of 1.4dB potential SDR increase by changing k (maximum of 3.57dB and minimum of 0.17dB). With such a low potential SDR improvement, one might wonder if k actually matters at all or again, if the SDR is failing to capture the actual separation performance.

The majority of cases where different values of k induce substantial changes in SDR correspond to popular songs with a classic pop musical set-up and repeating musical structures (Fig. 22)—the ideal scenario for the implemented KAM vocal separation as described in [41]. One could therefore infer that a track sensitive to different k values (i.e. higher SDR variance), fulfills KAM requirements for a successful source separation. Track 44 presents an excellent example as it has a high SDR median and high SDR variance (2.72 dB of potential SDR improvement). However, most of the tracks in the dataset fail to present such characteristics, introducing a question regarding the flexibility and adaptability of the implemented KAM for vocal separation.

Songs which fulfill KAM ideal requirements for vocal separation (sensitive to k or highly repetitive) are expected to present higher SDR values than more complex songs. However, Fig. 22 does not present such logic, which makes one further wonder if the SDR is indeed an adequate choice for performance evaluation or whether the kernel function should be further refined. Since recent research [18] seems to also doubt the choice of SDR as an evaluation metric, we can, for now, safely attribute the observed behaviour to the doubtful correlation between perceptual and SDR performance evaluation. In consequence, and to promote future perceptual studies on the matter, the source code is available online along side a web demo³ that allows the user to listen to vocal separation results for a selected k .

³ Available at <https://github.com/delialia/kam-demo>

KAM limitations	Prop. Extensions	Application	Introduced Parameters
similarity metric	→ temporal context in kernel	Vocal separation	C
not repeating source	→ Shift-Invariant KAM	Interference reduction	Δ, α, P
low SNR conditions	→ NMF + KAM	Interference reduction	R_1, R_2, TH_{HMM}
how to set k?	→ k-NN graph hubness indicator	Vocal separation	none

Table 7: Summary of KAM limitations and the extensions proposed in this thesis

Nevertheless, Fig. 23 shows the proposed method can be used as substitute to the current technique for fixing k. Both methods present similar results in most cases and although the proposed one presents lower SDR for some songs, it seems a small trade-off for a considerable decrease in computation time (in our experiments 1000 times faster than the standard method).

5.4 SUMMARY

In this chapter we introduced the notion of graphs and their properties and we showed how can it be beneficial to the KAM framework. In particular, we propose to exploit the natural graph structure of KAM by defining a k-NN graph from the mixture spectrogram. Such a graph representation offers a new perspective to the model as its topology is essentially determined by the sole parameter in the framework, k. Therefore, we propose to use the statistics of the k-NN graph as an indication on how to set k, starting a new discussion in the field which currently lacks a standard methodology to optimise such a parameter.

We concentrated on a vocal separation task where the background music is said to be highly repetitive in comparison to the vocal activity. In such an application, we propose to use the skewness of the degree distribution ("hubness") to inform the choice of k. We expect frames containing only background music to be popular nearest neighbours and therefore, appear as hubs in the corresponding k-NN graph. Therefore we propose to set k to the value maximising the hubness of the k-NN graph. The proposed method to automatically fix k is the first of its kind and allows for a quick computation at track level

(instead of the standard way of fixing k for a whole dataset) without the need of any training data. We can now therefore complete Table 7 and add the hubness as an indicator to pick an optimal k value.

In the following chapter we will continue to explore the use of graph theory in audio and question the use of time-frequency magnitude domain as the standard representation for audio tasks such as source separation. We take a step further by proposing a new graph-based representation for audio with particularly useful properties for audio related tasks.

6

INTRODUCING VISIBILITY GRAPHS FOR AUDIO

In Chapter 4, we discussed the importance of finding the appropriate nearest neighbours for the subset of methods in KAM employing the k-NN kernel function. Success in finding the suitable candidates ultimately relies on how we model the target source and on the model’s ability to differentiate the target source from unwanted overlaying sources. For example, in Section 4.4 we show how the basic KAM source model fails to identify the target source in presence of an overpowering overlaying interference, and how we can improve the model by introducing a preliminary supervised step generating an alternative interference-reduced signal representation.

Both of these models, as well as most of the models in the literature (refer to Chapter 1), operate in the time-frequency domain. Here we challenge that standard and question whether a change in the chosen data representation would bring a new perspective helpful to overcome the challenges imposed by the current methodologies. Following the discussion in Chapter 5, one may wonder what is the room for improvement left within the time-frequency domain and could argue that only a shift of paradigm would result in a considerable advance.

In Chapter 5 we introduced the notion of graphs and exploited the graph structure in KAM to automatically set its sole parameter k . Here we continue exploring the benefit of introducing graph theory to audio tasks by considering novel graph-based representations for audio signals.

Graphs are a tool of growing interest in the signal processing community for data representation and analysis. Their structure offers a new perspective, often unveiling non trivial properties on the data they represent. In the last decade, several methods to map time series into graphs have been proposed under the hypothesis that ap-

ropriate graph representations can preserve information from the original time series while dealing with non-linearity and multi-scale issues typical of complex signals [27, 97]. This line of research represents a bridge between nonlinear signal analysis and complex network theory, and has been successfully applied to extract meaningful information from a variety of different systems in physics [67, 119], finance [38, 66, 93], engineering [129], and neuroscience [13, 14].

The most notable algorithms to construct a graph from an ordered sequence of data points are either based on correlation [10, 87, 139], recurrence [25, 26, 37], dependence [79, 88], or visibility [68]. The visibility algorithms proposed by Lacasa et al. [68, 85] are amongst the most popular as they provide a deterministic and non-parametric mapping of a time series preserving full information of its linear and non-linear correlations. Such visibility algorithms can also effectively deal with non-stationary signals and are deemed computationally efficient. In consequence, visibility graphs have found numerous applications in diverse fields including image processing [61, 65], number theory [69], finance [47, 66], and neuroscience [112].

Every node in such a graph represents a datum of the time series, and two nodes are connected if they fulfil the "visibility" criteria analogous to the visibility between points on a landscape. The visibility between data will only depend on their relative height and location, creating a graph structure capturing the links between data. The success of this simple visibility mapping is partly due to its powerful properties. Visibility graphs preserve characteristics of the time series such as periodicity [98], and are invariant to several transformations of the time series, such as vertical and horizontal rescaling. Visibility graphs and their properties are fully defined in Section 6.1.

In Section 6.2 we introduce visibility graphs applied to magnitude spectra. Such a graph will preserve all the properties of visibility graphs as its construction remains the same. Therefore, as with time series, the visibility graph of spectra may reveal hidden structures in the signal not apparent in the magnitude domain. In particular, we focus on musical audio signals, and we propose the spectral visibility graph degree as a novel representation for audio analysis. We demon-

strate its use for robust similarity measures of harmonic signals in the empirical evaluation in Section 6.2.1.

The final Section 6.3 of this chapter addresses the computation of visibility graphs, which in its straightforward mode, presents a worst case time complexity quadratic in the length of series. Even though faster algorithms have been proposed, the current existing methods to compute visibility graphs are off-line algorithms and so require all data points in time series to be available before the graph is constructed. Such a rigid computation represents a major shortcoming limiting the real-world applications of visibility graphs. As an alternative, we present in Section 6.3, to the best of our knowledge, the first algorithm capable of computing visibility graphs on-line, whilst maintaining the efficiency of state-of-the-art algorithms.

6.1 VISIBILITY GRAPHS

A visibility graph is obtained from an ordered sequence of values by associating each datum to a node and connecting two nodes with an edge if the corresponding data points are visible from each other. A point a is visible from the point b if one can draw a straight line from a to b without passing underneath any intermediate points. In this thesis we will consider visibility as a symmetric relation, and so the resulting visibility graphs are undirected.

The natural visibility criterion (NV) allows the visibility line between a and b to take any slope [68], whereas the horizontal visibility criterion (HV) is restricted to horizontal lines [85], as shown in Figure 24.a and b respectively. More precisely, given a time series

$$y = g(t) \quad (41)$$

of length n , two points (t_a, y_a) and (t_b, y_b) are said to be naturally visible if every intermediate point (t_c, y_c) , such that $t_a < t_c < t_b$, fulfills the following simple geometrical criterion:

$$y_c < y_a + (y_b - y_a) \frac{t_c - t_a}{t_b - t_a} \quad (42)$$

This natural visibility criterion will therefore establish the connections between nodes in the resulting natural visibility graph (NVg).

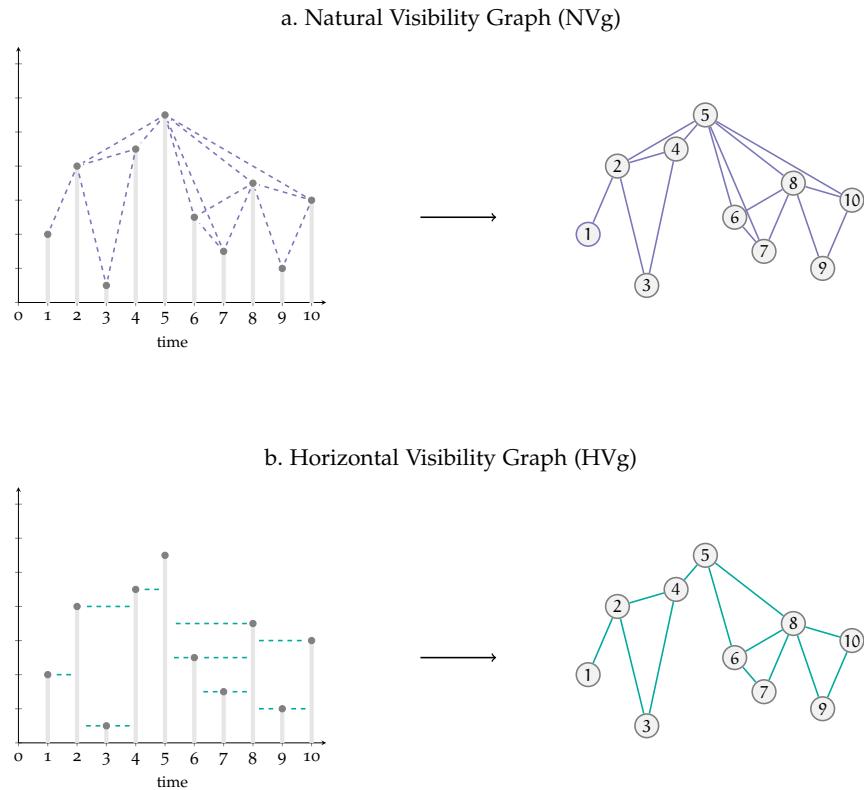


Figure 24: Visibility graphs map time series into complex networks following a natural (a) or horizontal (b) visibility geometrical criterion.

One can likewise map a time series into a horizontal visibility graph (HVG) where two points (t_a, y_a) and (t_b, y_b) are said to be horizontally visible if :

$$y_a, y_b > y_c \quad \forall c \text{ such that } t_a < t_c < t_b \quad (43)$$

In short, two points in a given time series are said to "see" each other if one can draw a line joining them without intercepting any intermediate data height. For natural visibility, the line can take any slope and for horizontal visibility, the line joining two data points must have zero slope. Figure 24 shows both the natural and horizontal visibility criteria at work on an arbitrary time series. Notice that horizontal visibility is a more stringent criterion than natural visibility, meaning that if two points are horizontally visible then they are also trivially visible when using the natural visibility criterion. Consequently, the horizontal visibility graph of a time series is always a sub-graph of the natural visibility graph associated to the same time series.

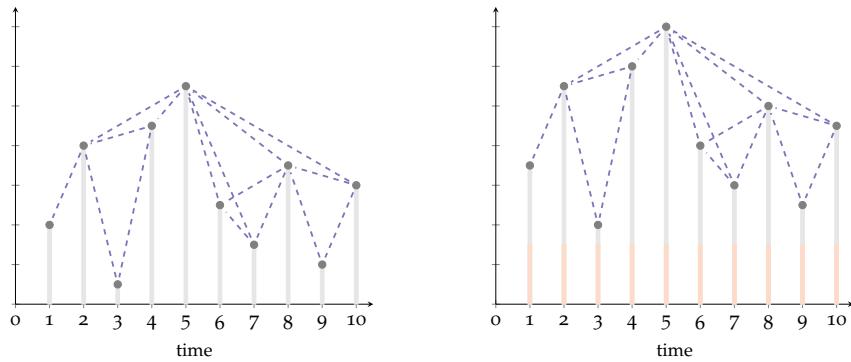


Figure 25: The visibility between points is invariant to several transformations including the vertical translation shown here.

From the definition of visibility it immediately follows that, for a set visibility criterion, the visibility graph associated to a given time series is unique. Moreover, any two subsequent data points of the time series are always connected by an edge, thus visibility graphs are connected and Hamiltonian [85]. In addition, visibility graphs are invariant to re-scaling on both horizontal and vertical axes (i.e., the first point on either side of a node i remains visible from i no matter how far apart they are), and invariant to vertical and horizontal translations (i.e., only the relative values of point determine visibility relations). Figure 25 illustrates a vertical translation of the data points in a sample time series. One may observe how the visibility between points remains the same after the transformation.

6.2 SPECTRAL VISIBILITY GRAPHS

Inspired by the invariant properties of visibility graphs, we propose to employ such a mapping for magnitude spectra, introducing visibility graphs to spectral analysis. We define the spectral visibility graph (SVg) of a given magnitude spectrum $\bar{z} = g(f)$ of $z \in \mathbb{C}^F$, where f denotes frequency and F is the total number of frequency bins, following the construction of visibility graphs for time series. From now onwards we will focus on the natural visibility (as it also contains the horizontal visibility) and refer to it simply as visibility.

Every frequency bin corresponds to a node, unlike previous audio graph-based representations where nodes are associated to feature vectors [89] or time frames (Section 5.2). Two nodes are connected if the associated frequency bins (f_a, \bar{z}_a) and (f_b, \bar{z}_b) see each other, fulfilling the visibility criterion

$$\bar{z}_c < \bar{z}_a + (\bar{z}_b - \bar{z}_a) \frac{f_c - f_a}{f_b - f_a} \quad (44)$$

where (f_c, \bar{z}_c) is every intermediate frequency bin such that $f_a < f_c < f_b$.

Following the definitions in Section 5.1, we can further consider the corresponding adjacency matrix A and find the degree vector $\vec{\kappa}$ and degree distribution vector \vec{p} , such that for every frequency bin $f = 1, 2, \dots, F$ in the spectrum its degree corresponds to

$$\kappa_f = \sum_{j=1}^F A(f, j) \quad (45)$$

Both the construction of the degree and degree distribution vector can be visualised in Figure 26.

Similarly to the degree vector of the visibility graph of time series, the SVg degree vector remains invariant under several transformations of the spectrum, including vertical and horizontal translation as well as vertical and horizontal rescaling. In the case of audio signals, a horizontal rescaling of the spectrum would correspond to a change in pitch and a vertical translation to the presence of uniform broadband noise. Being resilient to such transformations is a major advantage in the audio analysis of applications where the relation between peaks (i.e. harmonic content) is the subject of interest. Therefore, we propose the SVg degree vector $\vec{\kappa}$ as an alternative representation for magnitude spectra \bar{z} , both represented in Figure 26.

Taking a step further, let $Y \in \mathbb{C}^{F \times T}$ be the spectrogram of an audio time signal y , and \bar{Y} its magnitude, where F is the number of frequency bins and T the number of time frames. Here, the proposed representation $K \in \mathbb{N}^{F \times T}$ will take a matrix form such that every column $t = 1, 2, \dots, T$ will correspond to the degree vector $\vec{\kappa}_t$ of the visibility graph of frame t of \bar{Y} (Figure 27). More precisely, taking $A_t \in \mathbb{B}^{F \times F}$ as the visibility graph's adjacency matrix of the time

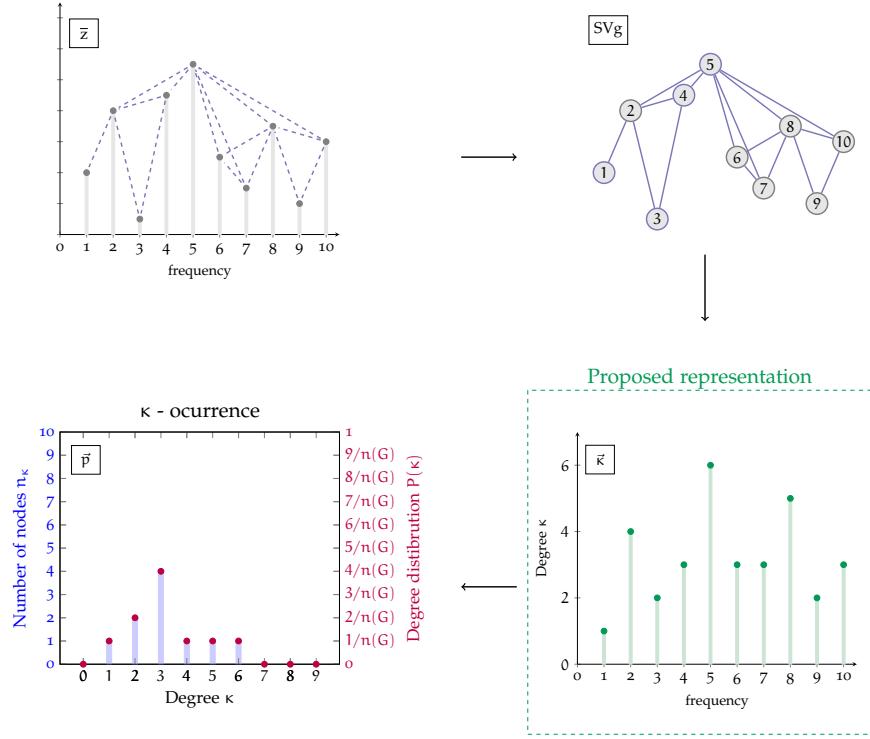


Figure 26: Proposed representation, the visibility graph degree vector, as an alternative to magnitude spectra.

frame's magnitude spectra t (i.e. column) of the spectrogram \bar{Y} , we define the degree matrix K associated to \bar{Y} such that:

$$K(f, t) = \sum_{j=1}^F A_t(f, j) \quad (46)$$

where $f = 1, 2, \dots, F$ and $t = 1, 2, \dots, T$. We propose to use K as an alternative representation to the spectrogram \bar{Y} .

Even though spectral peaks tend to take high values in the proposed representation, their prominence will depend on their surroundings. In other words, peaks close to each other will have less height than sparse ones, such as the harmonics of a musical note. Looking at Figure 26, one may notice how the height at position 4 lost pertinence in the degree domain, going from being the second maxima to being equal to lesser heights (6, 7 and 10); explained by its proximity to the maximum peak in 5. On the other hand, the heights at position 2 and 8 (equally spaced from the maximum) surrounded by smaller heights, gained relevance in the degree domain. Therefore, one can think the transformation into the degree domain, and so into the proposed rep-

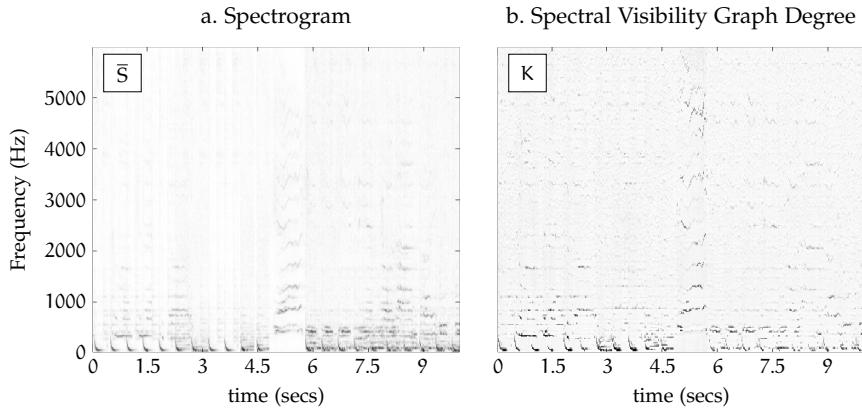


Figure 27: The spectrogram (a) and the proposed representation (b) of 10 seconds of track 51 of the dataset DSD100. Both representations are normalised by their own maximum and compressed by a factor of 0.6. The spectral visibility graph degree enhances the harmonics components of the signal.

resentation, as a sort of compression enhancing sparse peaks (i.e. harmonics) apparent in Figure 27.

As an audio analysis tool, the structure and properties of the proposed mapping directly relate to harmonic content analysis, and so we propose to examine the common case where both harmonic and broadband events overlap. In such a scenario, the harmonic energy in the spectrum will remain prominent up to a certain SNR, taking the harmonic event as the signal of interest and the broadband as noise. If the broadband event overpowers the harmonic content, it will overcast the harmonic contribution in the magnitude spectrum, complicating the analysis of its harmonic content.

As one can observe in previous chapters, a common task in audio analysis is the search for similar harmonic content between spectra (e.g. time frames in a spectrogram). In the presence of powerful additive broadband noise, most distance metrics fail to recognise the similarity of the harmonic content as they treat all spectral energy as equivalent. Such a scenario relates to a vertical translation of the magnitude spectrum and so the harmonic event spectrum with and without additive broadband noise should present a comparable visibility graph and degree vector. Therefore, unlike in the magnitude spectrum (e.g. Figure 27.a), the harmonic peaks in the proposed rep-

resentation (e.g. Figure 27.b) will remain salient in presence of additive broadband events, and so, one can now use standard distance metrics (e.g. l_1 or l_2 norm) to reliably measure harmonic similarity. Hence we propose the SVg degree as a novel domain for robust harmonic similarity measure in audio signals.

6.2.1 Empirical evaluation

To evaluate the proposed representation of audio signals for harmonic similarity measure we performed two experiments, one with synthesised data and a second one with real musical recordings. In both experiments the task is to find the correct nearest neighbour of a given harmonic event. We use three different representations of the audio signals: the magnitude spectrum (i.e. \bar{z}), the SVg degree (i.e. $\vec{\kappa}$) and the SVg degree distribution (i.e. \vec{p}).

Our proposed representation is the SVg degree; however, we included the degree distribution (cf. Figure 26) in the experiments as it has an additional pitch invariance that could benefit the task (i.e. the absolute location of peaks information is ignored). Since our goal is to compare these representations, we employ simple distance metrics (i.e. Euclidean and cosine) to conclude on which representation is more appropriate for harmonic similarity measurements. We use the mean reciprocal rank (MRR) as the evaluation metric, as we know before hand which is the one and only correct nearest neighbour, similarly to other audio query tasks with an unique correct target such as cover song retrieval [5, 33].

We can define the Euclidean distance d_E of two vectors $p = (p_1, p_2, \dots, p_N)$ and $q = (q_1, q_2, \dots, q_N)$ of the same size N as

$$d_E = \sqrt{\sum_{i=1}^N (q_i - p_i)^2} \quad (47)$$

and the cosine distance d_C between the same two vectors as normalised metric following:

$$d_C = \frac{\sum_{i=1}^N q_i p_i}{\sqrt{\sum_{i=1}^N q_i^2} \sqrt{\sum_{i=1}^N p_i^2}} \quad (48)$$

Therefore we expect the Euclidean distance to be more sensitive to similarity in peak values and the cosine distance to be more sensitive to the positioning of the peaks (i.e. the pattern).

For high frequency resolution spectra, the basic computation of visibility graphs ¹ is not ideal ($O(n^2)$). Therefore, here we used a significantly faster alternative to compute visibility graphs based on a "Divide & Conquer" approach ($O(n \log n)$) [70] that will be further discussed in Section 6.3. Python source code for our implementation and experiments is freely available online.²

In the first experiment we used part of the synthesised data from the experiment in Section 4.4.1: 12 synthesised instruments with the same midi file of 14 notes (A2 to G4) sampled at 44100Hz. Each instrument signal was divided into the distinct midi notes and then individually transformed into the magnitude frequency domain with a Fourier transform of size 16384, "clean" spectra. Only the first 2000 bins of the magnitude spectra were kept for the rest of the analysis.

Random normal noise was then added to the note signals at different SNR values and the result transformed to the frequency domain, "noisy" spectra. The pair-wise distances between all spectra, both clean and noisy, were then computed and sorted in ascending order. For every clean track, the rank of its noisy version was found and used to compute the MRR. This procedure is repeated for the spectral visibility graph degree representation as well as for the degree distribution. We can define the MRR as the mean of the noisy version's rank across all Q clean spectra queries, such that:

$$MRR = \frac{1}{Q} \sum_{i=1}^Q \frac{1}{rank_i} \quad (49)$$

¹ Original visibility graphs Fortran 90/95 implementation can be found at <http://www.maths.qmul.ac.uk/~lacasa/Software.html>

² Available at <https://github.com/delalia/vgspectra>

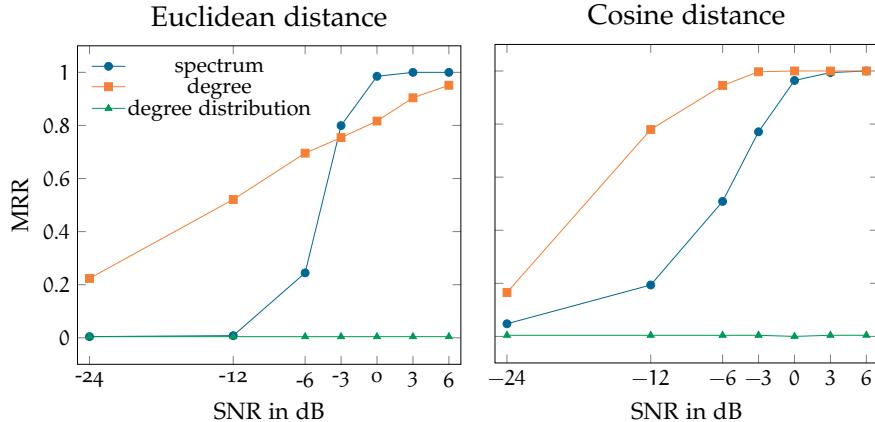


Figure 28: The average mean-reciprocal-rank (MRR) amongst all notes of all instruments in experiment 01: 12 synthesised instruments playing 14 notes, clean and with additive random noise. Pair-wise similarity between all signals in the frequency magnitude, degree and degree distribution domain. The clean notes act as query and the expected closest neighbour is their noisy version.

The average MRR across all notes of all instruments for different SNR is plotted in Figure 28. As expected the proposed method (orange solid line) achieves best results when the SNR is low. Moreover, since the location of the peaks are better preserved in the proposed representation, it always achieves best results whilst using the cosine distance metric. However we see a small dip in performance relative to the raw spectrum, using the Euclidean distance in the higher SNR cases. This can be explained by the bigger difference in value between the degree peaks of the clean and noisy signals than in the spectrum case. Even though the peaks remain prominent in the noisy case, the number of nodes the "peak node" sees is reduced compared to the clean peak degree as there are new data heights induced by the noise. In the case of high SNR, the noise does not overpower the harmonic content and so it does not introduce too much of a difference in the Euclidean distance.

In the second experiment we use the publicly available Demixing Secrets Dataset (DSD100), containing the stems and mixtures of 100 songs sampled at 44100 Hz [83]. In this case the query will be clean vocal frames (i.e. lead harmonic source) and the goal is to find their corresponding frames in the mixture. The magnitude spectrogram for both the vocal and mixture tracks is calculated, with a window size

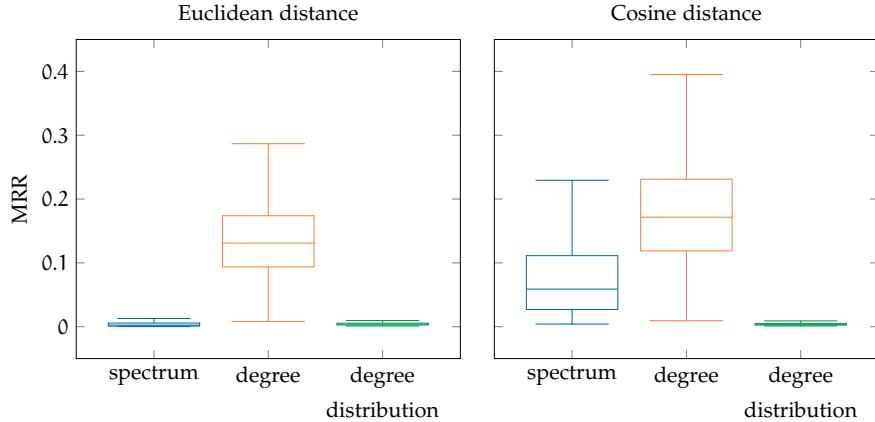


Figure 29: Mean-reciprocal-rank (MRR) of all mixtures in experiment 02: dataset Dev DSD100, vocal stems and their correspondent mixtures. Pair-wise similarity between the clean vocals and the mixture signals in the magnitude, degree and degree distribution domain for each track. The clean vocal time frames act as query and the expected closest neighbour is that time frame in the mixture.

of 2046 samples with 50% overlap, and only the first 500 frequency bins will be considered in the following (i.e. low-pass filter cut-off at around 10kHz). Based on the spectrogram energy of the vocal stem, we select the frames with vocal activity and use them as query frames. The pair-wise distance between the clean vocal query frames and all the frames in the mixture spectrogram is then calculated and sorted. The rank of the corresponding mixture frame containing the clean vocal query is then processed and stored to calculate the MRR. This procedure is repeated for the spectral visibility graph degree representation as well as for the degree distribution.

Figure 29 shows the results for experiment 02. The proposed representation is, in both cases (Euclidean and cosine distance), visibly much more suitable than the magnitude spectrogram and the degree distribution for the given task.

The fact that the degree distribution representation always achieved the worst results shows that the location of the harmonic peaks is a crucial piece of information for this type of harmonic similarity task. Even though the degree distribution was not advantageous in this case, there may be other audio analysis tasks for which it is useful,

such as those requiring pitch shift-invariance, like in Section 4.3 or [116].

Overall we can conclude that the proposed spectral visibility degree representation has properties valuable in audio analysis. In particular, the translation invariance of the proposed representation directly relates to a harmonic event in presence of broadband noise and we have here demonstrated its use for robust similarity measures of both synthetic and real harmonic sounds. Even though we have demonstrated one application of the proposed representation, we expect such a graph-based approach for audio analysis to find other useful applications in the future.

6.3 HOW TO COMPUTE VISIBILITY GRAPHS?

The straightforward computation of visibility graphs presents a worst case time complexity quadratic in the length of the series. Even though such complexity should not be an issue for medium-sized series ($10^4 - 10^5$ points), it remains inefficient for longer time series. Therefore, faster algorithms have been proposed employing a "Divide & Conquer" (DC) approach, reducing the average-case time complexity to $O(n \log n)$ [70].

Both of these approaches comprising the current existing methods to compute visibility graphs, are off-line algorithms as they require all the data points in the time series to be available before the graph is constructed. Consequently, the integration of new data points normally requires to re-compute the visibility graph from scratch, representing a major shortcoming limiting the real-world applications of visibility graphs.

Here we introduce, to the best of our knowledge, the first on-line algorithm to compute visibility graphs efficiently. The proposed algorithm employs an 'encoder/decoder' approach by means of a binary search tree representation of the time series (or any ordered sequence of data points). In particular, the time series is encoded into a binary search tree that can be updated every time a new chunk of time series

is available by merging its corresponding binary search trees. The resulting binary search tree can be decoded into a visibility graph when required. This introduced flexibility comes at no significant computational cost as the presented method shares the computational complexity of the current fastest visibility algorithm (DC).

6.3.1 *State of the art*

A straightforward approach to compute visibility graphs consists in checking whether any of the points of the time series is visible or not from every other point. This corresponds to evaluating the visibility criteria for every pair of points in the time series. Since we consider visibility as a symmetric relation, the total number of checks needed to obtain a visibility graph of a time series of n data points is equal to $n(n - 1)/2$, corresponding to a $O(n^2)$ time complexity.

In the case of horizontal visibility, one can take a step further and safely assume that no point after a value larger than the current value t_a will be horizontally visible from t_a . This observation effectively reduces the time complexity of the construction to $O(n \log(n))$ and, in the case of noisy (stochastic or chaotic) signals, it can be proved that this algorithm has an average-case time complexity $O(n)$ [85]. Nevertheless, all pairs of points need to be checked in the case of natural visibility. From now on, this simple approach will be referred to as the basic method for both natural and horizontal visibility computation³.

As an improved alternative for visibility computation, Lan et al. presented a "Divide & Conquer" approach [70]. This algorithm reduces the average case time complexity of the construction of the natural visibility graph to $O(n \log(n))$ and it significantly reduces computation time for most balanced time series.

The basic idea behind the "Divide & Conquer" algorithm is related to the horizontal visibility optimisation mentioned above. Once the maximum value M of the time series is known, one can safely assume

³ The original Fortran 90 implementations of basic algorithms to construct visibility graphs can be found at <http://www.maths.qmul.ac.uk/~lacasa/Software.html>

that the points on the right of M will not be naturally visible from the points on the left of M (the point M is effectively acting as a wall between the two sides of the time series). The same argument is then applied recursively on the two halves of the time series separated by M , where the local maxima subsequently found at each level are connected with an edge to the maxima at the level immediately above them. From now on, this improved method will be referred to as "Divide & Conquer" (or DC for short).

Both the basic method and DC are off-line approaches, meaning that they require all the points of the time series to be accessible at the beginning of the computation. This rigid requirement limits the applicability of visibility graphs, especially in fields like telecommunications or finance, where there is a constant incoming flow of new data to be processed and assimilated. Moreover, in such big data scenarios, one tends to favour an initial overall high level analysis that will reveal the need for further processing. This work-flow would benefit from dynamic algorithms unlike the ones presented above.

6.3.2 *Binary Search Tree Codec*

Here we propose a new method to compute visibility graphs on-line based on an encoding/decoding approach. In our method, the necessary visibility information is first encoded into an appropriately constructed binary search tree, and then successively decoded into a visibility graph when needed.

We can define a *tree* as a connected acyclic graph, where an acyclic graph, also known as *forest*, is simply a graph with no cycle. We say the tree is *rooted* when there is a node chosen as a *root*. The parent of a node is its neighbour on the unique path to the root, and its children are its other neighbours. Now we can define a *binary tree* as a rooted tree where each node has at most two children, designated as left or right child.

6.3.2.1 Encoding - Maximum Binary Search Tree

The construction of a maximum binary tree is fairly straightforward and its corresponding pseudo-code is shown in Algorithm 1. The first step is to sort the given time series in descending order of values, while storing the original position of each value in the time series. From now on, we will refer to the original positions as indices (i.e. t) and to the values of the times series simply as values (i.e. $g(t)$). In the case of repeated values in the sequence, the first encountered index will come first while sorting.

```
1 Node {
2     index : float    # x, input, argument
3     value : float   # f(x), output
4     left  : Node    # left child subtree
5     right : Node   # right child subtree
6 }
7
8
9
10
11 def buildTree(values : {float}, indexes: {float}):
12
13     root ← Node()
14
15     sorted_values = sort_descending(values)
16     sorted_indexes= indexes[getIndex(sorted_values)]
17
18     for (i, v) in (sorted_indexes, sorted_values):
19         root.add(Node(index = i, value = v))
20
21     return root
22
23
24
25
26 def add(self : {Node}, node : {Node}):
27
28     if self is empty :
29         self.index  = node.index
30         self.value  = node.value
31
32     else:
33         if node.index < self.index:
34             self.left.add(node)
35
36         else:
37             self.right.add(node)
```

Algorithm 1: Pseudocode of the algorithm used to build a maximum binary search tree

Once we have a list of values sorted in descending order, together with the corresponding indices, we follow the standard procedure to build a binary search tree based on the indices. Every entry in the index list will be a node and each node has a left and right child, as shown in the data structure proposed in Algorithm 1 (i.e., *Node*). The first node of the binary tree (the one with no parent) is called root.

In our case, the root will be the index of the datum corresponding to the maximum value in the time series, which is also the first entry in the index list.

The next index, corresponding to the point with the largest value smaller than the maximum, will then be added to the tree. If its index is smaller than the root, it will become the left child of root, while if its index is larger than the root it will become the right child of root (see function *add* in Algorithm 1). The next index to add will start off being compared to the root; if it's smaller, it will travel to the left of the tree and, if it's bigger, to the right. It will continue descending the tree in this manner until it finds an empty spot. We continue adding the indices in the list accordingly (see function *build_tree* in Algorithm 1) until there are no more data points (i.e. indices) to add.

In the case of the sample time series in Figure 30.a, the maximum is in position 5 and will therefore be the root of the binary tree. The point whose value is immediately smaller than the maximum is in position 4 (less than 5), so it will become the left child of the root. The third point in the list is in position 2, and will travel down the tree on the left-most branch (as it is smaller than both 5 and 4). The right branch of the tree is populated by the fourth point (in position 8), whose index is bigger than the root. In Figure 30.A one may appreciate the correlation between the time series and its associated binary tree structure. The visibility information captured by such a tree may also now be more apparent.

The time complexity of the procedure needed to encode the time series into the maximum binary search tree is $O(S + T)$ where $O(S)$ is the time complexity of sorting the series and $O(T)$ is the time complexity of the algorithm to construct the binary search tree. Sorting by comparisons is known to be $O(n \log n)$ (e.g., by using either Merge-Sort or QuickSort), while constructing a binary search tree costs on average $O(n \log n)$. Hence the overall average-case time complexity of the encoding step is $O(n \log n)$.

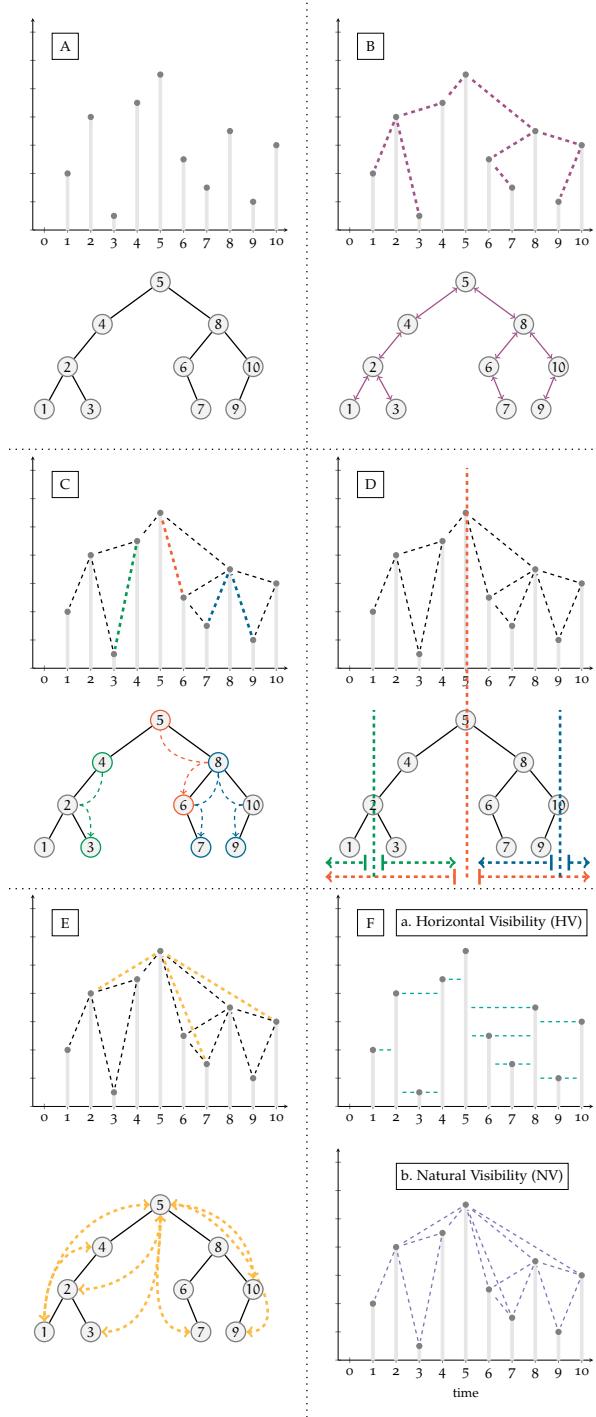


Figure 30: Representation of the different steps of the proposed algorithm for visibility graphs computation. In section A, the sample time series and its correspondent maximum binary search tree. Section B represents the connections deduced by the first connectivity rule. The second and third connectivity rules are illustrated in section C and D respectively. Section E shows the remaining checks needed to ascertain natural visibility. Finally, section F reports the horizontal and natural visibility graph associated to the original time series.

6.3.2.2 Decoding - Connectivity Rules

The structure of the maximum binary search tree encodes sufficient information about the time series to allow to efficiently construct the corresponding horizontal visibility graph. The decoding procedure is based on the following connectivity rules, also illustrated in Figure 30 :

1. All the nodes connected by an edge in the maximum binary search tree are visible to each other and therefore connected in the visibility graph (Figure 30.B);
2. Each node of the maximum binary search tree sees all the nodes in the left-most branch of the sub-tree rooted at its right child, as well as all the nodes in the right-most branch of the sub-tree rooted at its left child (Figure 30.C);
3. The nodes of the left sub-tree of a node i are not visible from the nodes of the right sub-tree of node i (Figure 30.D)

Note that, if there are no adjacent repeating amplitudes, the horizontal visibility graph is fully determined by these connectivity rules. In particular, when checking the connectivity rules, we simply skip a node if it has the same value as the current node. One can think of adjacent points with equal value as an interconnected ‘super node’, which takes the smallest index value when ‘looked’ from the left and the biggest index value when ‘looked’ from the right or from above.

Since the horizontal visibility decoding will always be fully determined by the three connectivity rules above, its time complexity is the sum of the time complexity of the rules. Essentially, each rule can be reduced to a series of look-ups in a binary search tree, and each look-up operation has time complexity $O(\log(n))$ in a balanced tree. These connectivity rules are applied to every node in the tree, and so the overall time complexity of decoding a horizontal visibility graph is $O(n \log(n))$. This represents a major improvement over the state-of-the-art algorithms, which can ramp up to $O(n^2)$ in the worst case scenario.

The construction of the natural visibility graph, instead, requires the creation of some connections that are not captured by the three connectivity rules above. Hence, in this case we need to perform additional visibility checks (Figure 30.E). In particular, for each node i we must check the natural visibility criterion with each node in the sub-tree rooted at the right child of i and with each node in the sub-tree rooted at the left child of i . These additional checks do not modify the average-case time complexity (which remains $O(n \log n)$), but the worst-case scenario still depends on the actual structure of the time series, and yields a time worst-case time complexity $O(n^2)$ which is realised by monotonically increasing or decreasing time series.

6.3.3 Time Complexity

In order to determine the time complexity of the proposed method, we will follow the standard procedure by considering the worst-case and average-case scenarios. In both scenarios, the time complexity of the encoding stage is determined by the time complexity of the sorting algorithm used, which in general is $O(n \log(n))$, and of the construction of the binary search tree, which is $O(n \log n)$. So in both cases encoding into a binary search tree costs $O(n \log n)$.

As discussed in the previous Section 6.3.2.2, decoding into a horizontal visibility graph is made through the three rules explained in Figure 30B-D, which require only a visit of the binary search tree (with time complexity $O(\log(n))$). Hence, the overall time complexity of encoding and decoding into a horizontal visibility graph remains $O(n \log(n))$.

The worst case for decoding into a natural visibility graph is that of monotonically increasing, monotonically decreasing, or constant series, whose corresponding binary search trees degenerate into a line. In this case, the second and third connectivity rules are trivial, leaving only the first rule and the additional natural visibility checks. More precisely, if the tree is a line we need to check the natural visibility among $(n - 1)(n - 2)/2$ pairs of nodes, while the visibility of the remaining $(n - 1)$ pairs of nodes is determined by the first connectiv-

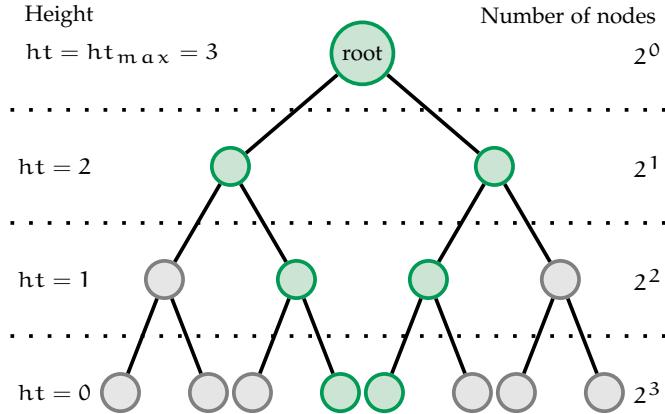


Figure 31: Representation of a perfectly balanced tree of height 4. The nodes in green are visible to the root and this visibility can be deduced by the proposed decoder (i.e. the connectivity rules). The number of nodes at each height in a balanced tree can always be expressed in base 2.

ity rule. Even though this requires $(n - 1)$ checks less than the basic implementation (which requires $n(n - 1)/2$), the time complexity will still be $O(n^2)$ for the worst case scenario.

For the average case we assume the maximum binary search tree to be balanced. This means that the connectivity rules of the decoder will significantly reduce the overall number of visibility checks. If we consider a perfectly balanced binary tree as shown in Figure 31, the inner left branch of the right sub-tree and the inner right branch of the left sub-tree of a node are visible to the parent node. These are represented in green in Figure 31 where the root is the parent node. This means that the visibility between the root and all the rest of nodes (the ones in blue) is unknown and needs to be checked.

Therefore we can deduce that the number of remaining visibility checks for the root in a balanced tree of height ht_{\max} is equal to $2^{ht_{\text{root}}+1} - 1 - 2ht_{\text{root}}$, where $2^{ht_{\text{root}}+1} - 1$ is the total number of nodes below the root while $2ht_{\text{root}}$ is the number of nodes whose visibility can be deduced by the three decoding rules (green nodes). Notice that the height of the root ht_{root} corresponds to the maximum height of the balanced tree ht_{\max} . The same reasoning applies to all the other nodes. More precisely, for a node at height ht , there will be $(2^{ht+1} - 1 - 2ht)$ remaining visibility checks to be performed.

In order to calculate the total number of remaining visibility checks, one needs to multiply the individual expression above by the number of nodes at that height $2^{ht_{max}-ht}$ and sum across all heights where the checks are needed (all except the last two). Therefore, one can express the total number of remaining natural visibility checks in a perfectly balanced binary tree as follows:

$$\sum_{ht=2}^{ht_{max}} 2^{ht_{max}-ht} [2^{ht+1} - (2ht + 1)] = \\ 2^{ht_{max}} [2(ht_{max} - 1) - \sum_{ht=2}^{ht_{max}} ht 2^{1-ht} - \sum_{ht=2}^{ht_{max}} 2^{-ht}] \quad (50)$$

Since the maximum height of a balanced tree with n nodes is $ht_{max} = \log_2(n)$, the total number of operation is dominated by the first term of the expression above,

$$2^{ht_{max}} 2(ht_{max} - 1) = 2n(\log_2(n) - 1) \quad (51)$$

while the remaining terms will only introduce logarithmic corrections. In conclusion, the time complexity of the decoding for natural visibility graphs is on average $O(n \log(n))$.

The proposed method has the same average-case time complexity than the DC algorithm, thus improving on the original basic algorithm for both horizontal and natural visibility graphs. In the Experiment section below we will see that in practice our algorithm out-competes the basic algorithm and performs as well as the DC approach, with the additional property of allowing for on-line assimilation of new data points.

6.3.4 On-line visibility graphs: merging binary trees

Every time a node is added to an existing binary search tree, it essentially travels down the tree, going left if smaller and right if larger, until it finds an empty space (see pseudocode function `add` in Algorithm 1). Therefore when a node is added to an existing binary tree there is no need to recalculate the tree structure from scratch. Due to the fact that the proposed encoder is a binary search tree, there is a possibility to efficiently update it on-line.

Given a time series and its corresponding binary search tree, we would like to integrate new data points in the tree structure without recomputing it from scratch. One could process the points of the newly available batch of data individually and include them in the existing tree structure by comparing both values and indices. However, other than being a time consuming approach for large numbers of points, processing points individually fails to include useful information of both the batch and the current tree structure. For example, let's consider that all the nodes in the batch to be added have larger indices than the nodes in the current tree structure, and so larger indices than the current root. This means, all the nodes in the batch will populate the right side of the resulting tree. If the nodes are inserted individually, this information will be overlooked producing an inefficient algorithm.

Therefore, we propose to take a different approach by treating the new batch of points as an entity. More precisely, we propose to compute the binary search tree of the new nodes and merge it with the previous tree structure. In this way, if all the new nodes indices are larger than the current root, one can include such information and produce an optimised algorithm, where potentially only one comparison is needed to merge the current with the batch tree. This is the case for real-time incoming data, as the batch's nodes always have larger time values (indices) than the previous points in the time series.

Furthermore, the proposed merge approach covers both append and insert operations. In terms of time series representation, this means one could update the binary tree codec with observations that happened later in time or with a higher time resolution. This novel introduced flexibility for visibility computation, opens the door to new applications such as big data or audio applications where the sampling rate may vary at different analysis stages.

In order to merge two trees, we propose to compare them by levels, increasing depth at every recursion of the merge function outlined in Algorithm 2. The comparison happens in two steps: firstly the node values at a level are compared to determine which node will occupy that location in the resulting tree; secondly, the node indices are com-

pared to determine which direction the rest of the nodes will travel down in depth.

Following the construction of the proposed binary search tree, the node with larger value will be chosen and the rest of the nodes will travel left if their indices are smaller than the chosen one and right otherwise. The nodes to be compared are the children of the chosen node with the nodes from the previous level that were not chosen; starting off by comparing the two roots of the trees to be merged.

```

def merge(input:{Node}):

    if input is empty: return null

    5      r     ← min_index(maxima_value(input))
    pool ← input \ {r}

    pool.append(r.left, r.right)

    10    for n in input \ {r} :

        for c in [n.left, n.right] :

            if sign(n.index - r.index)
               15      ≠ sign(c.index - r.index) :

                pool.append(c)
                n.remove(c)

    20    return Node(
            index = r.index,
            value = r.value,
            left   =
            25      merge({p | p ∈ pool, p.index < r.index }),
            right =
            merge({p | p ∈ pool, p.index > r.index }))
```

Algorithm 2: Pseudocode of the proposed algorithm to merge two binary trees defined by their root (class Node). The input is a list of roots to be merged.

Usually, the children of the nodes that travel down in depth are not included in the level comparison. However, when new data is to be inserted to the existing series, the child of the node traveling down could have an index corresponding to the other branch of the resulting tree. In this case, the connection between the node and that child will be broken thereafter.

6.3.5 Numerical Experiments

Here we show how the proposed visibility algorithm compares to the state of the art. All the related code necessary to run the following experiments is implemented in Python 2.7 and freely available online⁴. The machine used in the simulations is an early 2015 MacBook Pro Retina with a 2.9GHz Intel Core i5 processor and 16GB of RAM.

To put the presented algorithm into context [70], in Figure 32 we report the computation time needed by current visibility algorithms on different synthetic time series of increasing length. Since the actual efficiency of each algorithm depends to some extent on the character of the original time series, we considered uniform random noise (which has no structure and on average produces almost-balanced binary search trees), a Conway series (which has a quite rich structure and corresponds to a quite unbalanced tree), and a random walk series (which represents the more realistic scenario of a signal with both structure and noise).

Similarly to [70], we define a recursively generated Conway series of size n as $a(t) = \frac{t}{2}$, where

$$a(t) = a(a(t-1)) + a(t - a(t-1)) \quad \forall t \in [2, n] \quad (52)$$

and $a(1) = a(2) = 1$. The random walk time series $w(t)$ is generated by sampling α from an uniform distribution in the range $[0, 1]$, such that

$$w(t) = w(t-1) + \epsilon, \quad \text{for } \epsilon = \begin{cases} -1 & \text{if } \alpha < 0.5 \\ 1 & \text{if } \alpha \geq 0.5 \end{cases} \quad (53)$$

The code to generate both of these series is available online⁴.

In the first case of random series (first column in Figure 32) we observe the largest gap in computation time between the basic algorithm and the more efficient ones as it corresponds to the aforementioned average case where both algorithms (DC and the proposed one) significantly reduce the number of operations. Such differences are more prominent in the computation of the horizontal visibility graph (third row in Figure 32).

⁴ Available at <https://github.com/delialia/bst>

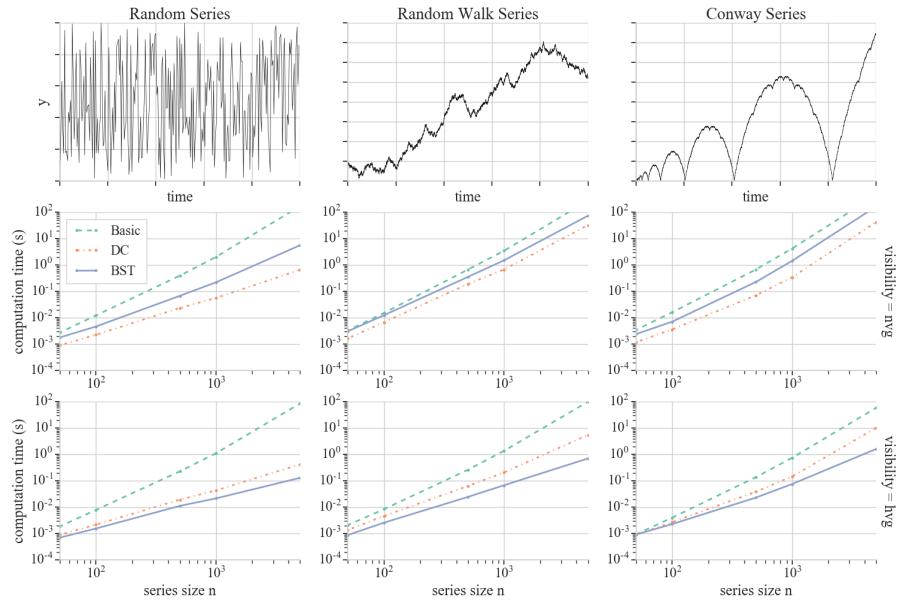


Figure 32: Computation time of the natural and horizontal visibility graph (nvg, second row; hvg, third row) of different time series (examples on first row) using the current visibility algorithms: Basic, Divide & Conquer (DC), and the proposed binary search tree (BST) method. Each point at every series size is the mean of the computation time for 10 series of that size.

The proposed method (BST) shows a similar trend to the state-of-the-art natural visibility computation method (DC) in Figure 32, although it appears to be overall slower. However, the proposed method clearly outperforms the state-of-the-art in horizontal visibility computation (third row). This comes at no surprise, as mentioned in Section 6.3.3, unlike natural visibility, the horizontal visibility is fully defined by the three aforementioned connectivity rules, and so more efficient than current methods (basic and DC).

Additionally, in Figure 33 we present a similar computational time analysis over real samples of speech (English language) [50] and financial data [93]. We sample 100 times series (first 1000 points) from the TIMIT acoustic-phonetic continuous speech corpus (630 American English speakers reading ten phonetically rich sentences recorded at 16kHz) [50], as well as 100 time series from the daily prices of US stocks traded in 2013 used in [93].

Figure 33 is particularly interesting as it clearly shows a correlation between time computation and the time series structure (please note

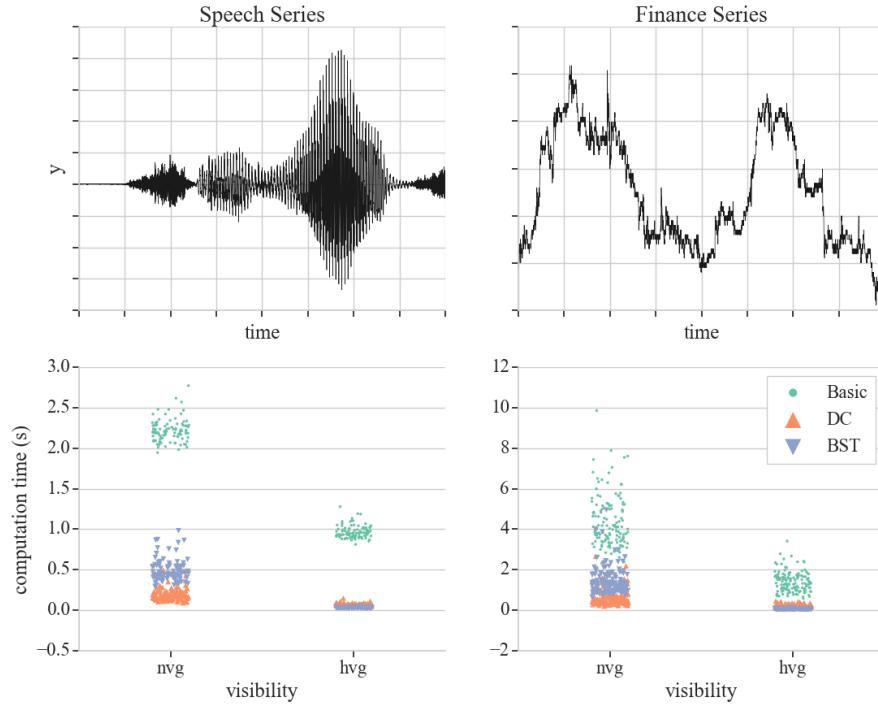


Figure 33: Current and proposed visibility algorithms computation time for 100 speech and finance time series of 1000 points. The speech time series are sampled from the training TIMIT dataset [50]. The finance time series corresponds to the 2013 quarterly data used in [93].

the different scale for time computation). Even though the time computation may differ, the DC and proposed method distribution vary very little between data types in comparison to the relatively high spread observed for the basic algorithm.

The horizontal visibility computation remains stable in both the DC and proposed method, and could potentially be considered independent of the data type given a time computation scaling factor. This behaviour was equally expected as the proposed method is fully defined by the aforementioned connectivity rules and has average-case time complexity $O(n \log n)$.

On the other hand, Figure 33 suggests that the efficiency of the computation of natural visibility graphs is subject to wider fluctuations. The position of the maximum in the time series affects the efficiency of both the DC and the proposed method, as it will determine the

number of additional visibility checks needed to obtain the natural visibility graph.

The English speech time series considered here will typically have its maximum somewhere towards the middle section of the signal (since we rarely tend to raise our voice at the end of our speech). Therefore the proposed method will most probably produce an almost balanced binary search tree for the speech time series, yielding a time complexity of $O(n \log n)$. For this reason, we observe a wider gap in computation time between the basic method and the faster alternatives for the speech data in Figure 33 than for the financial time series.

In terms of computation time, the proposed method and the DC one are closely related. They are both quicker than the basic implementation in both natural and horizontal visibility and they both present similar trends for increasing time series size (Figure 32). However, the proposed algorithm has proven to consistently be the quickest option for horizontal visibility graph computation. On the other hand, the DC algorithm in general does perform better than the proposed method for natural visibility computation. Even though at this point both DC and the proposed method seem equally good of an option for fast visibility computation, the presented algorithm has the additional property of allowing on-line assimilation of new data, which is something not easily achievable in either the basic approach or the DC algorithm.

The most straightforward way to assess the on-line functionality of the proposed method is to compare it with the equivalent off-line approach. In our case, it directly relates to the binary tree codec. Given a batch of new points to be added to the time series visibility analysis, in the off-line approach, the new batch is simply added to the time series itself and then the binary tree codec must be re-computed from scratch. In the proposed on-line approach, the next batch is encoded into its own binary tree that is then merged to the existing codec using the procedure detailed in Algorithm 2. Note that the decoding step remains the same for the on-line and off-line approach, and so the comparison will essentially be between computing a codec from

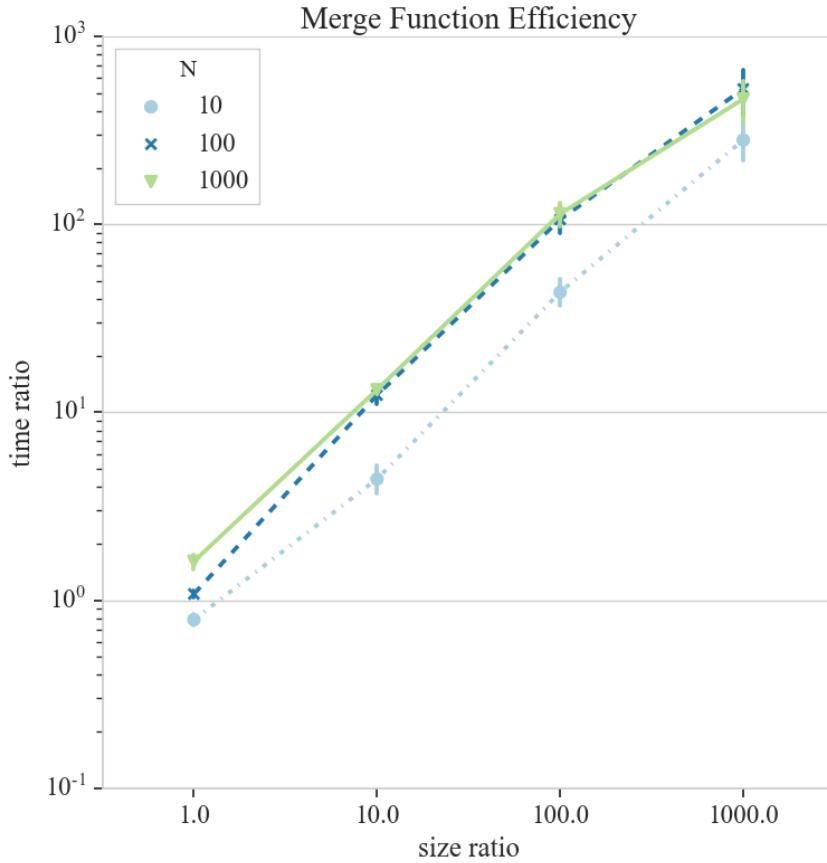


Figure 34: Given a random time series (size L) and a batch of new random points (size N) to be added to it, this plot shows the advantage, in terms of computation time, of the proposed on-line approach versus the off-line alternative. The proposed method computation time is the time it takes to build the maxima tree of the new points and merge it with the existing time series tree (i.e. $t_{\text{on-line}}$). The off-line alternative computation time is the time it takes to build a new maxima binary tree from scratch including the new points to the time series (i.e. $t_{\text{off-line}}$). The time ratio is the log scale of $t_{\text{off-line}}/t_{\text{on-line}}$, how much quicker the proposed method is. The size ratio is L/N , how much bigger the time series is compared to the batch to be added. Both append and insert scenarios are represented here, 10 random cases of each were computed. The point in the graph is the mean of these 20 cases and its uncertainty is captured by the error bars.

scratch (off-line) and merging two codecs into a single binary search tree (on-line).

Figure 34 shows how much quicker the computation of the on-line method (codec for new data + merging) is in comparison to the computation time of the off-line approach (codec from scratch), for different time series and batch sizes. In particular, the on-line approach is always better if the new batch to be added is equal or bigger than the existing time series, especially for large time series.

Overall, the proposed visibility algorithm represents a substantial improvement over the state-of-the art for horizontal visibility computation, and is on par with the most efficient natural visibility algorithm (i.e. DC) available. Moreover, the procedure to assimilate new data by means of merging the corresponding binary search tree encoding into the existing tree allows for efficient on-line computation of visibility graphs, and represents a substantial speed-up with respect to the existing off-line algorithms. This novel on-line capability broadens the applications for visibility graphs at no additional computational cost.

6.4 SUMMARY

Graph theory is emerging as a new source of tools for time series analysis and in this chapter we investigate its potential for audio related tasks. One promising method is to transform a signal into its visibility graph, a representation which captures many interesting aspects of the signal, as defined in Section 6.1.

In Section 6.2 we introduce the visibility graph for audio spectra and propose a novel representation for audio analysis: the spectral visibility graph degree. Such a representation inherently captures the harmonic content of the signal whilst being resilient to broadband noise. We present experiments in Section 6.2.1 demonstrating its utility to measure robust similarity between harmonic signals in real and synthesised audio data. The source code is available online⁵.

⁵ Available at <https://github.com/delialia/vgspectra>

However, the straightforward computation of visibility graphs is time-consuming and rigid, motivating the development of more efficient algorithms. In Section 6.3 we present a highly efficient method to compute visibility graphs with the further benefit of flexibility: on-line computation.

We propose an encoder/decoder approach, with an on-line adjustable binary search tree codec for time series as well as its corresponding decoder for visibility graphs. The proposed method for computation of visibility graphs offers an on-line computation solution at no additional computation time cost, and would allow to employ visibility graphs in the analysis of large-scale time series and for the on-line assimilation of new data. The source code is also available online⁶.

In the next chapter we will briefly outline possible steps to continue the research line presented in this thesis, under the name of future work.

⁶ Available at <https://github.com/delialia/bst>

Part IV

TAKE HOME MESSAGE

Future research avenues are discussed, followed by a conclusion on the main message of this dissertation.

FUTURE WORK

The work we have just discussed brings a new perspective, introducing graph theory to well established audio tasks. Therefore, multiple research avenues present themselves, exploring the applicability and flexibility of the proposed graph-based methodologies. For instance, one could wonder if we could learn source-specific graphs, or if the binary tree encoder proposed in Section 6.3 already holds a structure useful for audio tasks without needing to decode into a visibility graph.

However, one of the most natural steps as a continuation of this thesis would be to investigate the use of the proposed spectral visibility graph representation presented in Section 6.2 within the KAM framework. Would its vertical translation invariance be useful to define a broadband interference resilient kernel?

7.1 SPECTRAL VISIBILITY GRAPHS FOR KAM ?

As explained in the first part of this thesis, the basic idea behind KAM is that one can reconstruct the magnitude for a given source by analysing the values at the locations where the source is likely to assume similar values, ultimately relying on the assumed repetition of sound events in musical signals. The success of the separation will depend on the ability to identify similar sound events to the source of interest in the presence of overlaying sources. The source similarity is determined by a source-specific kernel function, which often corresponds to a k nearest neighbours (kNN) search based on the Euclidean distance. Such a simple kernel however implicitly assumes the source of interest to be dominant in the magnitude domain, which might not be the case in a low signal-to-noise ratio (SNR) scenario.

This means that basic KAM can fail to have desired results in the most difficult cases, and could even make audio interference more perceptible.

The work presented in Section 4.4 offers a remedy to this limitation of KAM, where a semi-supervised non-negative matrix factorisation (NMF) approach learns models for the signal and the noise, which are then used to achieve an adaptive interference resilient kernel for the similarity search in KAM. This improves the overall separation quality but at a significant computational cost: it introduces new parameters to be optimised/selected (refer back to Table 7), as well as introducing the need for training data and a prior dictionary learning stage. How could we secure KAM’s simplicity whilst ensuring a successful separation in low SNR scenarios?

In Section 6.2 we presented spectra visibility graphs as a novel alternative representation to magnitude spectrograms. Such a representation retains the harmonic peaks salience in presence of broadband events and has proven its adequacy for harmonic similarity measurements. The burst-like interferences in our application are all characterised by their powerful broadband nature, unlike the musical signal mainly characterised by its harmonic contribution. Hence, one can expect the spectra visibility graph to preserve the harmonic peaks of the musical signal in presence of interference. In this context, one could safely assume the source of interest to be dominant and define a successful kernel function. Therefore we propose to explore the potential of such spectra visibility graphs as an alternative representation to obtain an interference resilient kernel within the KAM framework in an interference reduction task.

If we were to develop the proposed method as an extension to the subset of the KAM framework presented in this thesis (in detail in Chapter 3), we could differentiate the search from the estimation space by introducing the novel spectra visibility graph as a representation for the k-NN kernel function.

Given a magnitude spectrogram $\bar{X} \in \mathbb{R}^{F \times T}$ of a input mixture $x(t)$, we set $K \in \mathbb{R}^{F \times T}$ to be the degree matrix of the spectra visibility graph corresponding to \bar{X} as defined in Section 6.2. Now we can define a

k-NN kernel function based on the Euclidean distance between the degree vectors of the time frames visibility graphs (i.e. columns of K).

Let \mathbb{K} be the set of all degree vectors \vec{k}_t of length F such that $|\mathbb{K}| = T$. The k-NN kernel function will specify for every degree vector \vec{k}_t a set of k neighbours $\mathcal{I}(f, t) \in \mathbb{K}^k$ containing a similar degree value of the music source of interest. Such that (f, \tilde{t}) is in $\mathcal{I}(f, t)$ if $\vec{k}_{\tilde{t}}$ is among the k most similar degree vectors.

The degree vectors directly correspond to the time vectors of the magnitude spectrum \bar{X} , and so even though the kernel is defined in a different domain, the localisation of the nearest neighbours remains the same. Therefore we can express the estimation problem in the same way as in KAM, using \bar{X} in the same model cost function defined Chapter 3 with the median operator as solution.

Since the proposed graph-based representation is one of the first of its kind, the separation performance comparison with standard methods working in the magnitude domain is not straightforward. As, in addition to choosing an adequate separation task for the evaluation, one could also explore the capability of the proposed representation to perform source detection. Similarly to the proposed method in Section 4.4, where the semi-informed NMF estimate serves as an interference detector, one can imagine the absolute values in K to also be sensitive to broadband presence, and therefore potentially serving as an indication of certain sources activity. Hence, the analysis and empirical evaluation of such a method remains to be seen.

8

CONCLUSION

This thesis is divided into four parts: Part [i](#) introducing the dissertation, Part [ii](#) interested in source separation methods based in median filtering, Part [iii](#) proposing a different perspective through graph theory and concluding in Part [iv](#).

After framing the dissertation in Chapter [1](#), the basic concepts of blind source separation were introduced in Chapter [2](#), along with a brief review on the relevant literature.

Most of the discussion in this thesis revolved around the family of source separation methods based on median filtering, known as the Kernel Additive Modelling (KAM) framework, which was fully defined in Chapter [3](#). The basic idea behind KAM relies on the repetition of music signals to reconstruct the magnitude spectrogram of the target source by analysing the values at the locations where it is likely to assume similar values.

In Chapter [4](#) we showed how the success of the separation depends on the ability of the proximity kernel to identify similar sound events to the source of interest in the presence of overlaying sources. We further discussed the flexibility and limitations of a particular popular kernel, the frame-wise k-NN, implicitly assuming repetition in both time and frequency, and expecting the target source to be energetically dominant. We proposed to introduce a temporal context in the kernel Section [4.2](#) to temporally stabilise the target source estimate. Then, we presented a shift-invariant kernel in Section [4.3](#) that expands the pool of potential neighbours by introducing a degree of freedom in the frequency direction in the baseline kernel function. We also presented an acceleration technique in the specmurt domain, Section [4.3.1](#), to speed up the search of similarity in the signal, separating for the first time the search from the processing space.

In Section 4.4 we challenged the use of KAM under low SNR conditions where the target source is not always the most prominent. In this case, we proposed to use a semi-informed NMF to yield an initial target source estimate fulfilling KAM's assumptions, combining for the first time a machine learning approach with KAM framework and showing their complementary nature.

At the end of Chapter 4 we started the discussion about the sole parameter of KAM framework, which is the number of nearest neighbours k . Until now, there was little to no discussion on how to set k which is expected to play a key role in the separation performance. In Chapter 5 we proposed a method to automatically optimise such a parameter, introducing graph theory concepts to the framework. We also analysed its impact in Section 5.3 and questioned the adaptability of KAM as well as the appropriateness of the standard evaluation metrics.

We pursued the use of graph theory in an audio context in Chapter 6 by introducing a powerful time series analysis tool, visibility graphs, to spectra. We derived a new representation for audio, spectral visibility graphs, as an alternative to the magnitude spectrogram commonly used in audio related tasks. The new representation is invariant to a number of transformations and enhances harmonics peaks. We showed its use for similarity measure of harmonic signals in presence of broadband noise.

Finally we discussed in Section 6.3 the current methods to compute visibility graphs and we proposed the first algorithm capable of computing a visibility graph on-line while remaining as efficient as the state-of-the-art. This contribution transcends the audio community, broadening the use of visibility graphs as a time series analysis tool to large-scale and on-line applications.

In short, the main contributions of this dissertation can be summarised as follows:

- Introduction of a temporal context in the proximity kernel
- Proposed shift-invariant kernel

- Integration of machine learning in KAM framework
- Integration of graph theory tools in KAM framework
- First method to automatically optimise the sole parameter in KAM
- Introduction of visibility graphs to spectra
- Novel graph-based representation for audio
- First on-line algorithm to compute visibility graphs

and the main teachings that may be useful for future research can be summarised as:

- Including a temporal context in the proximity kernel temporally stabilises the estimates
- Differentiating the search from the estimation space opens new avenues to improve KAM framework
- A shift-invariant kernel boosts separation results
- Machine learning can be used for low SNR scenarios in detection and creation of an initial model
- NMF and KAM compliment each other
- Graph theory tools offer a new modelling perspective in KAM
- The hubness of the k-NN graph is a good indicator to pick k
- Spectral visibility graphs enhance harmonic content in signals
- Visibility graphs invariance to vertical translation is useful for harmonic similarity
- Encoding a visibility graph in a binary search tree allows for on-line computation

Overall we intend this thesis to promote divergent thinking, as not every problem requires the same solution. We introduced a simple and efficient method for source separation and gave some indication

on how to modify it to suit different music recording scenarios. Its flexibility should allow, and hopefully inspire, the reader to device novel efficient modifications for other challenging applications. We then took a step further by re-defining the working space introducing graph theory to the model. This shift in paradigm brought a new perspective and so new ideas. The graph representations presented are efficient, non-parametric and deterministic. The thinking-space created, backed-up by a well established field in mathematics, challenges the latest trend of black-box thinking; and so one can't help but wonder: could graph theory shed some light on the current deep learning state-of-the-art? Is the new audio standard representation going to be graph based? Are we heading towards graph signal processing? Only time will tell, but, as a great scientist once said, the important thing is to never stop questioning.

Part V

APPENDIX

A

COMPLEXITY ANALYSIS FOR SHIFT INVARIANT KAM

Table 8: Notation for complexity analysis

	NOTATION	DIMENSION
Input mixture magnitude	\bar{X}_q	$F \times T$
Frames to be processed	$\chi_A \subseteq \bar{X}_q$	$F \times T_A$
Rest of frames to be compared with	$\chi_B = \bar{X}_q \setminus \chi_A$	$F \times T_B$

Table 9: KAM baseline method complexity

	OPERATION	DIMENSION	COMPLEXITY
Frame-wise distance	$D(\chi_A, \chi_B)$	$T_B \times T_A$	$O(FT_A T_B)$
Find k closest frames	sort	$T_B \times T_A$	$O(T_A T_B \log(T_B))$
TOTAL COMPLEXITY :			$O(T^2(F + \log(T)))$

Table 10: KAM shift invariant complexity

	OPERATION	DIMENSION	COMPLEXITY
Iterate over shifts	for every shift δ :	Δ	
Frame-wise distance	$\rightarrow D(\chi_A, \chi_B)$	$T_B \times T_A$	$O(\Delta FT_A T_B)$
Save minimum distance and its shift	\rightarrow minimum search	$T_B \times T_A$	$O(\Delta T_A T_B)$
Find k closest frames	sort	$T_B \times T_A$	$O(T_A T_B \log(T_B))$
Shift every k-NN for alignment	shift	$k \times F \times T_A$	$O(kfT_A)$
TOTAL COMPLEXITY :			$O(T^2(F^2 + \log(T)))$

Table 11: KAM shift invariant acceleration complexity

	OPERATION	DIMENSION	COMPLEXITY
Specmurt	$\mathcal{F}\langle \bar{X}_q \rangle$	$F \times T$	$O(TF\log(F))$
Frame-wise distance	$D(\chi_A, \chi_B)$	$T_B \times T_A$	$O(\Delta(F/2)T_A T_B)$
Find k closest frames	sort	$T_B \times T_A$	$O(T_A T_B \log(T_B))$
Specmurt analysis	$\mathcal{F}\langle \frac{\mathcal{J}\mathcal{F}\langle . \rangle}{\mathcal{J}\mathcal{F}\langle . \rangle} \rangle$	$k \times F \times T_A$	$O(kT_A F\log(F))$
Find optimal shift	maximum	$k \times F \times T_A$	$O(kFT_A)$
Shift every k-NN for alignment	shift	$k \times F \times T_A$	$O(kfT_A)$
TOTAL COMPLEXITY :		$O(T^2(F + \log(T))) + O(FT\log(F))$	

BIBLIOGRAPHY

- [1] Samer Abdallah and Mark Plumbley. "Polyphonic Transcription by Non-Negative Sparse Coding of Power Spectra". In: *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. 2004, pp. 318–325.
- [2] Shoko Araki, Francesco Nesta, Emmanuel Vincent, Zbynek Koldovský, Guido Nolte, Andreas Ziehe, and Alexis Benichoux. "The 2011 Signal Separation Evaluation Campaign (SiSEC2011): Audio Source Separation". In: *Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA / ICA)*. 2012, pp. 414–422.
- [3] Gonzalo R Arce. *Nonlinear signal processing: a statistical approach*. John Wiley & Sons, 2005.
- [4] Jørgen Bang-Jensen and Gregory Z Gutin. *Digraphs: theory, algorithms and applications*. Springer Science & Business Media, 2008.
- [5] Juan Pablo Bello. "Audio-based cover song retrieval using approximate chord sequences: testing shifts, gaps, swaps and beats". In: *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*. 2007, pp. 239–244.
- [6] Emmanouil Benetos, Margarita Kotti, and Constantine Kotropoulos. "Musical instrument classification using non-negative matrix factorization algorithms and subset feature selection". In: *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*. Vol. 5. 2006, pp. 221–224.
- [7] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. "Automatic music transcription: challenges and future directions". In: *Journal of Intelligent Information Systems* 41 (2013), pp. 407–434.
- [8] Nancy Bertin, Roland Badeau, and Emmanuel Vincent. "Enforcing Harmonicity and Smoothness in Bayesian Non-negative

- Matrix Factorization Applied to Polyphonic Music Transcription". In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.3 (2010), pp. 538–549.
- [9] Rachel M. Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Juan Pablo Bello. "MedleyDB: A Multi-track Dataset for Annotation-Intensive MIR Research". In: *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. 2014, pp. 155–160.
- [10] Giovanni Bonanno, Fabrizio Lillo, and Rosario N. Mantegna. "High-frequency cross-correlation in a set of stocks". In: *Quantitative Finance* 1.1 (2001), pp. 96–104.
- [11] Albert S. Bregman. *Auditory Scene Analysis: The Perceptual Organization of Sound*. MIT Press, 1990.
- [12] Judith C. Brown. "Calculation of a constant Q spectral transform". In: *Journal of the Acoustical Society of America* 89.1 (1991), pp. 425–434.
- [13] Ed Bullmore and Olaf Sporns. "Complex brain networks: graph theoretical analysis of structural and functional systems". In: *Nature Reviews Neuroscience* 10.3 (2009), pp. 186–198.
- [14] Ed Bullmore and Olaf Sporns. "The economy of brain network organization". In: *Nature Review Neuroscience* 13.5 (2012), p. 336.
- [15] Estefanía Cano, Corey Cheng, et al. "Melody line detection and source separation in classical saxophone recordings". In: *Proceedings of the International Conference on Digital Audio Effects (DAFx)*. 2009, pp. 1–6.
- [16] Estefanía Cano, Christian Dittmar, and Sascha Grolmisch. "Songs2See: learn to play by playing". In: *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. 2011.
- [17] Estefanía Cano, Christian Dittmar, and Gerald Schuller. "Rethinking Sound Separation: Prior Information and Additivity Constraint in Separation Algorithms". In: *Proceedings of the International Conference on Digital Audio Effects (DAFx)*. 2013.

- [18] Estefanía Cano, Derry FitzGerald, and Karlheinz Brandenburg. "Evaluation of Quality of Sound Source Separation Algorithms: Human Perception vs Quantitative Metrics". In: *Proceedings of the European Signal Processing Conference (EUSIPCO)*. 2016.
- [19] Estefanía Cano, Derry FitzGerald, Antoine Liutkus, Mark D Plumbley, and Fabian-Robert Stöter. "Musical source separation: An introduction". In: *IEEE Signal Processing Magazine* 36.1 (2018), pp. 31–40.
- [20] Yong-Choon Cho and Seungjin Choi. "Learning nonnegative features of spectro-temporal sounds for classification". In: *In Proceedings of InterSpeech*. 2004.
- [21] Pierre Comon and Christian Jutten. *Handbook of Blind Source Separation, Independent Component Analysis and Applications*. Academic Press, Elsevier, 2010.
- [22] Matt Cranitch, Marcin T Cychowski, and Derry FitzGerald. "Towards an inverse constant q transform". In: *Audio Engineering Society Convention 120*. Audio Engineering Society. 2006.
- [23] Antoine Deleforge and Walter Kellermann. "Phase-optimized K-SVD for signal extraction from underdetermined multichannel sparse mixtures". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 355–359.
- [24] Christian Dittmar, Patricio López-Serrano, and Meinard Müller. "Unifying local and global methods for harmonic-percussive source separation". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018, pp. 176–180.
- [25] Jonathan F. Donges, Reik V. Donner, and Jürgen Kurths. "Testing time series irreversibility using complex network methods". In: *Europhysics Letters (EPL)* 102.1 (2013), p. 10004.
- [26] Reik V Donner, Yong Zou, Jonathan F Donges, Norbert Marwan, and Jürgen Kurths. "Recurrence networks: a novel paradigm for nonlinear time series analysis". In: *New Journal of Physics* 12.3 (2010), p. 033025.

- [27] Reik V. Donner, Michael Molloy Small, Jonathan F. Donges, Norbert Marwan, Yong Zou, Ruoxi Xiang, and Jürgen Kurths. "Recurrence-based time series analysis by means of complex network methods". In: *International Journal of Bifurcation and Chaos* 21.04 (2011), pp. 1019–1046.
- [28] Jonathan Driedger and Meinard Müller. "Extracting Singing Voice from Music Recordings by Cascading Audio Decomposition Techniques". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 2015, pp. 126–130.
- [29] Jonathan Driedger, Stefan Balke, Sebastian Ewert, and Meinard Müller. "Template-Based Vibrato Analysis in Music Signals". In: *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. 2016, pp. 239–245.
- [30] Ngoc Q. K. Duong, Emmanuel Vincent, and Rémi Gribonval. "Under-Determined Reverberant Audio Source Separation Using a Full-Rank Spatial Covariance Model". In: *IEEE Transactions on Audio, Speech, and Language Processing* 18.7 (2010), pp. 1830 –1840.
- [31] Jean-Louis Durrieu and Jean-Philippe Thiran. "Musical audio source separation based on user-selected Fo track". In: *Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*. 2012, pp. 438–445.
- [32] Dalia El Badawy, Ngoc QK Duong, and Alexey Ozerov. "On-the-fly audio source separation". In: *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*. 2014, pp. 1–6.
- [33] Daniel P.W. Ellis and Graham E. Poliner. "Identifying 'Cover Songs' with Chroma Features and Dynamic Programming Beat Tracking". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Vol. 4. 2007, pp. 1429–1432.
- [34] Valentin Emiya, Emmanuel Vincent, Niklas Harlander, and Volker Hohmann. "Subjective and Objective Quality Assessment of Audio Source Separation". In: *IEEE Transactions on*

- Audio, Speech, and Language Processing* 19.7 (2011), pp. 2046–2057.
- [35] Sebastian Ewert and Meinard Müller. “Score-Informed Voice Separation for Piano Recordings”. In: *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. 2011, pp. 245–250.
 - [36] Sebastian Ewert, Bryan Pardo, Meinard Müller, and Mark D. Plumley. “Score-Informed Source Separation for Musical Audio Recordings: An Overview”. In: *IEEE Signal Processing Magazine* 31.3 (2014), pp. 116–124.
 - [37] J. H. Feldhoff, R. V. Donner, J. F. Donges, N. Marwan, and J. Kurths. “Geometric signature of complex synchronisation scenarios”. In: *Europhysics Letters (EPL)* 102.3 (2013), p. 30007.
 - [38] Paweł Fiedor. “Networks in financial markets based on the mutual information rate”. In: *Physical Review E* 89 (2014), p. 052801.
 - [39] Derry FitzGerald. “Harmonic/Percussive Separation Using Median Filtering”. In: *Proceedings of the International Conference on Digital Audio Effects (DAFx)*. 2010, pp. 246–253.
 - [40] Derry FitzGerald. “User assisted separation using tensor factorisations”. In: *Proceedings of the European Signal Processing Conference (EUSIPCO)*. 2012, pp. 2412–2416.
 - [41] Derry FitzGerald. “Vocal separation using nearest neighbours and median filtering”. In: *Proceedings of the Irish Signals and Systems Conference (ISSC)*. 2012, pp. 1–5.
 - [42] Derry FitzGerald. “The good vibrations problem”. In: *Audio Engineering Society Convention 134*. Audio Engineering Society. 2013.
 - [43] Derry FitzGerald, Matt Cranitch, and Eugene Coyle. “Extended Nonnegative Tensor Factorisation Models for Musical Sound Source Separation (Article ID 872425)”. In: *Computational Intelligence and Neuroscience* 2008 (2008).
 - [44] Derry FitzGerald, Zafar Rafii, and Antoine Liutkus. “User assisted separation of repeating patterns in time and frequency using magnitude projections”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2017, pp. 396–400.

- [45] Derry FitzGerald, Antoine Liutkus, Zafar Rafii, Bryan Pardo, and Laurent Daudet. "Harmonic/Percussive Separation Using Kernel Additive Modelling". In: *Irish Signals and Systems Conference (IET)*. 2014, pp. 35–40.
- [46] Derry Fitzgerald, Antoine Liutkus, and Roland Badeau. "Projet— spatial audio separation using projections". In: *Proceedings IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016, pp. 36–40.
- [47] Ryan Flanagan and Lucas Lacasa. "Irreversibility of financial time series: A graph-theoretical approach". In: *Physics Letters A* 380.20 (2016), pp. 1689 –1697.
- [48] Arthur Flexer, Dominik Schnitzer, and Jan Schlüter. "A MIREX Meta-analysis of Hubness in Audio Music Similarity." In: *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*. 2012, pp. 175–180.
- [49] Joachim Ganseman, Paul Scheunders, Gautham J. Mysore, and Jonathan S. Abel. "Source Separation by Score Synthesis". In: *Proceedings of the International Computer Music Conference (ICMC)*. 2010, pp. 462–465.
- [50] John S Garofolo, Lori F Lamel, William M Fisher, Jonathan G Fiscus, and David S Pallett. "DARPA TIMIT acoustic-phonetic continuous speech corpus CD-ROM. NIST speech disc 1-1.1". In: *NASA STI/Recon technical report n 93* (1993).
- [51] Steven L Gay and Jacob Benesty. *Acoustic signal processing for telecommunication*. Vol. 551. Springer Science & Business Media, 2012.
- [52] Stefan Uhlich Franck Giron and Yuki Mitsufuji. "Deep neural network based instrument extraction from music". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 2135–2139.
- [53] Emad M Grais, Gerard Roma, Andrew JR Simpson, and Mark D Plumbley. "Single-channel audio source separation using deep neural network ensembles". In: *Audio Engineering Society Convention 140*. Audio Engineering Society. 2016.

- [54] Daniel W. Griffin and Jae S. Lim. "Signal estimation from modified short-time Fourier transform". In: *IEEE Transactions on Acoustics, Speech and Signal Processing* 32.2 (1984), pp. 236–243.
- [55] Udit Gupta, Elliot Moore, and Alexander Lerch. "On the perceptual relevance of objective source separation measures for singing voice separation". In: *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2015, pp. 1–5.
- [56] Stefanie Heinicke, Ammie K Kalan, Oliver JJ Wagner, Roger Mundry, Hanna Lukashevich, and Hjalmar S Kühl. "Assessing the performance of a semi-automated acoustic monitoring system for primates". In: *Methods in Ecology and Evolution* 6.7 (2015), pp. 753–763.
- [57] Romain Hennequin, Bertrand David, and Roland Badeau. "Score Informed Audio Source Separation Using A Parametric Model Of Non-Negative Spectrogram". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 2011, pp. 45–48.
- [58] Po-Sen Huang, Minje Kim, Mark Hasegawa-Johnson, and Paris Smaragdis. "Singing-Voice Separation from Monaural Recordings using Deep Recurrent Neural Networks." In: *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. 2014, pp. 477–482.
- [59] Xuedong D. Huang, Yasuo Ariki, and Mervyn A. Jack. *Hidden Markov Models for Speech Recognition*. Edinburgh University Press, 1990.
- [60] Aapo Hyvärinen and Erkki Oja. "Independent component analysis: algorithms and applications". In: *Neural networks* 13.4-5 (2000), pp. 411–430.
- [61] Jacopo Iacovacci and Lucas Lacasa. "Visibility graphs for image processing". In: *IEEE Transactions in Pattern Analysis and Machine Intelligence* (Apr. 19, 2018).
- [62] JianPing Jing and Guang Meng. "A novel method for multi-fault diagnosis of rotor system". In: *Mechanism and machine theory* 44.4 (2009), pp. 697–709.

- [63] Tzzy-Ping Jung, Scott Makeig, Colin Humphries, Te-Won Lee, Martin J McKeown, Vicente Iragui, and Terrence J Sejnowski. "Removing electroencephalographic artifacts by blind source separation". In: *Psychophysiology* 37.2 (2000), pp. 163–178.
- [64] Maksim Kitsak, Lazaros K Gallos, Shlomo Havlin, Fredrik Liljeros, Lev Muchnik, H Eugene Stanley, and Hernán A Makse. "Identification of influential spreaders in complex networks". In: *Nature physics* 6.11 (2010), p. 888.
- [65] Lucas Lacasa and Jacopo Iacovacci. "Visibility graphs of random scalar fields and spatial data". In: *Physical Review E* 96 (2017), p. 012318.
- [66] Lucas Lacasa, Vincenzo Nicosia, and Vito Latora. "Network structure of multivariate time series". In: *Scientific reports* 5 (2015), p. 15508.
- [67] Lucas Lacasa and Raul Toral. "Description of stochastic and chaotic series using visibility graphs". In: *Physical Review E* 82 (3 2010), p. 036120.
- [68] Lucas Lacasa, Bartolo Luque, Fernando Ballesteros, Jordi Luque, and Juan Carlos Nuno. "From time series to complex networks: The visibility graph". In: *Proceedings of the National Academy of Sciences* 105.13 (2008), pp. 4972–4975.
- [69] Lucas Lacasa, Bartolome Luque, Ignacio Gómez, and Octavio Miramontes. "On a Dynamical Approach to Some Prime Number Sequences". In: *Entropy* 20.2 (2018).
- [70] Xin Lan, Hongming Mo, Shiyu Chen, Qi Liu, and Yong Deng. "Fast transformation from time series to visibility graphs". In: *Chaos: An Interdisciplinary Journal of Nonlinear Science* 25.8 (2015), p. 083105.
- [71] Vito Latora, Vincenzo Nicosia, and Giovanni Russo. *Complex networks: principles, methods and applications*. Cambridge University Press, 2017.
- [72] Jonathan Le Roux, Nobutaka Ono, and Shigeki Sagayama. "Explicit consistency constraints for STFT spectrograms and their application to phase reconstruction". In: *Proceedings of the ISCA Tutorial and Research Workshop on Statistical And Perceptual Audition (SAPA)*. 2008, pp. 23–28.

- [73] Jonathan Le Roux, Hirokazu Kameoka, Nobutaka Ono, and Shigeki Sagayama. "Fast Signal Reconstruction from Magnitude STFT Spectrogram Based on Spectrogram Consistency". In: *Proceedings of the International Conference on Digital Audio Effects (DAFx)*. 2010, pp. 397–403.
- [74] Jonathan Le Roux, JR Hershey, A Liutkus, F Stöter, ST Wisdom, and H Erdogan. "SDR-half-baked or well done?" In: *Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA, Tech. Rep* (2018).
- [75] Daniel D. Lee and H. Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization". In: *Nature* 401.6755 (1999), pp. 788–791.
- [76] Daniel D. Lee and H. Sebastian Seung. "Algorithms for Non-negative Matrix Factorization". In: *Advances in Neural Information Processing Systems*. 2000, pp. 556–562.
- [77] Simon Leglaive, Roland Badeau, and Gaël Richard. "Multi-channel Audio Source Separation With Probabilistic Reverberation Priors". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24.12 (2016), pp. 2453–2465.
- [78] Simon Leglaive, Romain Hennequin, and Roland Badeau. "Singing Voice Detection with Deep Recurrent Neural Networks". In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 2015, pp. 121–125.
- [79] Wei Liao, Jurong Ding, Daniele Marinazzo, Qiang Xu, Zhengge Wang, Cuiping Yuan, Zhiqiang Zhang, Guangming Lu, and Huafu Chen. "Small-world directed networks in the human brain: Multivariate Granger causality analysis of resting-state fMRI". In: *NeuroImage* 54.4 (2011), pp. 2683 –2694.
- [80] Antoine Liutkus, Roland Badeau, and Gérald Richard. "Gaussian Processes for Underdetermined Source Separation". In: *IEEE Transactions on Signal Processing* 59.7 (2011), pp. 3155 – 3167.
- [81] Antoine Liutkus, Derry Fitzgerald, and Zafar Rafii. "Scalable audio separation with light kernel additive modelling". In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2015, pp. 76–80.

- [82] Antoine Liutkus, Derry FitzGerald, Zafar Rafii, Bryan Pardo, and Laurent Daudet. "Kernel Additive Models for Source Separation". In: *IEEE Transactions on Signal Processing* 62.16 (2014), pp. 4298–4310.
- [83] Antoine Liutkus, Fabian-Robert Stöter, Zafar Rafii, Daichi Kitamura, Bertrand Rivet, Nobutaka Ito, Nobutaka Ono, and Julie Fontecave. "The 2016 Signal Separation Evaluation Campaign". In: *International Conference on Latent Variable Analysis and Signal Separation*. Springer. 2017, pp. 323–332.
- [84] Patricio Lopez-Serrano, Christian Dittmar, and Meinard Müller. "Mid-Level Audio Features Based on Cascaded Harmonic-Residual-Percussive Separation". In: *Audio Engineering Society Conference: 2017 AES International Conference on Semantic Audio*. 2017.
- [85] Bartolo Luque, Lucas Lacasa, Fernando Ballesteros, and Jordi Luque. "Horizontal visibility graphs: Exact results for random time series". In: *Physical Review E* 80.4 (2009), p. 046103.
- [86] Paul Magron, Roland Badeau, and Bertrand David. "Phase reconstruction of spectrograms with linear unwrapping: application to audio signal restoration". In: *2015 23rd European Signal Processing Conference (EUSIPCO)*. IEEE. 2015, pp. 1–5.
- [87] R.N. Mantegna. "Hierarchical structure in financial markets". In: *Eur. Phys. J. B* 11.1 (1999), pp. 193–197.
- [88] Daniele Marinazzo, Mario Pellicoro, and Sebastiano Stramaglia. "Kernel Method for Nonlinear Granger Causality". In: *Physical Review Letters* 100 (2008), p. 144103.
- [89] Brian McFee and Dan Ellis. "Analyzing song structure with spectral clustering". In: *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. 2014, pp. 405–410.
- [90] Brian McFee, Jong Wook Kim, Mark Cartwright, Justin Salamon, Rachel M Bittner, and Juan Pablo Bello. "Open-Source Practices for Music Signal Processing Research: Recommendations for Transparent, Sustainable, and Reproducible Audio Research". In: *IEEE Signal Processing Magazine* 36.1 (2018), pp. 128–137.

- [91] Dave Moffat and Mark Sandler. "Automatic Mixing Level Balancing Enhanced through Source Interference Identification". In: *Audio Engineering Society Convention 146*. Audio Engineering Society. 2019.
- [92] Meinard Müller. *Fundamentals of Music Processing*. Springer Verlag, 2015.
- [93] Nicoló Musmeci, Vincenzo Nicosia, Tomaso Aste, Tiziana Di Matteo, and Vito Latora. "The multiplex dependency structure of financial markets". In: *Complexity 2017* (2017).
- [94] Gautham J. Mysore, Paris Smaragdis, and Bhiksha Raj. "Non-negative hidden Markov modeling of audio with application to source separation". In: *Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*. 2010, pp. 140–148.
- [95] Arun Narayanan and DeLiang Wang. "Ideal ratio mask estimation using deep neural networks for robust speech recognition". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2013, pp. 7092–7096.
- [96] Aditya Arie Nugraha, Antoine Liutkus, and Emmanuel Vincent. "Multichannel audio source separation with deep neural networks". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing 24.9* (2016), pp. 1652–1664.
- [97] Angel M Nuñez, Lucas Lacasa, Jose Patricio Gomez, and Bartolo Luque. "Visibility algorithms: A short review". In: *New Frontiers in Graph Theory*. InTech, 2012.
- [98] Angel Nuñez, Lucas Lacasa, Eusebio Valero, Jose Patricio Gómez, and Bartolo Luque. "Detecting series periodicity with horizontal visibility graphs". In: *International Journal of Bifurcation and Chaos 22.07* (2012), p. 1250160.
- [99] Ken O'Hanlon, Hidehisa Nagano, Nicolas Keriven, and Mark D. Plumbley. "Non-Negative Group Sparsity with Subspace Note Modelling for Polyphonic Transcription". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing 24.3* (2016), pp. 530–542.

- [100] Alexey Ozerov, Emmanuel Vincent, and Frédéric Bimbot. "A general flexible framework for the handling of prior information in audio source separation". In: *IEEE Transactions on Audio, Speech, and Language Processing* 20.4 (2012), pp. 1118–1133.
- [101] Alexey Ozerov, Cédric Févotte, Raphaël Blouet, and Jean-Louis Durrieu. "Multichannel nonnegative tensor factorization with structured constraints for user-guided audio source separation". In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*. 2011, pp. 257–260.
- [102] Fatemeh Pishdadian, Bryan Pardo, and Antoine Liutkus. "A multi-resolution approach to common fate-based audio separation". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 566–570.
- [103] Matthieu Puigt, Emmanuel Vincent, and Yannick Deville. "Validity of the independence assumption for the separation of instantaneous and convolutive mixtures of speech and music sources". In: *Proceedings of the International Conference on Independent Component Analysis and Signal Separation (ICA)*. 2009, pp. 613–620.
- [104] Lawrence Rabiner and Bing-Hwang Juang. *Fundamentals of Speech Recognition*. Prentice Hall Signal Processing Series, 1993.
- [105] Miloš Radovanović, Alexandros Nanopoulos, and Mirjana Ivanović. "Nearest neighbors in high-dimensional data: The emergence and influence of hubs". In: *Proceedings of the 26th Annual International Conference on Machine Learning*. ACM. 2009, pp. 865–872.
- [106] Zafar Rafii and Bryan Pardo. "Music/Voice Separation Using the Similarity Matrix." In: *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*. 2012, pp. 583–588.
- [107] Zafar Rafii and Bryan Pardo. "Online REPET-SIM for real-time speech enhancement". In: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE. 2013, pp. 848–852.

- [108] Zafar Rafii and Bryan Pardo. "REpeating Pattern Extraction Technique (REPET): A Simple Method for Music/Voice Separation." In: *IEEE Transactions on Audio, Speech, and Language Processing* 21.1 (2013), pp. 71–82.
- [109] Zafar Rafii, Antoine Liutkus, Fabian-Robert Stoter, Stylianos Ioannis Mimalakis, Derry FitzGerald, and Bryan Pardo. "An overview of lead and accompaniment separation in music". In: *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 26.8 (2018), pp. 1307–1335.
- [110] Christopher Raphael. "Automatic Segmentation of Acoustic Musical Signals Using Hidden Markov Models". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 21.4 (1998), pp. 360–370.
- [111] Shoichiro Saito, Hirokazu Kameoka, Keigo Takahashi, Takuya Nishimoto, and Shigeki Sagayama. "Spectrum analysis of polyphonic music signals". In: *IEEE Transactions on Audio, Speech, and Language Processing* 16.3 (2008), pp. 639–650.
- [112] Speranza Sannino, Sebastiano Stramaglia, Lucas Lacasa, and Daniele Marinazzo. "Visibility graphs for fMRI data: Multiplex temporal graphs and their modulations across resting-state networks". In: *Network Neuroscience* 1.3 (2017), pp. 208–221.
- [113] Mikkel N. Schmidt and Morten Mørup. "Nonnegative matrix factor 2-D deconvolution for blind single channel source separation". In: *Proceedings of the International Conference on Independent Component Analysis and Blind Signal Separation*. Springer Berlin/Heidelberg, 2006, pp. 700–707.
- [114] Christian Schörkhuber, Anssi Klapuri, Nicki Holighaus, and Monika Dörfler. "A Matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution". In: *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society. 2014.
- [115] Prem Seetharaman, Fatemeh Pishdadian, and Bryan Pardo. "Music/Voice separation using the 2D fourier transform". In: *Proceedings of the IEEE Workshop on Applications of Signal Pro-*

- cessing to Audio and Acoustics (WASPAA)*. IEEE. 2017, pp. 36–40.
- [116] Prem Seetharaman and Zafar Rafii. “Cover song identification with 2d fourier transform sequences”. In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017, pp. 616–620.
 - [117] Xavier Serra. “A System for Sound Analysis/Transformation/Synthesis based on a Deterministic plus Stochastic Decomposition”. PhD thesis. Stanford University, 1989.
 - [118] Xavier Serra. “Musical Sound Modeling With Sinusoids plus Noise”. In: *Musical Signal Processing*. Swets & Zeitlinger, 1997.
 - [119] A H Shirazi, G Reza Jafari, J Davoudi, J Peinke, M Reza Rahimi Tabar, and Muhammad Sahimi. “Mapping stochastic processes onto complex networks”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2009.07 (2009).
 - [120] Paris Smaragdis. “User guided audio selection from complex sound mixtures”. In: *Proceedings of the ACM Symposium on User Interface Software and Technology (UIST)*. 2009, pp. 89–92.
 - [121] Paris Smaragdis and Judith C. Brown. “Non-Negative Matrix Factorization for Polyphonic Music Transcription”. In: *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. 2003, pp. 177–180.
 - [122] Paris Smaragdis and Gautham J. Mysore. “Separation by Humming: User Guided Sound Extraction from Monophonic Mixtures”. In: *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*. New Paltz, NY, USA, 2009, pp. 69–72.
 - [123] Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka. “Sparse and shift-invariant feature extraction from non-negative data”. In: *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP*. 2008, pp. 2069–2072.
 - [124] Daniel Stoller, Sebastian Ewert, and Simon Dixon. “Wave-u-net: A multi-scale neural network for end-to-end audio source separation”. In: *19th International Society for Music Information Retrieval Conference (ISMIR)* (2018).

- [125] Fabian-Robert Stöter, Antoine Liutkus, and Nobutaka Ito. "The 2018 signal separation evaluation campaign". In: *International Conference on Latent Variable Analysis and Signal Separation*. Springer. 2018, pp. 293–305.
- [126] Fabian-Robert Stöter, Antoine Liutkus, Roland Badeau, Bernd Edler, and Paul Magron. "Common fate model for unison source separation". In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2016, pp. 126–130.
- [127] Fabian-Robert Stöter, Stefan Uhlich, Antoine Liutkus, and Yuki Mitsufuji. "Open-Unmix - A Reference Implementation for Music Source Separation". In: *Journal of Open Source Software* (2019).
- [128] Dan Stowell, Dimitrios Giannoulis, Emmanouil Benetos, Mathieu Lagrange, and Mark D Plumbley. "Detection and classification of acoustic scenes and events". In: *IEEE Transactions on Multimedia* 17.10 (2015), pp. 1733–1746.
- [129] Iván González Torre, Bartolo Luque, Lucas Lacasa, Jordi Luque, and Antoni Hernández-Fernández. "Emergence of linguistic laws in human voice". In: *Scientific reports* 7 (2017), p. 43862.
- [130] R Romo Vázquez, Hugo Velez-Perez, Radu Ranta, V Louis Dorr, Didier Maquin, and Louis Maillard. "Blind source separation, wavelet denoising and discriminant analysis for EEG artefacts and noise cancelling". In: *Biomedical Signal Processing and Control* 7.4 (2012), pp. 389–400.
- [131] Emmanuel Vincent. "Improved Perceptual Metrics for the Evaluation of Audio Source Separation". In: *Proceedings of the International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA)*. 2012, pp. 430–437.
- [132] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte. "Performance measurement in blind audio source separation". In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.4 (2006), pp. 1462–1469.
- [133] Emmanuel Vincent, Tuomas Virtanen, and Sharon Gannot. *Audio source separation and speech enhancement*. John Wiley & Sons, 2018.

- [134] Tuomas Virtanen. "Monaural Sound Source Separation by Non-negative Matrix Factorization With Temporal Continuity and Sparseness Criteria". In: *IEEE Transactions on Audio, Speech and Language Processing* 15.3 (2007), pp. 1066–1074.
- [135] Duncan J Watts and Steven H Strogatz. "Collective dynamics of 'small-world' networks". In: *Nature* 393.6684 (1998), p. 440.
- [136] Douglas Brent West et al. *Introduction to graph theory*. Vol. 2. Prentice hall Upper Saddle River, NJ, 1996.
- [137] Kevin W Wilson, Bhiksha Raj, Paris Smaragdis, and Ajay Divakaran. "Speech denoising using nonnegative matrix factorization with priors". In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2008, pp. 4029–4032.
- [138] Wei Xu, Xin Liu, and Yihong Gong. "Document clustering based on non-negative matrix factorization". In: *Proceeding of the International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*. 2003, pp. 267–273.
- [139] Yue Yang and Huijie Yang. "Complex network-based time series analysis". In: *Physica A: Statistical Mechanics and its Applications* 387.5 (2008), pp. 1381 –1386.