FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

# Musical Cross-Synthesis using Matrix Factorisation

**Francisco Daniel Andrade Fonseca**

Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Supervisor: Matthew E. P. Davies

July 18, 2019

# Resumo

O aumento do poder de processamento computacional conjugado com avanços nos algoritmos de *machine learning*, levou a que músicos e entusiastas começassem a usar sistemas baseados em inteligência artificial no processo de criação musical. Para além disto, algumas das ferramentas fornecidas pelo domínio de investigação de *Music Information Retrieval* já são amplamente usadas especialmente no contexto de música eletrónica e performances ao vivo. O sistema aqui apresentado procura explorar estas duas áreas com o propósito de criar uma *framework* flexível para *musical cross-synthesis*.

O foco principal é então desenvolver um novo método para a análise e manipulação criativa de conteúdo musical. Dada uma música alvo e uma música fonte, o objetivo é reconstruir a estrutura harmónica e rítmica do alvo usando componentes tímbricos da fonte, de forma a que quando combinados, o alvo e a fonte produzam um resultado sonoro agradável. Para tal, foi usado um algoritmo de *matrix factorisation*, mais especificamente, o algoritmo *Shift-Invariant Probabilistic Latent Component Analysis* (PLCA). Essencialmente, as funções base da fonte, neste caso, *beat-synchronous Constant-Q Transform* (CQT) *vectors*, são usadas para calcular as matrizes de activações que melhor aproximam o alvo, representado por um *beat-synchronous* CQT *spectrogram*. A propriedade de invariância ao deslocamento da PLCA permite que cada função base possa ser transposta musicalmente, aumentando assim a flexibilidade da fonte para reconstruir o alvo. Para obter o resultado de *cross-synthesis*, as funções base, já transpostas, são invertidas para o domínio dos tempos e concatenadas adequadamente.

A avaliação do sistema foi feita através de um *listening test* que compara os resultados obtidos usando três abordagens diferentes. Duas delas têm como base o algoritmo PLCA, fazendo no entanto uso de diferentes representações para a fonte e o alvo, uma usa a CQT e a outra o *Chromagram*. A terceira abordagem, é uma implementação mais simples que usa uma medida de distância Euclidiana para determinar as funções base da fonte mais adequadas para reconstruir o alvo (fonte e alvo representados por um *Chromagram*). Isto é feito com o propósito de avaliar a qualidade subjetiva dos resultados, o uso do algoritmo PLCA comparado a uma abordagem mais simples e o uso da CQT comparada com o uso do *Chromagram*. Para além disto, uma representação musical obtida quando a fonte é apenas uma nota e o alvo uma melodia simples foi também avaliada.

Os principais resultados mostram que a qualidade das combinações obtidas está altamente relacionada com a coerência da estrutura da reconstrução do alvo. Foi também concluído que o uso da PLCA conjugada com uma representação complexa do alvo e da fonte, como a CQT, melhora a estrutura harmónica e temporal da reconstrução. Relativamente à representação musical, esta mostrou ser suficientemente clara e adequada para representar melodias simples.

# Abstract

The development of computing processing power in conjunction with advances in machine learning algorithms has led to musicians and enthusiasts using systems based on artificial intelligence to enhance and expand their musical creation process. Besides this, some of the tools provided by Music Information Retrieval are already widely used especially in electronic genres and live performances. The system presented here seeks to explore both of these fields in order to a offer a flexible framework for musical cross-synthesis.

The main focus is to develop a new method for the creative analysis and manipulation of musical audio content. Given a target song and a source song, the goal is reconstruct the harmonic and rhythmic structure of the target with the timbral components from the source, in such a way that the target and the source create a pleasant mix when combined. For this purpose, we propose the use of a matrix factorisation method, more specifically, Shift-Invariant Probabilistic Latent Component Analysis (PLCA). The PLCA algorithm uses beat-synchronous Constant-Q Transform (CQT) basis functions of the source to calculate the activation matrices that best approximate the beat-synchronised CQT of the target. The shift-invariant property of the PLCA allows each basis function to be subjected to a range of possible pitch shifts which increases the flexibility of the source to represent the target. To create the resulting musical cross-synthesis the beat synchronous, pitch-shifted CQT basis functions are inverted and concatenated in time.

The evaluation of the system was done using a listening test that compared the results obtained using three different approaches, two using the PLCA but with different input representations, the CQT and the Chromagram, and the other one being a simpler implementation that uses a Euclidean distance measure to find the basis functions, represented by a Chromagram, that best match the target. This is done with the purpose of evaluating the subjective quality of the results, the use of the PLCA against a simpler approach and the use of the CQT compared with the Chromagram. Also, a musical representation obtained when the source is a single note and the target a monophonic melody is also evaluated.

The main results show that the quality of the obtained mix is highly related to the coherence of the internal structure of the reconstruction. Furthermore, it has been shown that using the PLCA with a complex input representation, such as the CQT, improves both the temporal and harmonic structure of the reconstruction. The musical representation has also shown to be clear and understandable enough to represent simple melodies for musically trained listeners.

# Agradecimentos

Antes de tudo, gostaria de agradecer aos meus pais por todo o amor, apoio e paciência incondicional que deram ao longo de toda a minha vida, amor, apoio e paciência que espero um dia retribuir.

À Zeza, por todo o amor, carinho, dedicação, por me ter aturado e compreendido sempre que preciso e por me fazer ver o mundo de uma forma mais bonita.

Ao Matthew, pela supervisão, dedicação e entusiasmo pelo tema que me motivou a explorar novas ideas e pela oportunidade de trabalhar numa área que me dá muito prazer. Ao Professor Rui Penha, que despertou em mim o interesse pela ligação entre música e tecnologia.

A todos os elementos do Sound and Music Computing, por me terem feito sentir imediatamente integrado.

A todos os meus amigos que me foram acompanhando ao longo dos anos: Bababa, Toni, Ramalho, Tufa, Mikes, Fonseca, Pedro, Adães, Té e Marcelo.

Francisco Fonseca

*“The important thing about the relationship
between music and technology is that it’s entirely circular.”*

Brian Eno

# Contents

**1 Introduction** **1**
1.1 Context . . . 1
1.2 Motivation . . . 1
1.3 Objectives . . . 2
1.4 Report Structure . . . 2
1.5 Publication Resulting from this Dissertation . . . 3
1.6 Sound Examples . . . 3

**2 Background and State of the Art** **5**
2.1 Music and Technology . . . 5
2.1.1 Music Information Retrieval . . . 5
2.1.2 Creative MIR . . . 6
2.2 Representations of Musical Audio Signals . . . 7
2.3 Non-Negative Matrix Factorisation . . . 10
2.4 Recombination of Audio Signals . . . 13
2.5 Conclusions . . . 17

**3 Approach** **19**
3.1 Problem Definition . . . 19
3.2 Pre-Processsing . . . 21
3.2.1 Input Representation . . . 21
3.2.2 Beat Processing . . . 22
3.2.3 Beat-Synchronisation . . . 24
3.3 Probabilistic Latent Component Analysis . . . 24
3.3.1 Shift-Invariance . . . 25
3.3.2 Input Parameters . . . 26
3.4 Post Processing . . . 27
3.4.1 Activation Filtering . . . 27
3.4.2 Signal Reconstruction . . . 28
3.4.3 Beat-by-Beat RMS Normalisation . . . 31
3.5 GUI . . . 32

**4 Evaluation** **37**
4.1 Experiment Design . . . 37
4.1.1 Data Preparation . . . 38
4.1.2 Listening Test . . . 40
4.2 Results . . . 40
4.3 Discussion . . . 41

**5 Conclusion and Future Work 47**

5.1 Summary . . . . . 47

5.2 Future Work . . . . . 48

5.3 Perspective on the project . . . . . 49

**References 51**

# List of Figures

2.1 Shepard's helix of pitch perception. Taken from [31] . . . . . . . . . . . . . . . . 9
2.2 Music sheet of the vibraphone recording. . . . . . . . . . . . . . . . . . . . . . . . 9
2.3 STFT vs CQT Spectrogram. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
2.4 Chromagram. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 10
2.5 NMF example. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 12
2.6 *CataRT* interface. Taken from [41]. . . . . . . . . . . . . . . . . . . . . . . . . . 14
2.7 *Factorsynth* interface in cross-synthesis mode. Figure extracted from [44]. . . . . 16

3.1 Flowchart of the proposed solution. . . . . . . . . . . . . . . . . . . . . . . . . . 20
3.2 Chromagram of a recording of an electric piano playing a melody. . . . . . . . . 22
3.3 CQT spectrogram of a recording of an electric piano playing a melody. . . . . . . 22
3.4 Example of beat correction. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
3.5 Beat-synchronous CQT vs Beat-by-Beat beat-synchronous CQT. . . . . . . . . . 25
3.6 Effect of circular-shifting the basis functions on the activation matrix. . . . . . . 26
3.7 Example of the PLCA algorithm output. . . . . . . . . . . . . . . . . . . . . . . 28
3.8 Activation matrix before (left) and after (right) the filtering of the number simultaneous activations. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 29
3.9 Example of the pitch-shifts mapping and filtering. Each graphic represents the pitch-shift values associated with each basis function. . . . . . . . . . . . . . . . 30
3.10 Example of tuning correction. . . . . . . . . . . . . . . . . . . . . . . . . . . . . 32
3.11 Peak normalisation vs RMS normalisation vs Beat-by-Beat RMS normalisation. . 33
3.12 Graphical User Interface. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 35

4.1 Musical score of the options presented in the last question. . . . . . . . . . . . . 39
4.2 Musical representation presented in the last question. . . . . . . . . . . . . . . . 39
4.3 Ranking System . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
4.4 Chart graph of Table 4.1 and 4.2. . . . . . . . . . . . . . . . . . . . . . . . . . . 41
4.5 Chart graph of Table 4.3 and 4.4. . . . . . . . . . . . . . . . . . . . . . . . . . . 42
4.6 Last question results. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 43
4.7 Relationship between the mix ranking and the structure ranking. . . . . . . . . . 45

# List of Tables

4.1 Mean ranking by mix. . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 40
4.2 Overall mean of the mixes and standard deviation. . . . . . . . . . . . . . . . . . . 41
4.3 Mean ranking by reconstruction. . . . . . . . . . . . . . . . . . . . . . . . . . . 42
4.4 Overall mean of the reconstructions and standard deviation. . . . . . . . . . . . . 42

# Abreviaturas e Símbolos

| | |
|---|---|
| CQT | Constant-Q Transform |
| NMF | Non-Negative Matrix Factorisation |
| FFT | Fast Fourier Transform |
| DFT | Discrete Fourier Transform |
| DGT | Discrete Gabor Transform |
| STFT | Short-Time Fourier Transform |
| MIR | Music Information Retrieval |
| DAW | Digital Audio Workstation |
| BPM | Beats per Minute |
| MIDI | Musical Instrument Digital Interface |
| 12-TET | 12-Tone Equal Temperament |
| PLCA | Probabilistic Latent Component Analysis |
| RPCA | Robust Principle Component Analysis |

# Chapter 1

# Introduction

## 1.1 Context

Artificial intelligence algorithms are finding their ways into our lives in very direct ways, for instance, in self-driving cars, and in less obvious and more intrusive ways like online advertising recommendations. Many of these, such as context-aware voice recognition software, e.g. Siri[1] by Apple or Cortana[2] by Microsoft, have the goal of simplifying and enhancing certain modern life tasks. Regarding music applications, machine learning can be used to enhance the music creation process as a composition tool, making use of Markov Chains [1] [2] [3], as a recommendation system, useful for music streaming platforms like Spotify[3] [4], and for music content recombination, [5] [6]. The interest in having a computer understanding, creating and playing music is not new as it can be seen by examples from the late 1940s/1950s such as the CSIRAC computer and the computer music language MUSIC created by Max Mathews in 1954 at the Bell Labs [7]. Algorithmic advances, e.g. the Fast Fourier Transform (FFT), which was introduced in 1965 [8], allowed more efficient computer analysis of real-world signals, such as musical ones, laying the foundation for information retrieval and manipulation of musical content. More powerful successors to Mathews's MUSIC like Max[4] and the similar open-source Pure Data[5] were introduced by Miller Puckette in the 1980s/1990s. A more recent example is Magenta[6] by Google, "*An open source research project exploring the role of machine learning as a tool in the creative process*", which makes use of TensorFlow [9], a machine learning framework.

## 1.2 Motivation

The task of repurposing musical content done totally manually using tools such as a Digital Audio Workstation (DAW) is a quite slow and hard process, so tools for automatic mixing such as

[1] https://www.apple.com/siri/
[2] https://www.microsoft.com/en-us/cortana
[3] https://www.spotify.com
[4] https://cycling74.com/products/max/
[5] https://puredata.info/
[6] https://magenta.tensorflow.org/

harmonic mixing, something already offered in some commercial software, e.g. *Traktor*[7], or for automatic mashup creation can attract big audiences as electronic music is one of the most popular genres today. This work proposes to explore ways of retrieving relevant musical information from audio signals using music signal processing and recent machine learning techniques in order to produce new ways of musical cross-synthesis, that is, to blend the characteristics of two or more sounds, with the purpose of creating tools to aid creative musical content repurposing.

## 1.3 Objectives

The main goal of this research is to devise new methods of reconstructing the harmonic and rhythmic structure of a certain target song with the timbral components of a source song, in such a way that both the target can be mixed together to create musically interesting results. From a technical perspective the research will address the decomposition of source and target excerpts into basis functions and respective activations using matrix factorisation methods, with cross-synthesis being the result of reconstructing the target using the basis functions of the source. To achieve this goal a list of objectives was defined. These are:

- Compile a dataset of musical recordings for use in the project.
- Determine the best ways of representing musical content in this context.
- Devise an approach for musical cross-synthesis using matrix factorisation.
- Investigate the use of content transformation, through techniques of pitch-shifting and time-stretching, for more flexible cross-synthesis.
- Explore techniques for the incorporation of user-input into the workflow.
- Design and execute a subjective listening experiment on the obtained results.

## 1.4 Report Structure

This report contains four chapters besides the introduction one presented now. Chapter 2 contains a review regarding the state of the art. In Chapter 3 the challenges of this project are described and a solution to these challenges is proposed. Chapter 4 presents the approach used to evaluate the system and a discussion on the obtained results. Finally, Chapter 5 summarises the contributions and the proposed future work.

[7] https://www.native-instruments.com/en/products/traktor/

## 1.5 Publication Resulting from this Dissertation

This dissertation led to the presentation of the following paper:

- Francisco Fonseca, Matthew E. P. Davies. "Musical Cross-Synthesis using Matrix Factorisation" in *DCE19 - 3rd Doctoral Congress in Engineering*, 2019. Winner of the best oral presentation in the DCE Electrical and Computing Engineering Symposium.

.

## 1.6 Sound Examples

A set of cross-synthesis examples obtained in the context of this work can be found at `https://web.fe.up.pt/~up201403798/MatFac/`.

# Chapter 2

# Background and State of the Art

## 2.1 Music and Technology

Music has a history of being connected with the state of the art of technology present in each era. Ancient Egyptians, around 3000BC, developed stringed instruments such as harps and lyres, which at the time were very advanced mechanisms [10, Ch.2]. Around the same time Ancient Greeks invented the hydraulic organ, the predecessor to the pipe organ. Instruments like the harpsichord and new ways of music notation were developed during the Middle Ages. In the Baroque era (1600–1750) advances in keyboard technology allowed for improvements in existing instruments such as the pipe organ and for the creation of new ones, for instance, the piano [10, Ch.3]. Inventions of the late 1800s and early 1900s such as microphones, the phonograph, and tape recorders allowed for the music exploration that would come. *Musique concrète* born in the *Studio d'Essai* in the late 1940s is a genre that explores the manipulation of recorded sounds, using inventive techniques, such as tape looping and splicing. *Elektronische Musik* as described by Meyer-Eppler in his thesis in 1949 [11], is the creation of purely electronic generated music, differing from *musique concrète* which used sounds from acoustical sources. In the 1960s the use of electronic instruments in popular music, e.g the use of the Electro-Theremin, invented in the late 1950s, in Good Vibrations by the Beach Boys [12], laid the foundation for the uprising of electronic music that happened during the following decades. New methods of sound synthesis, such as frequency modulation synthesis developed by Chowing [13], introduced a new palette of possible sounds present in synthesizers, drums machines and vocoders made available to the public during the 1970s. The increase of computing power democratised the ability to produce music, making it possible to have a musical studio at home with just a computer. Since then, new ways of manipulating audio, like sampling, and ways of interaction, such as MIDI, provoked the emergence of new genres while changing the paradigm of the existing ones [14] [15].

### 2.1.1 Music Information Retrieval

Music Information Retrieval (MIR) is an interdisciplinary research field that gathers people with very distinct backgrounds such as musicology, computer science, psychology, with the focus of retrieving meaningful information from music. As described in [16] some of the typical sub-fields of MIR research are: Feature Extraction, Similarity, Classification and Applications. Feature Extraction tackles the extraction of relevant features from music content, including tasks such as:

- Harmonic/Melodic Analysis: tonality estimation, music transcription, melody extraction and structural analysis.
- Rhythm Analysis: onset detection, beat tracking and tempo estimation.
- Sound Analysis: timbre description.

Similarity normally takes advantage of databases of song features and distance measures in order to find similarities between different songs, or chunks of audio. Classification, uses both feature extraction and similarity analysis to classify genre, instruments, artists and albums. This allows for applications such as audio identification or fingerprinting that can quickly identify a fragment of music (e.g Shazam[8]), automatic playlist generation and music recommendation (e.g Spotify[9]).

The most relevant topics for this project are in the sub-fields of feature extraction and similarity, as the goals of the project require performing tasks such as harmonic analysis, structural analysis and beat tracking using signal processing as well as to find similarities between two pieces of music using machine learning techniques.

### 2.1.2 Creative MIR

One other sub-field that is also going to be in focus in this project, which is becoming more relevant and is the creative use of MIR [17]. This has the aim of creating systems that can retrieve information from musical pieces and then recombine it in some musically creative way. Some of the topics being researched recently, as mentioned in [17] [18], are:

- Automatic Mash-Up systems: Examples include *AutoMashUpper* [19] , *Beat-Sync-Mash-Coder* [20].
- Education Tools: Such as *Yousician*[10] that allows for more interaction between the user and the learning process.
- Automatic Accompaniment: *OMax*, by Assayag [21], *Audio Oracle* by Dubnov [22].
- New Interfaces for Musical Expression: *Wekinator*[11] [23], a simple machine learning platform.

Creative MIR also explores how user interaction can enhance the music creation process while trying to maintain accessibility for musicians. An extensive study on the place of the user in the

[8] https://www.shazam.com
[9] https://www.spotify.com
[10] https://yousician.com
[11] http://www.wekinator.org

MIR process as well as the challenges that personalization faces in this kind of systems (Intelligent Information Systems) is presented in [24]. In [25] a discussion about the current state of the evaluation of user-centric MIR is made.

## 2.2 Representations of Musical Audio Signals

Classic techniques such as the FFT, although powerful tools to perform audio signal analysis, do not offer the best possible representation in the frequency dimension of musical content, due to the FFT having a constant resolution throughout the whole range of frequencies. This is not ideal in the context of the 12-tone equal temperament (12-TET) as in low frequencies the difference between 2 semitones in frequency is a lot smaller that in the upper range of frequencies. For example, knowing that the frequency of a note (using MIDI note numbers) in the 12-TET system can be obtained using the following equation (2.1),

$$P_n = 2^{(d-69)/12}.P_a \tag{2.1}$$

where:

$d$ = MIDI note number
$P_a$ = 440Hz (the reference pitch (A4), with its MIDI note number being 69)

the difference between C1 and C#1 is $\approx$ 2Hz while the difference 5 octaves above, between C6 and C#6, is $\approx$ 62Hz.

It is in this context that the Constant-Q Transform (CQT) was introduced by Brown [26], being calculated as follows:

$$X[k] = \frac{1}{N[k]} \sum_{n=0}^{N[k]-1} J[k,n]x[n]e^{-j2\pi Qn/N[k]} \tag{2.2}$$

$J[k,n]$ = Window function, in this case a Hamming window
$x[n]$ = The $n$th sample of the digitised temporal function

This transform follows an auditory perception approach as it offers logarithmic spacing in frequency, with a variable resolution and a variable window size that depend on the centre frequency of each frequency bin, as presented below:

$$f_k = (2^{1/o})^k f_{min} \tag{2.3}$$

$$N[k] = \frac{SR.Q}{f_k} \tag{2.4}$$

$$\Delta f = \frac{f_k}{Q} \tag{2.5}$$

where:

$f_k$ = frequency of the $k^{th}$ spectral component
$o$ = number of bins per octave (usually a multiple a 12)
$f_{min}$ = lowest frequency detectable
$N[k]$ = length of the window in samples at frequency $f_k$
$SR$ = sampling rate
$Q$ = frequency to resolution ratio
$\Delta f$ = resolution

Therefore the CQT allows for a better representation of musical audio signals compared to the (linear frequency) FFT, by allowing the observation of patterns such as the harmonics of each note in a clearer way [26]. However, the longstanding problem of the CQT was it being a non-invertible transform due to the temporal decimation being greater than the window size in the high frequency range, which meant that some frequencies were never analysed. Thus, in recent years several approaches tried to solve this constraint. Velasco et al. [27] presented a way of constructing an invertible-CQT using non-stationary Gabor Frames. In [28], an implementation of Gabor frames with time-frequency resolution evolving over time or frequency is proposed and used in solution presented in [27].

Another important property of the invertible CQT is that pitch-shifting can be easily implemented. In [29], it is explained that the frequency of a spectral peak can be scaled by a factor $\alpha$ by shifting the corresponding CQT bin by $m$ CQT bins, as described in equation (2.6), where $B$ is the number of bins per octave.

$$m = Blog_2(\alpha) \tag{2.6}$$

This means that to pitch-shift a complete signal a simple *linear* shift of all the CQT bins by the same amount is enough, which does not happen with the ordinary Short-Time Fourier Transform (STFT). However, to retain horizontal and vertical phase coherence all the coefficients within the same region need to be multiplied with the exponential $Z_u = e^{j\Delta f_{k,p} H_k}$ where $f_{k,p}$ is the difference between the centre frequencies of the old and the new peak bin in the frame $p$ of the CQT and $H_k$ the hop size between frames. The applied phase rotations need to be accumulated from one frame to the next [29].

Another way of representing musical signals is by using the Harmonic Pitch Class Profiles (HPCP), that allows to decompose audio signals in 12 pitch classes, also known as chroma. The Chromagram, as described in [30] represents the intensities of the 12 semitones, the pitch classes, over time.

$$s(t,c) = G(s(t,f); \forall f = 2^{c+h}) \tag{2.7}$$

Equation (2.7), from [30], describes a possible implementation to obtain the Chromagram, $s(t,c)$. $s(t,f)$ is a time frequency representation of the input signal, like a spectrogram, where $t$ represents time, $f$ frequency, $c$ chroma value, $c \subset [0,1)$, and $h$ tone height, $h \in \mathbb{Z}$. $c$ and $h$ are used

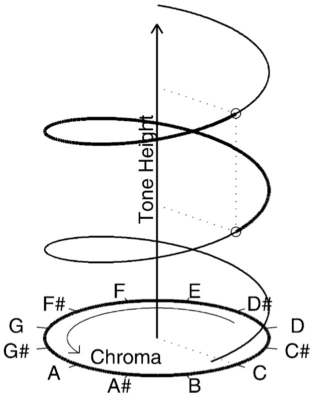


Figure 2.1: Shepard's helix of pitch perception. Taken from [31]

to represent the perceptual structure of pitch, proposed by Shepard [31], represented in Figure 2.1, where $c$ is divided into twelve equal intervals (the twelve tones in the 12-TET) and $h$ represents the tone height or, in musical terms, the octave number. $G$ is an aggregation function, usually a summation. In $G$ a log-warping is also considered necessary to map from linear frequency to chroma. This means, e.g, that the energy of each frequency that represents the musical note $A$, $A_2 = 110Hz, A_3 = 220Hz, A_4 = 440Hz$ will be summed to obtain the $A$ component of the chroma representation. These more compact representations are often used in harmonic analysis, being useful for tasks such as chord estimation, key estimation and melody extraction.

The semitone spectrogram, an adaptation of the more common spectrogram, also uses a vertical logarithmic scale to better represent a musical signal. In [32] a way to calculate a log-frequency spectrum is described. First a simple DFT with a particular length that resolves most musical intervals in the bass region is applied to a signal. Then a spectrogram is calculated and the each magnitude obtained is mapped onto bins that correspond to pitch.

Now, different representations of a vibraphone playing the note A in different octaves, from A3 (220Hz) to A7 (7040Hz) (Figure 2.2), will be presented.

It is possible to observe that in the STFT spectrogram (Figure 2.3a), the interval between octaves doubles at every octave. However, in the CQT spectrogram (Figure 2.3b) every octave interval is equal, providing a much appropriate representation of musical signals, as stated before. In the Chromagram (Figure 2.4), as the note being is played is always the same it is possible to observe that almost all of the energy is concentrated in the pitch class that represents the A note although in the last octaves (higher frequencies) some energy is spread into the harmonics.

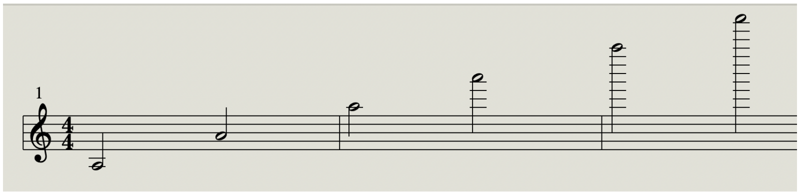
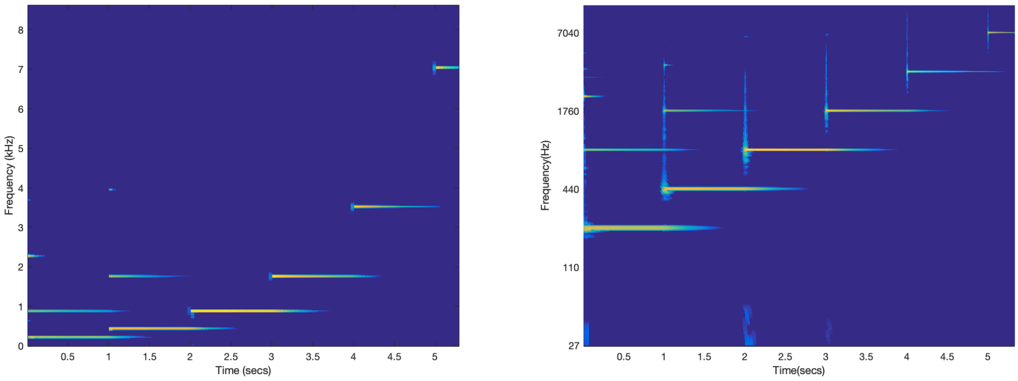


Figure 2.2: Music sheet of the vibraphone recording.



(a) STFT Spectrogram

(b) CQT Spectrogram

Figure 2.3: STFT vs CQT Spectrogram.

Now that the main musical representations have been introduced, a description of how machine learning algorithms can retrieve meaningful information from them using MIR concepts is going to be presented.

## 2.3 Non-Negative Matrix Factorisation

Non-Negative Matrix Factorisation (NMF) is a method that factorises an $n \times m$ matrix ,$V$, into two matrices, $W$ and $H$, so that $V \approx WH$, introduced by Lee and Seung [33]. The dimensions of these matrices are $n \times r$ and $r \times m$ respectively, $r$ being the rank of the factorisation, usually smaller than $n$ or $m$ so that the product $WH$ represents a compressed version of $V$. Each row of $W$ can be seen as a basis function and the correspondent column in $H$ represents its respective weight/activation. There is a constraint of non-negativity, meaning that $V,W$and $H$ don't have negative values, so multiple basis functions can be combined as only additive combinations are allowed. To evaluate the accuracy of the approximations obtained form $W.H$ a cost function is used. In [34], Lee and

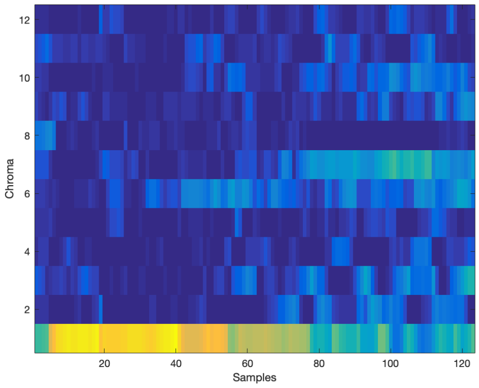


Figure 2.4: Chromagram.

Seung give two examples of possible ways of implementing cost functions. One is to measure the Euclidean distance between $V$ and $WH$, which is going to be referred as $V_{apx}$.

$$||V - V_{apx}|| = \sum_{ij}(V_{ij} - V_{apx,ij})^2 \tag{2.8}$$

For this cost function Lee and Seung defined in the same paper the respective update rule for $H$ (2.9) and $W$ (2.10), known as the *multiplicative update rule*, that is still one of the most popular approaches due to its simplicity.

$$H_{\alpha\mu} \leftarrow H_{\alpha\mu}\frac{(W^TV)_{\alpha\mu}}{(W^TWH)_{\alpha\mu}} \tag{2.9}$$

$$W_{i\alpha} \leftarrow W_{i\alpha}\frac{(VH^T)_{i\alpha}}{(WHH^T)_{i\alpha}} \tag{2.10}$$

This rule works because as both $W$ and $H$ get to a stationary point the Euclidean distance and the cost function become invariant. A more recent approach to the update rules is described in [35], where an alternative is based around an additive update algorithm and is stated to have a faster convergence compared to the multiplicative algorithm.

The other approach to the cost function presented in [34], is one that can be reduced to the Kullback-Leibler divergence, so that $V$ and $V_{apx}$ can be seen as normalised probability distributions.

An implementation of NMF that uses a probabilistic model is presented in [36], in which the basic model used is the Probabilistic Latent Component Analysis (PLCA) defined by:

$$P(x) = \sum_z P(z)\prod_{j=1}^{N} P(x_j|z_i) \tag{2.11}$$

Paraphrasing from the original article: $P(x)$ is an N-dimensional distribution of a random variable $x = x_1, x_2, ...x_n$, $z$ is a latent variable, i.e. a variable that is not directly observed but inferred from a model, and $P(x_j|z_i)$ are 1-dimensional distributions. This is a comparable model to the one used in NMF, where $P(x_j|z_i)$ represent the basis functions present in the $W$ matrix and the activations in $H$. However the weights also present in $H$ are in this model represented in $P(z)$ as the priors of the latent variable. To obtain an approximate distribution of $P(x)$ a weighted sum of the marginal products is done, shown in equation (2.11). Then, this model is expanded (equation 2.12) so the latent variables become shift-invariant across multiple dimensions, in order to allow for better analysis of more complex distributions.

$$P(x,y) = \sum_z P(z)\int\int P(\tau_x,\tau_y|z)P(x-\tau_x)P(y-\tau_y|z)d\tau_x d\tau_y \tag{2.12}$$

Instead of multiplying the marginal distribution a convolution of kernel distributions (the basis functions), $P(x,\tau|z_i)$, is performed with an impulse distribution (activation matrix), $P(y-\tau|z)$.

Now that the NMF and similar methods have been presented, it is possible to understand how they can be useful in a musical context. A representation of a musical audio signal is the $V$ matrix that is going to be factorised, the $W$ matrix will represent components of that signal, such as notes or instruments, and the $H$ matrix will show the respective activations of each component. In

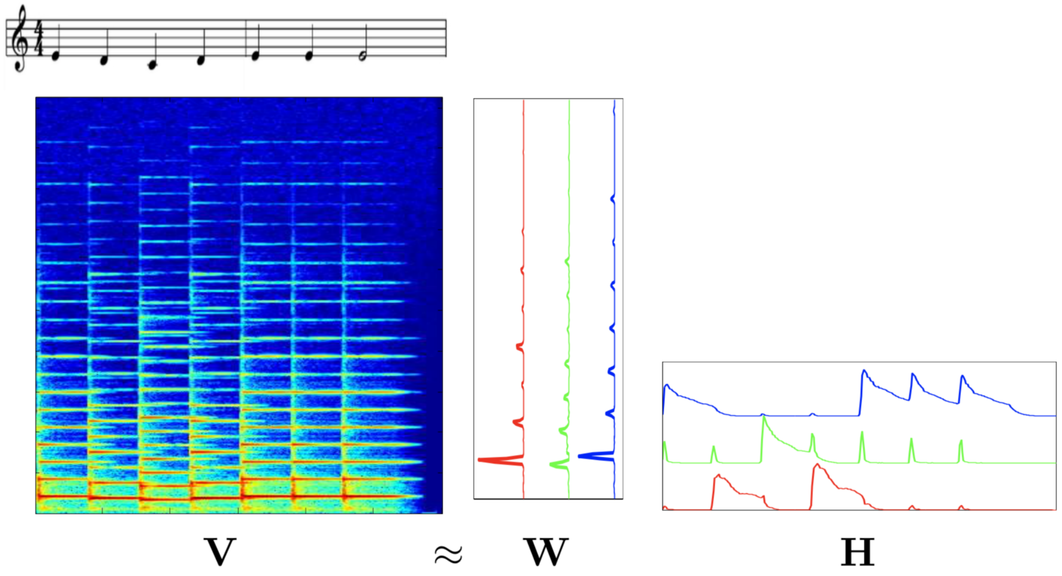


Figure 2.5: $V$ represents the recording spectrogram, $W$ the basis functions and $H$ the activation matrices obtained using a NMF. The music sheet of the recording is represented on the top. Taken from https://ccrma.stanford.edu/~njb/teaching/sstutorial/part2.pdf.

Figure 2.5 NMF is applied to the spectrogram of a piano recording of *Mary Had a Little Lamb*, an American nursery rhyme, with $r = 3$, meaning that the signal will be decomposed into three basis functions, which is very convenient in this case as the song only has three notes. Therefore it is possible to observe that each column of $W$ represents a note, each row of $H$ where that note is being played and how much weight it has, allowing to separate frequency from time. In a more musical context, this means separating the harmonic content of a musical piece from its place in time. This is an example of a NMF implementation in a music context using a standard spectrogram to represent the musical signal, that works well to illustrate the capabilities of the NMF because the input signal is very simple, and the rank of the factorisation is the best possible. For a more complex signal, e.g with multiple instruments playing at the same time, a better representation and a more complex NMF would be required as well as a higher rank of factorisation. For example, if the shift-invariant PLCA implementation was used in the example presented above (with a logarithmically spaced spectrogram), only one basis function instead of three would be needed to obtain an approximation of the input signal.

Some other musical applications of the NMF are source separation [37] [38], adaptive signal effects [39] and similarity suggestions [4]. In [38] a way is presented to separate musical instruments, in this case a piano and a trumpet from a recording of those two instruments together. In [4] it is explained how Spotify, one of the most popular streaming platforms currently, uses a matrix factorisation technique called Logistic Matrix Factorisation to create lists of related artists, and thus demonstrating that matrix factorisation can find uses in radically different problems related, in the broadest sense, to musical data.

## 2.4 Recombination of Audio Signals

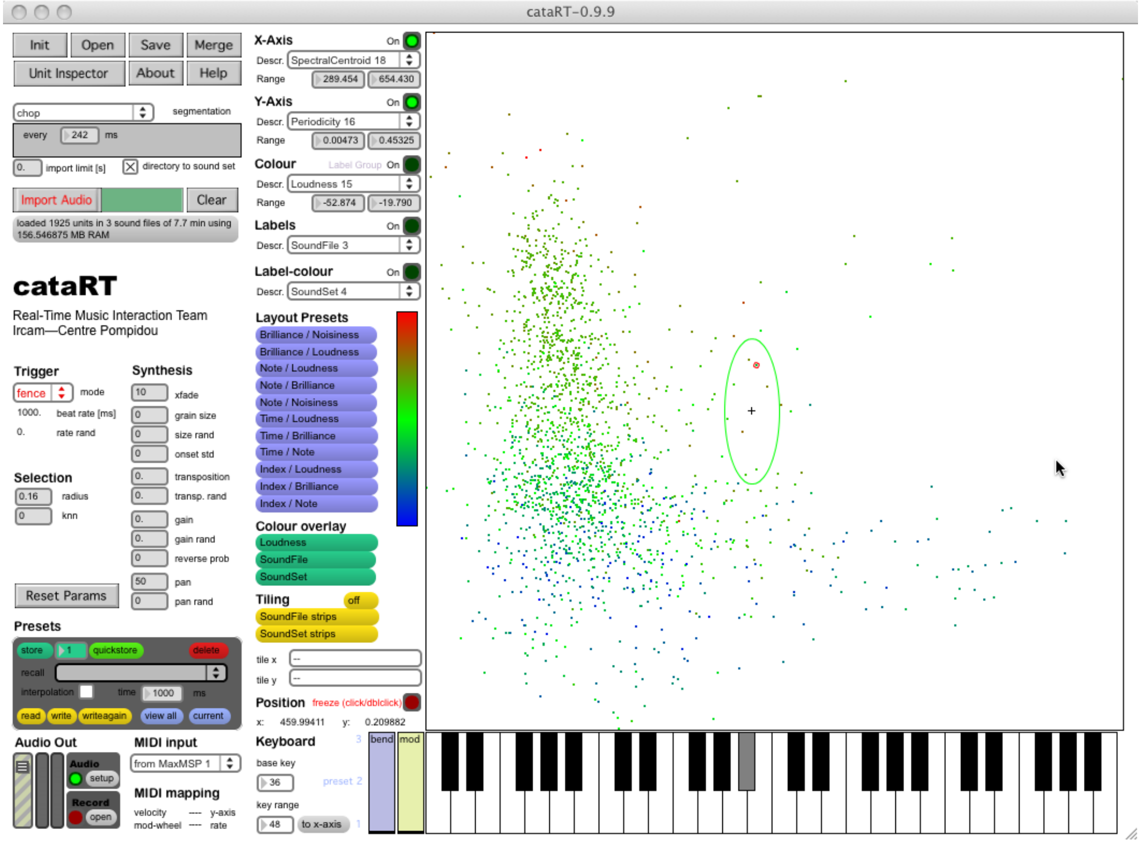
As mentioned before, a field of study of MIR that is being further explored in the last years is creative MIR, which tries to discover new creative ways of recombining the information gathered about the features of musical signals. Some of the work already done in this field addresses music mashups, which is basically a blend between two or more different songs in order to create entertaining result and surprise the listener. Most mashups consist of laying down a vocal track of a certain song on top of the instrumental version of another. Mashups have become more popular since software tools to analyse and combine audio became available to commercial usage. Features extracted using MIR techniques such as key and BPM are essential to this process, simplifying the workflow necessary to create mashups. Another approach is audio mosaicing, that has the goal of transfering features from a source signal, such a timbre, to a target signal.

In [19] a tool to automatically create mashups is presented. The process consists in first analysing the input song, using beat tracking, semitone spectrograms/chromagrams and phrase segmentation, then estimating the mashbility criterion using harmonic similarity of beat-synchronous chromagrams for all possible key shifts, to finally combine the input audio phrase with a phrase of another song with high mashbility. This final section requires beat-matching, pitch-shifting and loudness adjustments. The user can also manually choose the subset of songs that the algorithm will try to find mashbility with, having options such a *artist/album mashups* and *style mashups*.

Other example is concatenative synthesis. As Schwarz defines in [40], "*concatenative synthesis methods use a large database of source sounds, segmented into units, and a unit selection algorithm that finds the sequence of units that match best the sound or phrase to be synthesised, called the target*". Schwarz also created a tool for real-time concatenative synthesis called *CataRT*, [41]. *CataRT* analyses the input audio or pre-recorded sample by segmenting them using transient analysis or by silence thresholds to then create a descriptor containing information such as: the fundamental frequency, aperiodicity, loudness, spectral centroid, sharpness, spectral flatness, high frequency energy, mid frequency energy, high frequency content, first order autocorrelation coefficient and energy. All the descriptors are saved to a big $(N,D)$ matrix. To select the unit that is going to be played, the distance between the position selected in the 2D descriptor space, $x$, to all units in the $(N,D)$ matrix is calculated. The user interface provided by this system is shown in Figure 2.6.

This is formulated in the equation (2.13) where $\mu$ is the matrix $(N,D)$ and $\sigma$ the standard deviation of each descriptor.

$$d = \frac{(x-\mu)^2}{\sigma} \tag{2.13}$$



loaded 1925 units in 3 sound files of 7.7 min using 156.546875 MB RAM

Figure 2.6: *CataRT* interface. Taken from [41].

*EarGram* by Bernardes [42] follows the same conceptual approach used in *caraRT* but offers new tools of visualisation of the descriptor space (corpus), such as features vector to represent the units and various options for their method of recombination with the purpose of having different creative possibilities. These methods are: *SpaceMap/SoundscapeMode*, *InfiniteMode* and *Meter Mode*. In the *SpaceMap/SoundscapeMap* mode the aim is to create sonic textures. The *InfiniteMode* restricts the musical gesture in order to not repeat itself during an indefinite amount of time. In *MeterMode* targets are defined according to a meter.

A very recent work exploring another way of recombining audio was presented by Tralie in [43], in which he tries to recreate how the rock band Alien Ant Farm would cover Michael Jackson’s “Bad”, based of their cover of another song by Michael Jackson, “Smooth Criminal”. This can be seen as style transfer. Here the cover (A’) and the original song (A) are synchronised using beat detection and time-stretching. Next an NMF is applied to the CQT of A, A’ and to the song to be covered, B. The idea is for each basis function to represent an instrument, so there is a correspondence between the basis of A, $(A_1,A_2,A_3)$ and A’,$(A'_1,A'_2,A'_3)$. For example, an acoustic guitar in A, $A_1$, can be translated to a electric guitar in A’, $A'_1$. This can be seen as a translation dictionary between instruments. The last step is to use an invertible CQT, in this the case the one already described in this report [28], to use the basis of A with the activations of B,$(H_1,H_2,H_3)$, and then to replace the basis of A with the ones of A’, using the translation dictionary, to finally achieve the wanted cover recreation, B'.

In [5], Driedger explores the idea of creating audio mosaics, that is, reshaping a target signal with characteristics of a source signal, i.e. its timbral content, similar to what is intended in this project. One of the main results obtained in the paper is having a swarm of bees buzzing the melody of "Let it Be", by The Beatles. First, multiple pitch-shifted versions of the bees recording were concatenated in order to have a bigger range of pitches in the source. NMF is applied to the Let it Be recording, in order to learn its activation matrix, $H$, and then the spectrogram of the source is multiplied by $H$. To this preliminary result restrictions are then applied in order to eliminate effects such a stuttering and loss of timbral content. This is done by:

- Restricting the number of possible adjacent horizontal activations to reduce the stuttering due to using the same spectral sample over and over again.
- Restricting the number of possible simultaneous vertical activations to reduce phase cancellation in order to maintain the energy distribution from the source.
- Forcing a diagonal shape also in adjacent activations, to preserve temporal characteristics of the source and not lose the buzzing sound of the bees.

Another example of audio mosaicing is presented in [6], however with a very different goal from the examples presented until now. Here, the idea is to automatically convert a pop music into chiptunes, also known as 8-bit music, widely used in the 1980s/1990s mostly in the context of videogames. This is done by first, separating the vocals from the background instrumental of the source music, by using an algorithm of robust principle component analysis (RPCA), in order to extract its main melody. The vocals (foreground) analysis consists in pitch-tracking using a algorithm called *pYIN* while the instrumental (background) analysis is done by applying NMF to the separated instrumental part, $V$, using chiptone notes as the matrix $W$ in order to extract the correspondent activations matrix $H$. The number of simultaneous activations in $H$ is also controlled as in [5], in order to make the signal less busy and noisy. Each of these processes is then followed by smoothing using a median filter. There is also another constraint applied to the NMF that restricts the maximum/minimum pitch of the vocals based on the instrumental maximum/minimum pitch. The final step is to synthesize the desired result, process that is made in the time-domain using techniques such as overlap-and-add.

In [44] Burred presents the software *Factorsynth* that explores sound processing using matrix factorisation. The most relevant feature to describe in the context of this work is the cross-synthesis mode where two input audio files are selected. Figure 2.7 shows the user interface presented when this mode is selected and will allow to simplify explanation of the software.

First the user introduces the audio files and selects the number of components for each one of them to be calculated during factorisation. After the computation is done, the displays areas are filled with the components (top right) and the activations (bottom left). Then the matrix that allows to combine components and activations is divided. In blocks on the diagonal, the blue circles represent the components/activations connections within each of the input sounds. The

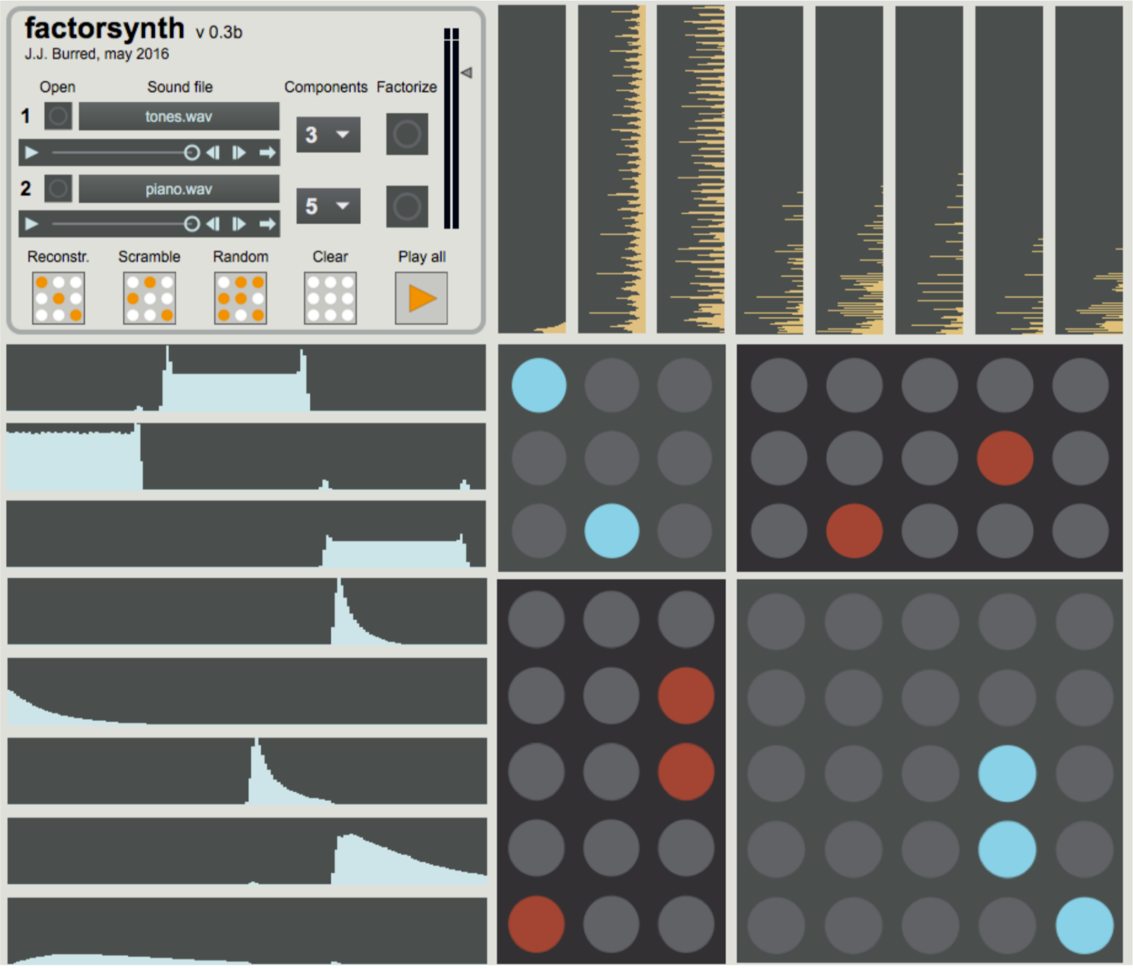


Figure 2.7: *Factorsynth* interface in cross-synthesis mode. Figure extracted from [44].

other blocks control connections (red circles) between the components of one sound with the activations from the other, therefore obtaining cross-synthesis.

One transverse problem to all of these kinds of recombination of musical audio is how to evaluate the results obtained in a systematic way. In [19] Davies et al. used listening experiments to understand the enjoyability of the mashups given by their automatic system. Another evaluation method also used in this same work was an objective analysis of the database and of the segmentation performance. A listening test is also described in [6] and focused on three main aspects, those being: pitch accuracy, 8-bit resemblance, overall performance. Each participant listened to four songs that were presented without name, and gave a classification from one (very poor) to five (excellent) to each aspect.

Therefore it can be seen that listening methods are commonly used to evaluate if the results offer a good listening experience. One other aspect important to evaluate is also the degradation introduced in audio quality while processing the signals, due to pitch-shifting/time-stretching which is going to be extensively used in this project.

## 2.5 Conclusions

As demonstrated, a lot of work in the area of MIR, creative MIR and use of machine learning in musical analysis has been done in the last years. This project intends to build upon some of the papers presented in this section, and at the same time introduce new ways of recombining audio signals with the purpose of creating a engaging musical creation process, driven by the idea of recombination of musical signals.

# Chapter 3

# Approach

In this chapter a detailed explanation of the different stages that this project went through during its development phase is presented. When developing the system, the focus was on creating a flexible framework that users, such as musicians and sound designers, could interact with in order to create interesting results easily and quickly.

The chapter is subdivided in 5 sections. The first gives an overview of the problem definition, the second one details the pre-processing done on the audio files, the third describes the algorithm of matrix factorisation, the fourth explains the post-processing applied to the results obtained from the matrix factorisation and finally the last details the functionality of the user interface.

## 3.1 Problem Definition

The problem consists on the creation of a framework that is able to reshape a certain piece of music to fit a certain music target. As stated before, the main goal of this research is to devise new methods of playing one song using the timbral content of another in such a way that both the target and the source material are recognisable by the listener and can be mixed together to create musically interesting results.

Other questions that this work will address are: further exploration of the creative MIR subfield, user-interaction, by allowing users to control certain high-level parameters regarding the algorithm, evaluation of the results obtained through the use of listening tests, as well as the use of machine learning techniques to analyse musical signals.

To address the problem a music recombination software using machine learning created in MATLAB program is proposed. In Figure 3.1 a flowchart of the main steps concerning the proposed solution is presented.

The first step is to create a dataset of audio files of songs from different genres to test the system. After that, and entering more in the actual solution, an adequate representation for the signals of the dataset needs to be chosen. Of all the ones presented in Chapter 2, the CQT and the Chromagram are going to be used. Although the CQT is a more complete and adequate representation compared to the others, it was decided to also include the Chromagram in order to

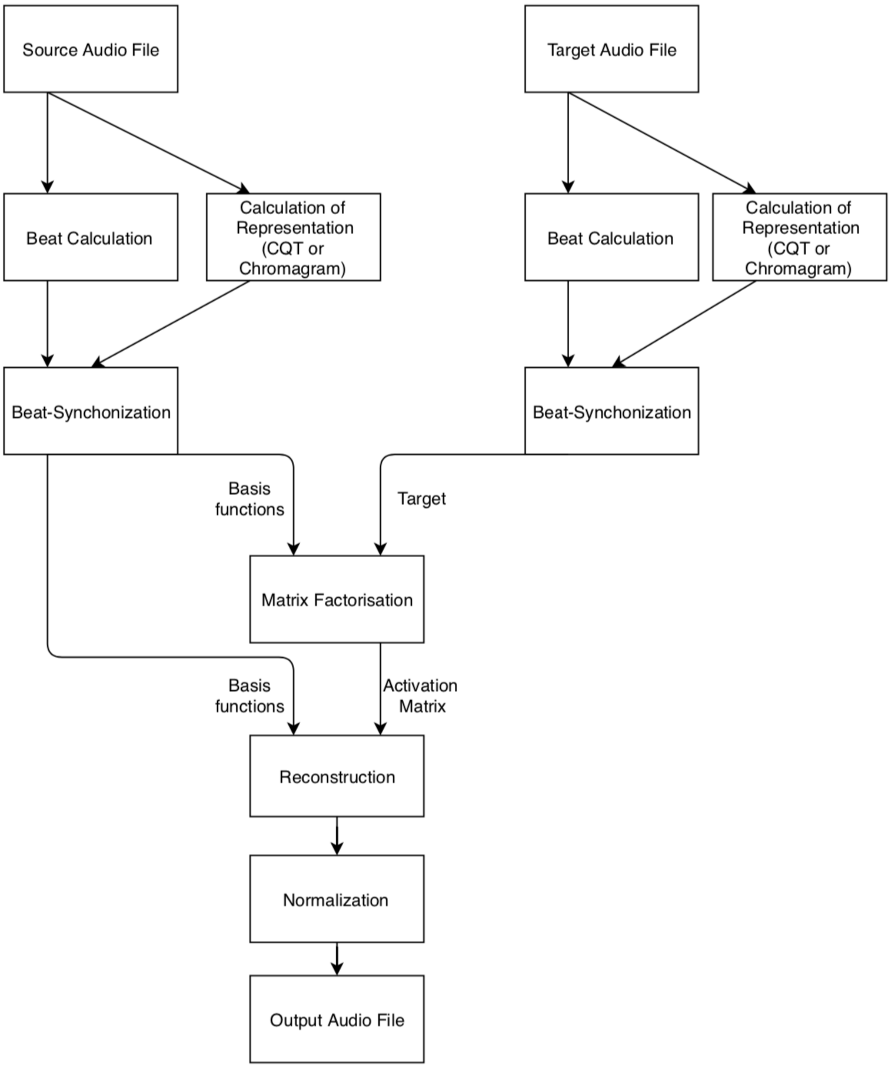


Figure 3.1: Flowchart of the proposed solution.

allow for more flexibility but also to evaluate the impact of using the CQT compared to using the Chromagram in the matrix factorisation context.

At the same time some operations related with beat calculation for both the target and the source with the purpose of synchronising the beats with their representation, in order to perform cross-synthesis in musical time, need to be performed.

The method to retrieve information from the chosen representation is the NMF, more particularly the PLCA implementation proposed in [36] that allows for an impulse distribution in the activations matrix and for shift-invariant basis functions. Using the source basis functions and the target the PLCA will calculate an activation matrix for each source basis functions that contain information about temporal activation (at which beat of the target), pitch-shift (by how much semitones is the basis going to be transposed) and intensity (how strong is the activation).

The recombination section will perform pitch-shifting using the RubberBand[12] library or the CQT directly, depending on the type of representation chosen. This is done to match the basis-functions to the target in terms of harmonic content. Another point of interaction is introduced here as the user can choose the maximum number semitones for pitch-shifting. In order for the reconstructed audio be comparable in volume to the target it tries to match it needs to be normalised in amplitude.

## 3.2 Pre-Processsing

Before feeding the input signals into the matrix factorisation algorithm, some processing needs to happen to represent the signals in a more adequate way. This will involve calculating the CQT transform and Chromagram, beat estimation and beat synchronisation in order to synchronise the source and the target audio.

### 3.2.1 Input Representation

The first step for the algorithm was to decide which representations were the most adequate to use for the input audio signals. Of the alternatives already presented in chapter 2, two were implemented, the Chromagram and the Constant-Q Transform.

#### 3.2.1.1 Chromagram

To run the Chromagram on the audio files, the implementation used is the one given by the Vamp Plugins[13], more specifically the *NNLS* function. This function is also used to calculate the tuning of both the source and target in order to correct possible small differences in tuning, during the reconstruction phase, between the two audio signals.

#### 3.2.1.2 Constant-Q Transform

To perform the CQT on the signals two different toolboxes were considered, those being the ones presented in [27] and [45]. Both offer an invertible CQT implementation and provide functions in order to easily perform pitch-shifting, operation that is going to be required during the reconstruction phase. Although the implementation in [45] allowed for a faster computation of the CQT, the implementation in [27] provided a full matrix representation (as opposed to a sparse matrix structure which would be non-trivial to process by matrix factorisation). The parameters chosen for the CQT are:

- Number of bins/octave: 48, resulting in 4 bins/semitone.
- Maximum Frequency: Sampling Frequency/2, usually 22050$Hz$.

[12] https://breakfastquay.com/rubberband/

[13] https://www.vamp-plugins.org

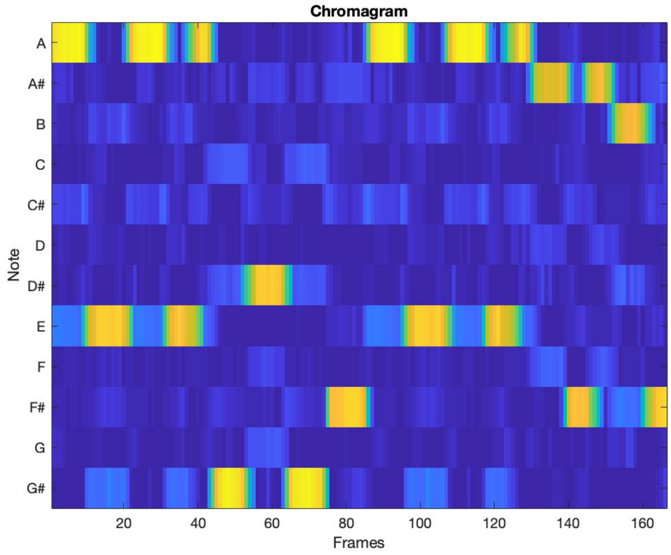


Figure 3.2: Chromagram of a recording of an electric piano playing a melody.

- Minimum Frequency: Maximum Frequency/$2^9$, usually $43Hz$.
- Compression Factor: 0.7, used to make the visual representation clearer.

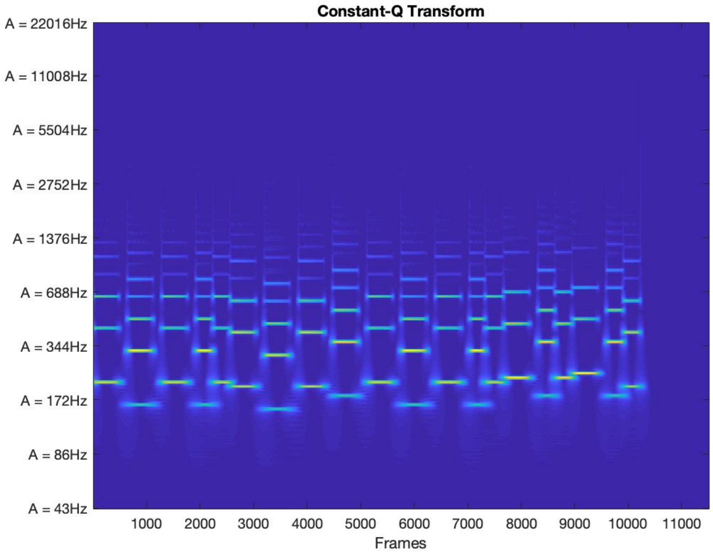


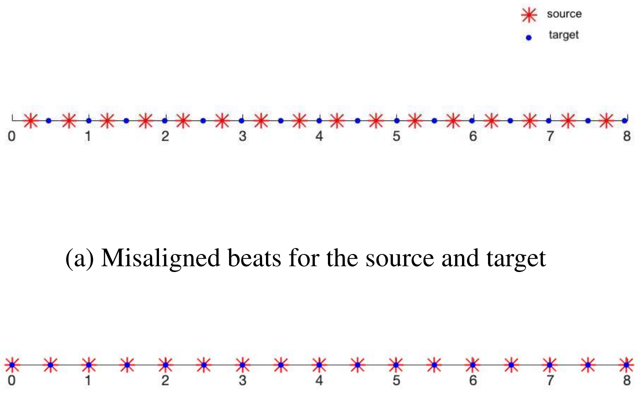
Figure 3.3: CQT spectrogram of a recording of an electric piano playing a melody.

### 3.2.2 Beat Processing

To conduct the cross-synthesis in musical time and reduce the amount of data to analyse, the representation of the source and target is beat-synchronised. Thus, a way to calculate where these beats are in the music and how to synchronise them with the input representation, process called beat-synchronisation, is also needed.

#### 3.2.2.1 Beat Calculation

The beats were extracted using the Bar and Beat Tracking plugin from Sonic Visualiser[14]. This algorithm (like many beat tracking systems) runs into a problem of sometimes detecting the off-beat and thus leading to a misalignment beats between the source and the target, as shown in Figure 3.4. In order to fix this problem, a beat correction algorithm was implemented which identified the mean inter-beat-interval and then offsetting either the beats of the source or the target by half of this duration.



(a) Misaligned beats for the source and target

(b) The alignment after applying the beat correction

Figure 3.4: Example of beat correction.

#### 3.2.2.2 Predetermined Beats

For short musical examples of constant tempo (e.g. musical loops), it may not be necessary to use a beat tracking algorithm and instead use predetermined beats calculated accordingly to the information introduced by the user. That is, if the user knows the BPM of each source and of the target, he/she can simply introduce them in the user interface, and the beats locations will be calculated accordingly. If the introduced BPM values are correct, this approach involves less calculations and can offer very accurate results. Furthermore, it can allow the user to specify their preferred tempo (which may be double or half of which is estimated by the beat tracking algorithm).

[14] https://vamp-plugins.org/plugin-doc/qm-vamp-plugins.html#qm-barbeattracker

### 3.2.3 Beat-Synchronisation

After obtaining the input representation and the location of the beats, the next step is to synchronise the representation with its corresponding beats to calculate the basis functions to be used in the NMF algorithm. This is done by using a function called *BeatSynchonize* that is presented in [46]. The *BeatSynchonize* function calculates the median value per bin for all frames within each beat to generate beat-synchronous vectors. To accurately perform this task the value of the time resolution needs to be adequate. For the Chromagram case, the default time resolution of 2048/44100 was used. In the CQT, it is possible to apply the same logic to obtain a beat synchronous version, however the temporal resolution is much higher (i.e. many more frames per beat), which can result in significant blurring between beat frames. As an alternative it is possible to isolate each beat of the audio signal and then calculate a new CQT per beat, in a strictly, beat by beat way. An example of the differences between this new approach and the previous one is shown in Figure 3.5. Although a more complex process, this also allows the possibility of combining different basis functions from different audio files into the framework, as the isolation of the beats allows them to be independently transformed back to the audio domain.

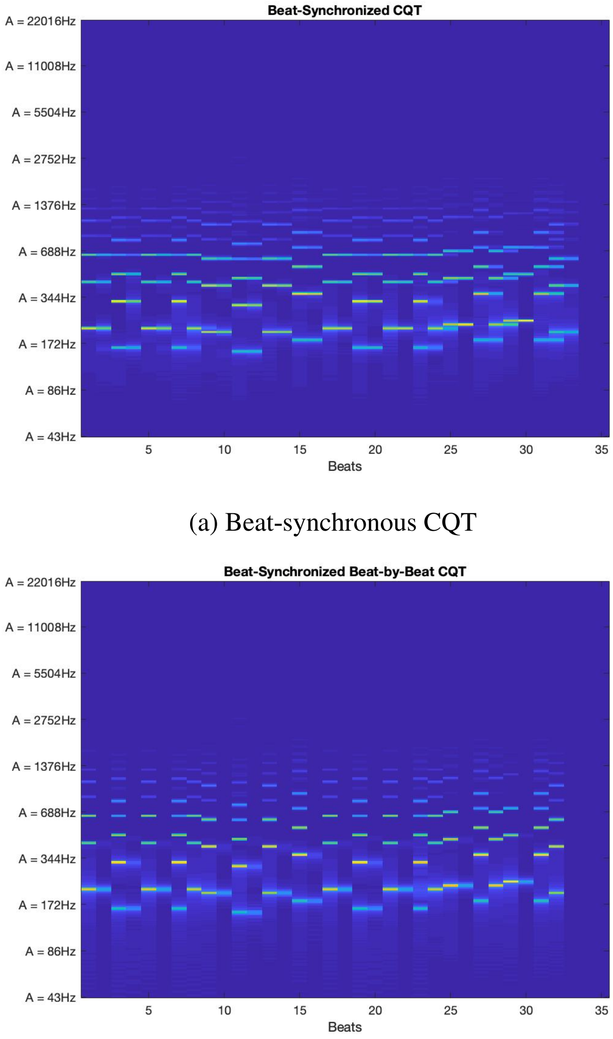
## 3.3 Probabilistic Latent Component Analysis

Regarding the use of the NMF, the implementation being used is the one presented in [38], PLCA, which main function is described in section 3.1. For a certain target song, we use its beat-synchronised CQT spectrogram (or the Chromagram) as the target, $x$, use the basis functions from the source as $w$ and have the PLCA algorithm calculate the activation matrices, $h$, that best approximate the target. In order to allow for a reconstruction of the signal using only the original basis functions, the option that that controls the update of the $W$ matrix is turned off.

$$[w,h,z] = cplca(x,K,T,z,w,h,iter,sz,sw,sh,lz,lw,lh,pl,ps) \tag{3.1}$$

Inputs:

$x$ = target
$K$ = number of basis functions
$T$ = size of basis functions
$z$ = initial value of $z$
$w$ = initial value of $w$
$h$ = initial value of $h$
$iter$ = number of iterations
$lz$ = update flag of $z$
$lw$ = update flag of $w$ (zero, in order to not update the basis functions)
$lh$ = update flag of $h$
$ps$ = number of maximum semitones pitch-shifts



(a) Beat-synchronous CQT

(b) Beat-by-Beat beat-synchronous CQT

Figure 3.5: It is possible to see that the notes of one beat don't bleed as much to the next beat, making the melodic representation of the musical piece more accurate when the Beat-by-Beat beat-synchronous CQT is used.

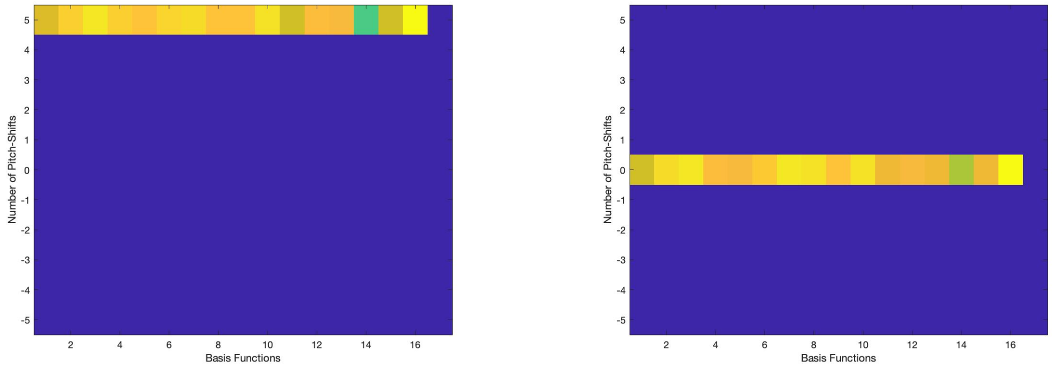
Outputs:

$w$ = input $w$ (as their initial value is not altered, the basis functions stay the same)
$h$ = cell that for each basis function contains an activation matrix
$z$ = basis functions priors, that indicate the relevance to the reconstruction of a certain basis function

### 3.3.1 Shift-Invariance

One change made to the existing implementation was to add the capability of pitch-shifting the basis-functions, made possible because of their shift-invariant property, meaning that the activation matrix can indicate both the temporal location of the basis function and a vertical shift which simulates a pitch-shifting operation. This was done by adding a parameter to the function, *ps*, that changes the dimensions of $H$ according to the maximum number of pitch-shifts, instead of the dimensions being equal to the ones of the basis functions. Thus, each $H$ activation matrix has $2 * ps + 1$ rows and as many columns as the number of beats of the target.

This however, introduced the problem of the activation matrix not being centred around 0 but instead being on the top. To solve this problem the basis functions of the source are circular shifted down by the number of maximum pitch-shifts, allowing for the activation matrix to be centred around 0, so that both pitch-shifts up and down are performed correctly. In Figure 3.6, an example of the effect the circular shifting has on the activation matrix is given. The target and the source files are the same, so the expected result is to obtain an almost perfect reconstruction of the target file, that is, the basis functions would not be altered (i.e. would not suffer pitch-shifting), but it can be seen that without circular shifting the basis functions (Figure 3.6a), the activation matrix is centred on the top (maximum value of pitch-shift) which forced that the values would be later corrected to represent 0 pitch-shifts, but, simply circular-shifting down the basis functions the amount of maximum pitch-shifts allows the activation matrix to be calculated correctly directly from the PLCA algorithm.



(a) Activation matrix before basis functions shifting when target is equal to the source

(b) Activation matrix after basis functions shifting when target is equal to the source

Figure 3.6: Effect of circular-shifting the basis functions on the activation matrix. To better illustrate this effect the activation matrices from each basis functions were concatenated and the number of simultaneous activations was limited to one

.

### 3.3.2 Input Parameters

While computing either the CQT or the Chromagram it is needed to consider that the input of the PLCA is restricted to be non-negative. The Chromagram already offers a non-negative representation but regarding the CQT, which is complex, its magnitude is used as the input. If the chosen representation is the CQT, the phase is retained for re-synthesis. For the Chromagram, the pitch-shifting is done on the corresponding beat segments and thus there is no need for phase information.

Regarding the remaining input parameters, the number of basis functions used is usually the maximum possible, this being the number of beats of the source (as well as if the beats are subdivided into smaller metrical units), and the number of iterations depends if it is either the CQT or the Chromagram being used to represent the signals, as these converge after a different number of iterations (more iterations for the CQT, typically around 200, and fewer for the Chromagram, around 20).

Although the remaining input parameters can be almost totally controlled by the user, there are some restrictions and explanations that are needed to understand the system. Regarding the amount of pitch-shifts, the maximum value for the Chromagram is 5 semitones as anything above this will cause an error on the algorithm due to the activation matrix being bigger that the input representation (considering no pitch-shift and pitch-shifting up and down, the activation matrices would have 13 rows while the Chromagram matrix only has 12) so usually a value between 3 and 5 semitones is used. The CQT however requires a different approach regarding the pitch-shifting. Early experiments showed that using a small value like 20 bins (equivalent to 5 semitone pitch-shifts) did not produce good results, so a different approach was taken. To the PLCA algorithm, the number of pitch-shifts bins given is quite high, e.g. 48/96/144 (1, 2 and 3 octaves respectively) which will create a big activation matrix. Later the pitch-shift values given in the activation matrix are interpreted. One of the processes involved in this interpretation is the filtering, this is, all the values above a certain pitch-shift are discarded. Even though that the CQT allows for pitch-shifts bigger that 5 semitones, the pitch shifting of audio signals is a lossy process that harms audio quality, so the maximum pitch-shift is usually limited to 20 bins (pitch-shifts of 5 semitones).

## 3.4 Post Processing

After getting the activation matrix from the PLCA algorithm, the process of reconstructing the signal back to the time domain can start. However, in order to obtain the best possible results some post processing on the obtained data is required.

### 3.4.1 Activation Filtering

Before starting the reconstruction phase, the number of simultaneous vertical activations, that is, the number of activations in a certain beat in the $H$ matrix is filtered according to the value that the user introduced. This is done to prevent musically confusing results that are obtained when multiple audio snippets from the source are layered. The idea of doing this was taken from related works such as [5], that also uses it with a similar purpose. In Figure 3.8 an example of this filtering is presented. In the example, to better illustrate the effect that this filtering has, the activations matrices from the basis functions were concatenated. In this case the maximum number of simultaneous activations was defined as 3, so, in every beat, the three highest values were chosen and only those remained in the final version of the activation matrices. It can easily

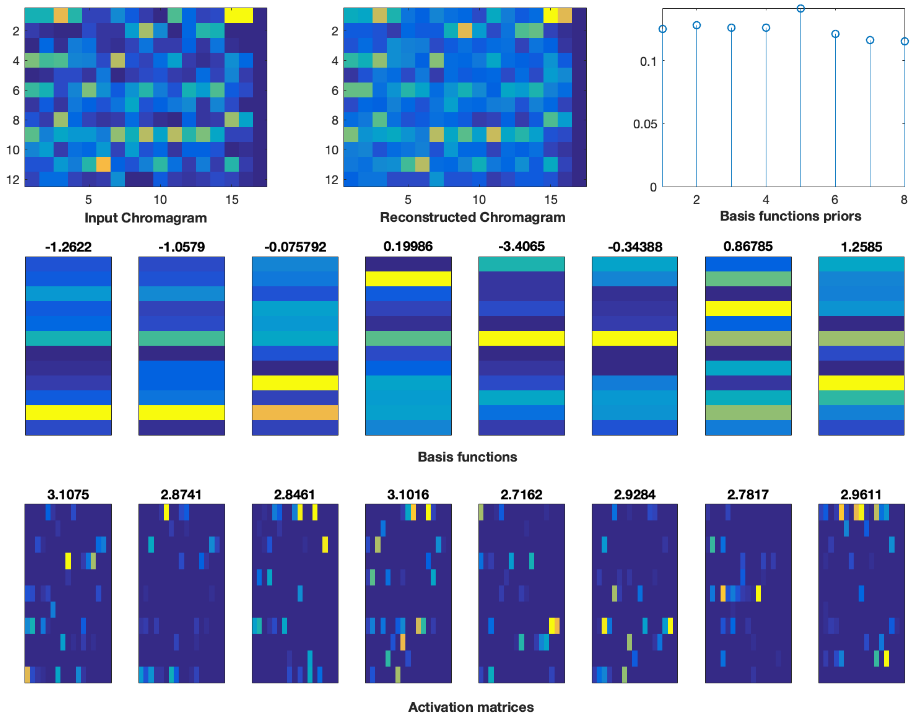


Figure 3.7: Example of the PLCA algorithm output. It is possible to observe the target, the reconstructed target and the basis functions priors in the top row, the basis functions from the source in the middle row and the activation matrix correspondent to each basis function in the last row.

be seen that after the filtering a lot of activations with very little importance were removed, which will make the result obtained sound more logical and musical, but at the expense of a less precise reconstruction of the target from the basis functions of the source.

### 3.4.2 Signal Reconstruction

After this is done the reconstructed signal is created. For each basis function, the associated activation matrix is analysed, that is, a loop over the target beats is executed. Within each target beat of each basis function, if there is an activation, the basis function is pitch-shifted, multiplied by its prior and by the activation weight and is placed accordingly in the output vector.

#### 3.4.2.1 Pitch-Shifting

However, extracting the pitch-shift value requires some interpretation in order to obtain the correct value.

For the Chromagram the value of the pitch-shift is simply extracted as it already corresponds to the actual pitch-shift that needs to happen. This value is passed to the *Rubberband* function,

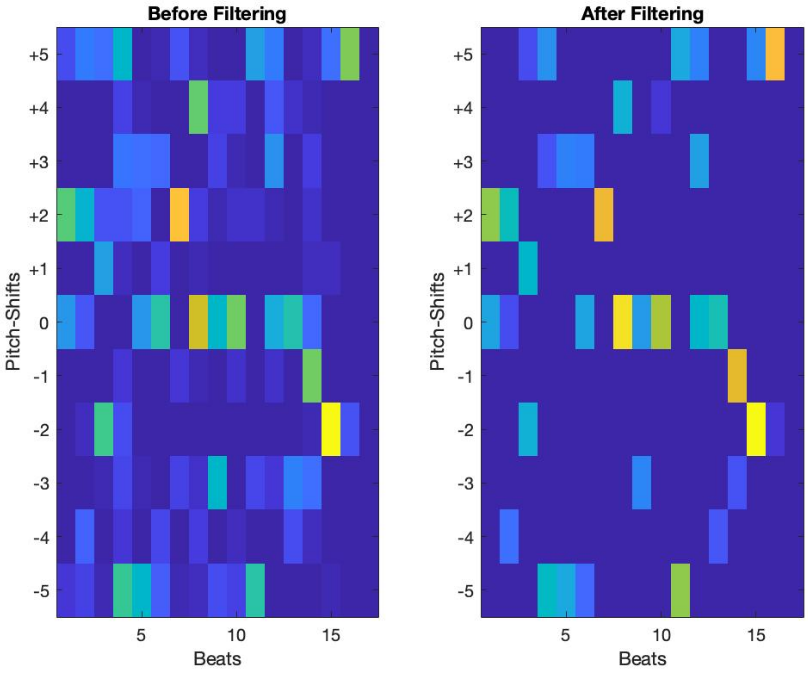
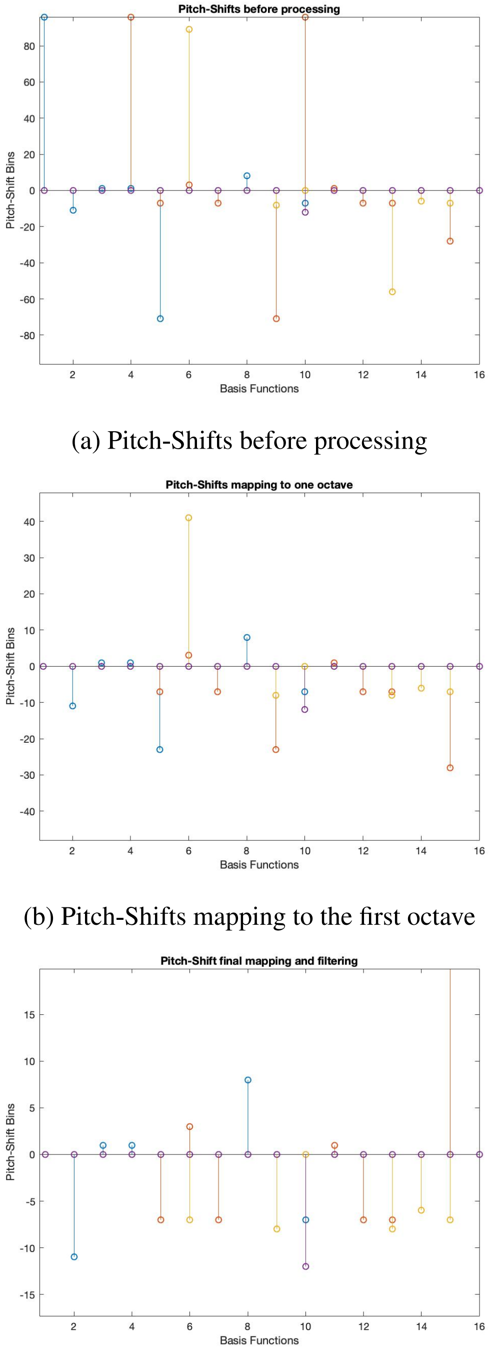


Figure 3.8: Activation matrix before (left) and after (right) the filtering of the number simultaneous activations.

that performs audio pitch-shifting, and time-stretching if the beats from the source and target have different lengths.

For the CQT, in some cases, there needs to be some mapping and filtering regarding the pitch-shifts in order to obtain good results. As mentioned before, the number of pitch-shifts given to the PLCA is quite high (e.g. 96/144). However pitch-shifts of such magnitude do not make sense in a musical context so they need to adapted. First the values outside of the first octave (>48 or <-48) are mapped to the first octave. e.g., if the value of the pitch-shift is originally 88, it is going to be mapped into 40, one octave below (i.e. 48 bins below). Next, if the value of the pitch-shift is higher in absolute terms than the maximum number of pitch-shits allowed, it is again mapped to one octave above or below. Following on the example presented above, let us assume that the maximum pitch-shift value is 20 CQT bins. The pitch-shift, originally 88 and now 40, is still higher than 20 so it is going to be brought down another octave, becoming -8 (40-48). The logic behind this process is that, if the harmonic match occurred at that specific pitch-shift, the octave value is not as important as the harmonic relation between the target beat and the shifted basis function. Finally, the pitch-shifts are filtered, this time to make sure that they don't surpass to the number of maximum pitch-shift introduced by the user. An example of this process is found in Figure 3.9. To actually perform pitch-shifting we use the CQT shift property instead of the *Rubberband* function, that in this case is only used for time-stretching.



(a) Pitch-Shifts before processing

(b) Pitch-Shifts mapping to the first octave

(c) Pitch-Shifts after final mapping and filtering

Figure 3.9: Example of the pitch-shifts mapping and filtering. Each graphic represents the pitch-shift values associated with each basis function.

#### 3.4.2.2 Tuning Correction

Another task that needs to be done to the source audio file and its respective basis functions during this phase is tuning correction.

If the representation used is the Chromagram the tuning needs to be corrected using the *Rubberband* library by calculating the tuning ratio between the target audio and the source audio and multiplying it by the pitch-shifts values. The tuning is calculated using the *NNLS* function as mentioned above.

However, if the representation used is the CQT, there are 4 bins/semitones, so tuning differences will be automatically corrected due to the pitch-shift value already containing the tuning correction required. This however can sometimes cause the reconstructed target not to have a constant tuning, due to all of the pitch-shift values not being necessarily apart by semitones. The musical result of this is the presence of some microtonality in the audio.

The differences between the two approaches can be seen in Figure 3.10 where the source is an audio file with tuning being A=440Hz and the target is the same audio file but tuned at A=420Hz. In the activation matrix corresponding to the CQT, the pitch-shift in every beat is the same, which is expected, being just above -1 semitone, more specifically, every beat is pitch-shifted down 3/4 of a semitone. To verify if this value performs an accurate correction, the difference in semitones between 440Hz and 420Hz was determined using the following formula that calculates the number of semitones relative to the note $C_0$ [15]:

$$S = 12 * log_2(f/16.352) \tag{3.2}$$

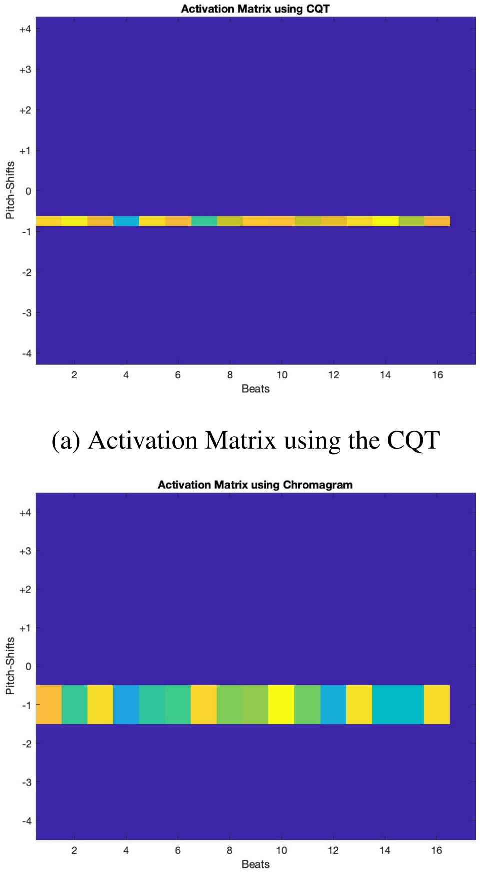
As the difference in semitones between 440Hz and 420Hz is 0.8 semitones, this correction is quite accurate (0.75 semitones). This level of resolution can not be achieved with the standard 12-bin Chromagram representation, so its possible to see that every beat is pitch-shifted down 1 semitone. In order to compensate for that lack of resolution, the tuning ratio between the source and the target is calculated and multiplied by every pitch-shift in order to obtain the best possible result.

### 3.4.3 Beat-by-Beat RMS Normalisation

After the signal is reconstructed it usually has an inconsistent volume compared to the target audio that it tries to match, so the mix of the 2 audio files sounds unbalanced. To correct this problem, a simple approach of peak normalisation was applied initially but some limitations came with it, such as big variations in volume, caused by very high peaks (percussive hits), that would lower the volume quite a lot in some sections of the reconstructed signal.

Thus, a more perceptual approach was followed, called RMS, Root Mean Square, normalisation. The RMS is defined as the root square of the mean square of a set of numbers, and can be used as an approximation of loudness, i.e., the subjective perception of sound pressure. After applying RMS normalisation to the signal and noticing better results this approach was further

[15] retrieved from https://www2.ling.su.se/staff/hartmut/bark.htm



(a) Activation Matrix using the CQT

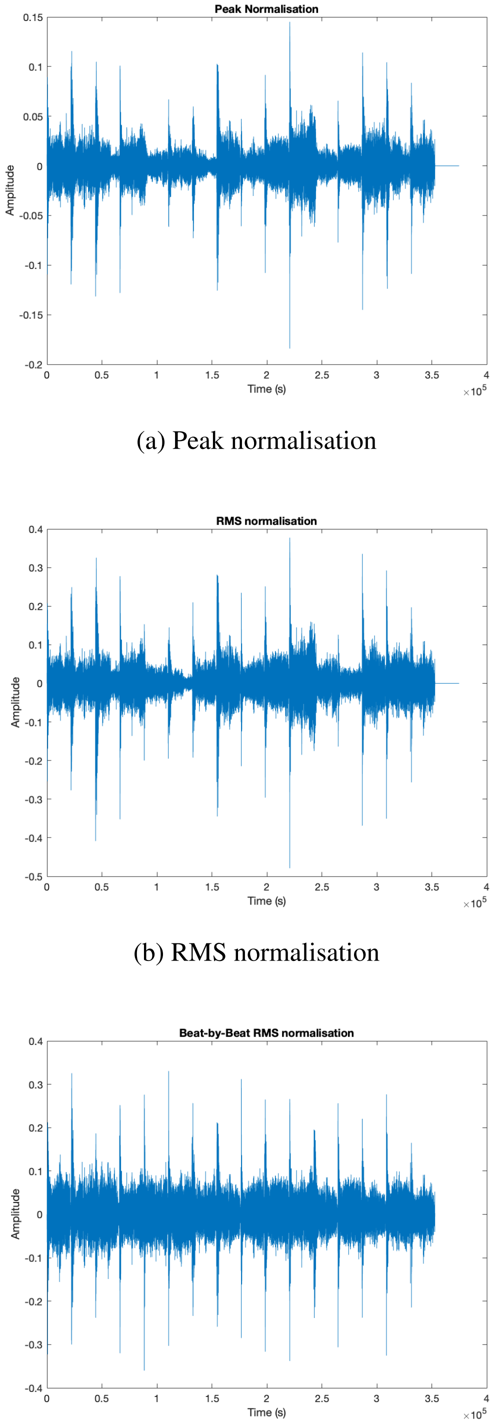
(b) Activation Matrix using the Chromagram

Figure 3.10: Activation Matrices (obtained by concatenating all the activation matrices of the basis functions) when the target audio (A=420Hz) is the same as the source audio (A=440Hz) but with a different tuning. a) Using the CQT, b) Using the Chromagram

developed in order to have RMS normalisation at a beat level. To this end, we extract the RMS of each of the beats of the target audio, and normalise the corresponding beat of the reconstructed audio to that RMS value, which allows for a much better and consistent mix as reconstructed audio follows the target volume a lot more accurately. This method needs to use a threshold to make sure that very quiet beats do not get wrongly normalised. Still, in some cases this normalisation creates values over 1, so if that happens it is performed a peak normalisation to 1 to guarantee that the signal does not suffer digital clipping. Also, to reduce some of very high peaks that appear in the audio sometimes, a median filter is also applied to smooth the signal. An example of the results obtained by each one of the approaches is presented in Figure 3.11.

## 3.5 GUI

To make the interaction with the system simpler for the user, a Graphical User Interface (GUI) was developed in MATLAB. This GUI, presented in Figure 3.12, gives easy access to some high



Peak Normalisation

RMS normalisation

Beat-by-Beat RMS normalisation

(a) Peak normalisation

(b) RMS normalisation

(c) Beat-by-Beat RMS normalisation

Figure 3.11: Example of the same reconstructed audio being normalised in three different approaches. A more consistent volume throughout the audio file is obtained when the Beat-by-Beat RMS Normalisation approach is used, allowing for the mix with the target audio to usually sound better.

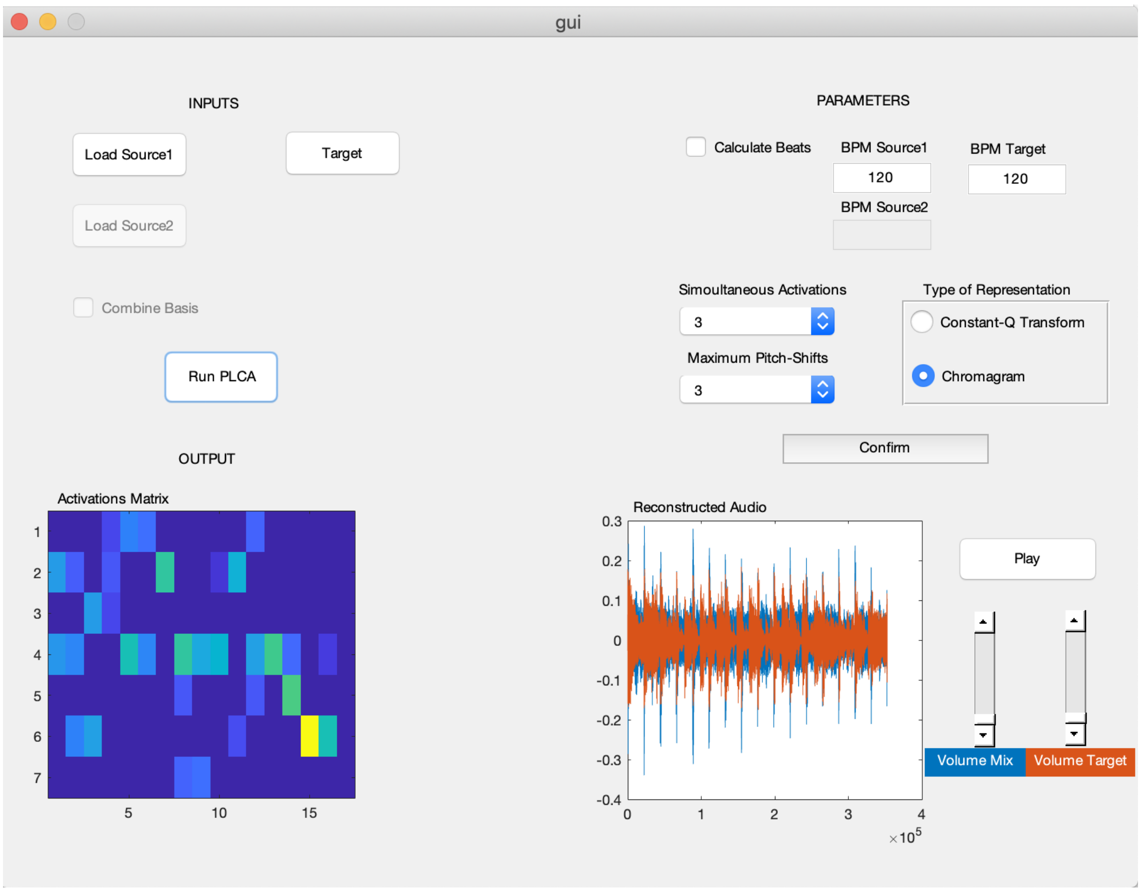
levels parameters that control the result obtained by the algorithm and allows the user to run the algorithm multiple times faster as well as to explore different approaches without having the need of manually writing commands in the MATLAB console.

The GUI is divided in three sections, those being: inputs, parameters, and output.

In the inputs section the user can choose the audio files to use as the source and as the target. The possibility of adding another source file and combining the basis functions of both the sources is unlocked if the representation chosen is the CQT, due to the Beat-by-Beat CQT allowing for this process.

The parameters section allows to control how the beats are calculated, that is, if the BPM is inserted by the user and therefore beats are calculated manually or if the beats are calculated by the Bar and Beat Tracking plugin from Sonic Visualiser. Also it is here that the type of representation for the audio files, either the CQT or Chromagram, is chosen. Regarding the post processing the user can indicate the maximum number of simultaneous activations and the maximum number of pitch-shifts. After introducing all the parameter values the confirm button saves them. Finally, the user can press the Run PLCA button in order to start running the algorithm.

In the output section, after the PLCA and post processing are complete, two representations are presented. On the left all the activation matrices from the basis functions are concatenated and presented and on the right the output stereo audio file is presented. The right channel (in red) contains the target audio and the left channel (in blue) the reconstructed target. A volume slider is available for each of one the channels so the user can listen to the files separately or combined.



gui

INPUTS

PARAMETERS

□ Calculate Beats

BPM Source1: 120

BPM Target: 120

BPM Source2:

□ Combine Basis

Simultaneous Activations: 3

Maximum Pitch-Shifts: 3

Type of Representation:
○ Constant-Q Transform
● Chromagram

OUTPUT

Figure 3.12: Graphical User Interface.

# Chapter 4

# Evaluation

This chapter describes the methodology used to test and evaluate the performance of this system when compared against a similar but simpler approach. Other cross-synthesis approaches described in Chapter 2 were not comparable against the one presented here, as their main goal is focused around trying to recreate the target audio as closely as possible by combining very small audio chunks (10-50ms long) from the source, following a concatenative synthesis approach. Instead, in this system the main goal is to manipulate the source at a beat level in order to create a good mix when the result is played along with the target. Also, using a simpler approach of basically the same algorithm can give insight about the advantages brought by the use of machine learning, i.e. the PLCA algorithm.

## 4.1 Experiment Design

A listening test was developed with the purpose of analysing three different aspects of obtained results.

The first one is to evaluate the quality of the mixtures, that is, if the reconstructed target sounds good to listeners when combined with the actual target. To do this, a more naive approach was developed. Most of the core algorithm remains the same as the one described in Chapter 3, with the principle change being in the approach that is used to try to match the target using the source. The source and target are represented by beat-synchronous chroma vectors and in order to calculate their similarity the Euclidean distance measure is used. In this approach the number of simultaneous activations is limited to one, both because no intuitive way of allowing for more activations was found but as well to study if the use of simultaneous activations improves the results. However, the possibility of the pitch-shifting is included in this approach. Thus, three different approaches, the PLCA using the Beat-by-Beat CQT, the PLCA using the Chromagram, and the Euclidean distance using the Chromagram are compared in a listening test. The parameters used for each one of approaches were:

- Number of simultaneous activations: 3 for the PLCA CQT and PLCA Chroma, 1 for the Euclidean Chroma

- Maximum pitch-shift: ± 3 semitones for all the approaches

The second one also consists in a listening test comparing the same three approaches, but this time only the reconstructed target is evaluated in terms of temporal and harmonic structure.

The third and final purpose is to evaluate one result discovered during the development of the system that wasn't initially considered. When the target is a simple monophonic melody and the source a single note from an instrument the activation matrix is essentially a filter of a more complex representation like the CQT, and creates a musical representation that offers melodic and temporal information about a certain piece of music. The reconstructed target audio file is then the target melody being played by the instrument of the source. So, the goal is to understand if a musician or someone who has the ability to read music can easily identify a melody using this representation. In this sense, we are using the activation matrix as a kind of music transcription for simple melodies, but with the difference that it is pitch invariant, and therefore only shows relative pitch changes rather than absolute.

### 4.1.1 Data Preparation

To conduct the listening test some aspects were considered when choosing the audio files to be used. These aspects were:

- Audio Quality: Only WAV files were used.

- BPM: All the audio files had a similar value of BPM (maximum difference of 10BPM between the slowest and the fastest) so the audio quality wouldn't be affected too much when all the files were normalised to same BPM value, in this case, 120BPM.

- Duration: All the audio files used were 8 seconds long after tempo normalisation to allow sufficient musical material to judge the quality of the results, but not so long as to introduce difficulties in consistently rating examples over time. In addition, all songs were aligned according to their downbeat.

- Rhythm Structure: None of the selected song snippets are syncopated which allows for better rhythmic alignment between the source and the target (i.e. so the focus can be on the harmonic match without rhythmic interference from syncopation).

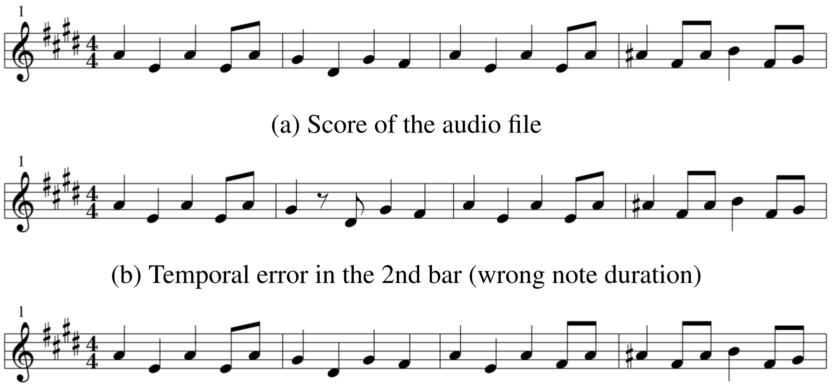
- Genres: Songs from a broad range of genres such as hip-hop, pop, disco, funk, indie were chosen in order to explore very different types of results.

Six songs were chosen with four examples of cross-synthesis being obtained[16], with each one using the three different approaches mentioned above. The first example used different parts of the same song (*DARE* by Gorillaz) as the target and the source. The second example used as the source an hip-hop song (*King Kunta* by Kendrick Lamar) and as the target a chorus from an indie pop song (*Follow the Leader* by Foxygen). The third used the same hip hop song as the target and

[16] With sound examples available at `https://paginas.fe.up.pt/~up201403798/MatFac/`

a bass heavy indie song as the source (*The Less I Know The Better* by Tame Impala). The last example combined two funk/disco tracks, *Que Tal America* by Two Sound Men as the source and *Shakey Ground* by The Temptations as the target.

For the final question, the DAW *Logic Pro X*[17] was used to create three audio files of melodies being played on a piano. One is the melody that was used as the target to create the presented representation (Figure 4.1a) and the other two are very similar melodies with slight differences, one with a note duration error (Figure 4.1b) and the other with a melodic error (Figure 4.1c). Figure 4.2 presents the musical representation used in the listening test, obtained when the target is the audio file correspondent to the score of Figure 4.1a.



(a) Score of the audio file

(b) Temporal error in the 2nd bar (wrong note duration)

(c) Melodic error in the 3rd bar (wrong note)

Figure 4.1: Musical score of the options presented in the last question.

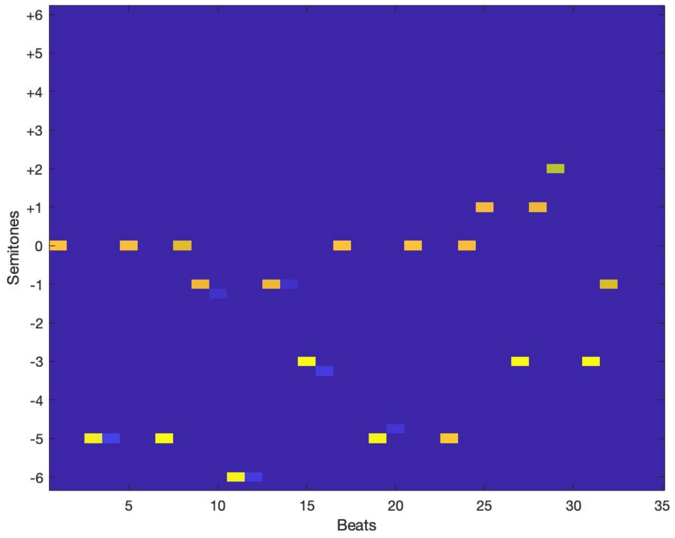


Figure 4.2: Musical representation presented in the last question.

[17] https://www.apple.com/logic-pro/

### 4.1.2 Listening Test

The listening test was conducted in an online platform, *SoGoSurvey*[18] and consisted in three sets of questions. In the first set of questions each participant was asked to rank the mixes obtained using the three approaches. The second set of questions is similar but it is the structure of the reconstructed target that is ranked. The ranking system is the one presented in Figure 4.3.

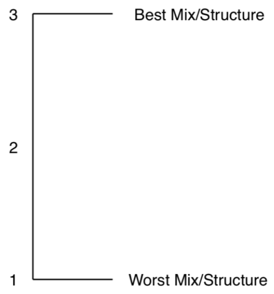


Figure 4.3: Ranking System

The last question was only asked if the participant answered positively to the question regarding their ability to read music. One musical representation is presented and the participant is asked to identify the correct melody after listening back to the options.

## 4.2 Results

In this section the results obtained are presented. The listening test had 19 participants (16 male, 3 female) of which 11 answered yes to the question regarding the ability of reading music or being a musician.

Regarding the first set of questions, that evaluated the quality of the obtained mixes, the mean ranking of each approach for each mix and the overall mean mix ranking are presented in Tables 4.1 and 4.2. In Figure 4.4 a bar chart is presented to help visualise the results.

| Mean Ranking - 1st Set of questions: perceived quality of the mixtures | | | |
|---|---|---|---|
| | Euclidean Chroma | PLCA Chroma | PLCA CQT |
| Mix 1 | 1.45 | 1.79 | 2.74 |
| Mix 2 | 1.21 | 2.53 | 2.26 |
| Mix 3 | 2.63 | 1.16 | 2.21 |
| Mix 4 | 2.00 | 1.84 | 2.16 |

Table 4.1: Mean ranking by mix.

Regarding the second set of questions, that evaluated the quality of the obtained mixes, the mean ranking of each approach for each reconstruction as well as the overall mean reconstruction

[18] https://www.sogosurvey.com

| | Euclidean Chroma | PLCA Chroma | PLCA CQT |
|---|---|---|---|
| Mean | 1.83 | 1.83 | 2.34 |
| StdDev | 0.54 | 0.48 | 0.23 |

Table 4.2: Overall mean of the mixes and standard deviation.

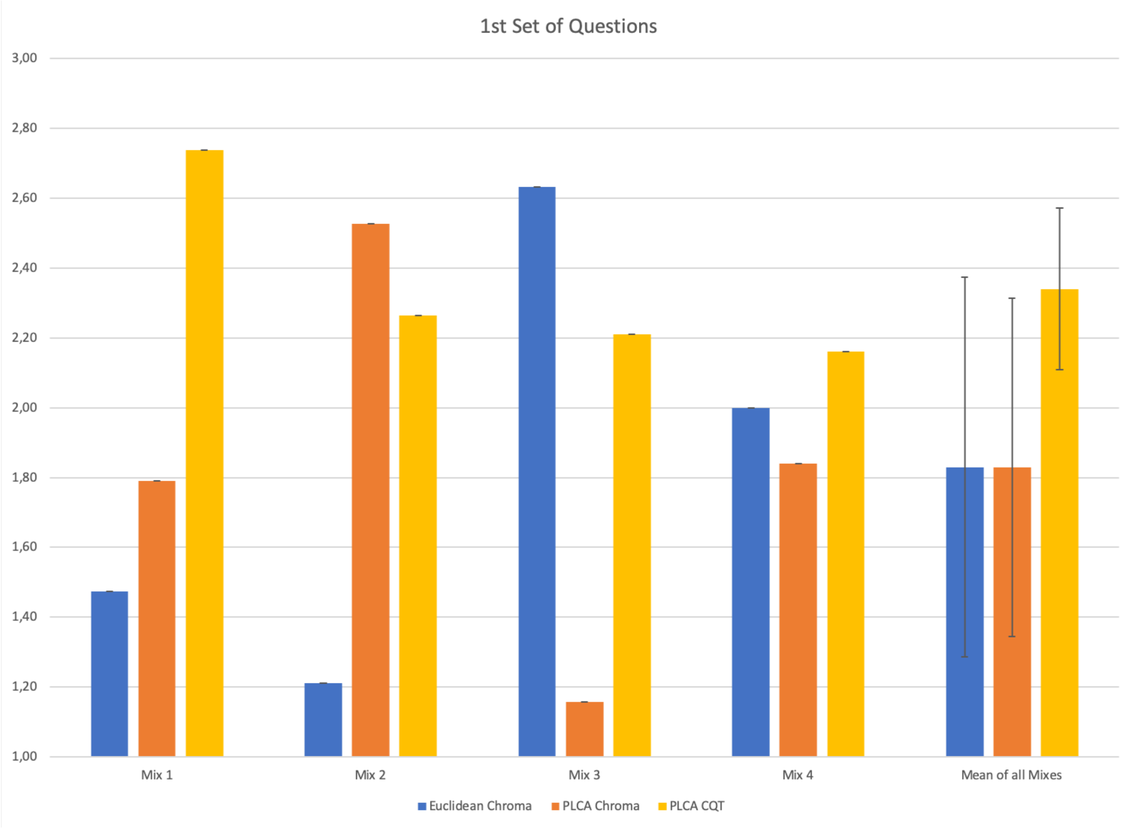


Figure 4.4: Chart graph of Table 4.1 and 4.2.

ranking are presented in Tables 4.3 and 4.4. In Figure 4.5 a chart graph is presented to help visualise the results.

Finally, regarding the last question, that evaluated the quality of a musical representation, the results are presented in Figure 4.6.

## 4.3 Discussion

In this section, the results presented in the previous section are analysed and discussed. The most important and more immediate goal of this experiment was to find out if the results obtained sounded good to the participants. It is important to note that because rankings were obtained, they only provide information in a relative way about the quality of the mixtures. However to complement this information, participants provided positive feedback about the musical and creative

| Mean Ranking - 2nd Set: perceived structure of the reconstructions | | | |
|---|---|---|---|
| | Euclidean Chroma | PLCA Chroma | PLCA CQT |
| Reconstruction 1 | 1.26 | 1.74 | 3.00 |
| Reconstruction 2 | 2.05 | 1.68 | 2.26 |
| Reconstruction 3 | 2.58 | 1.21 | 2.21 |
| Reconstruction 4 | 1.95 | 1.95 | 2.11 |

Table 4.3: Mean ranking by reconstruction.

| | Euclidean Chroma | PLCA Chroma | PLCA CQT |
|---|---|---|---|
| Mean | 1.96 | 1.65 | 2.40 |
| StdDev | 0.47 | 0.27 | 0.35 |

Table 4.4: Overall mean of the reconstructions and standard deviation.

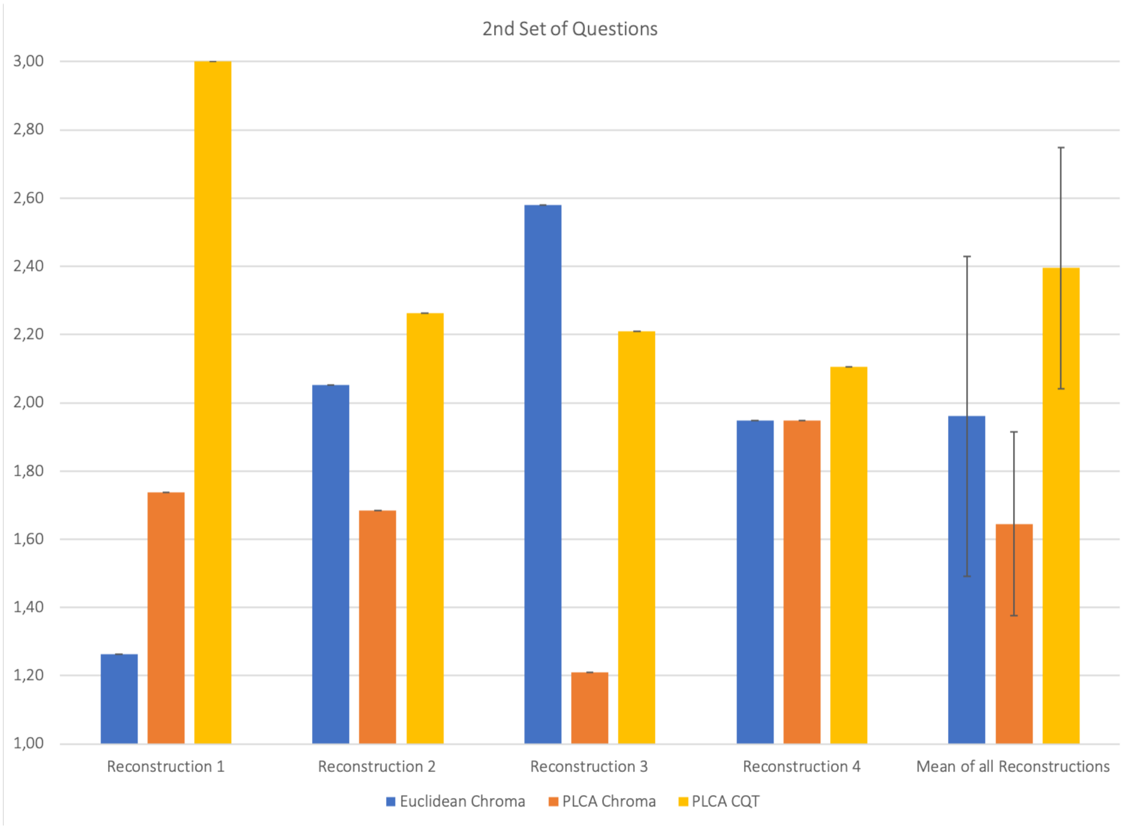


Figure 4.5: Chart graph of Table 4.3 and 4.4.

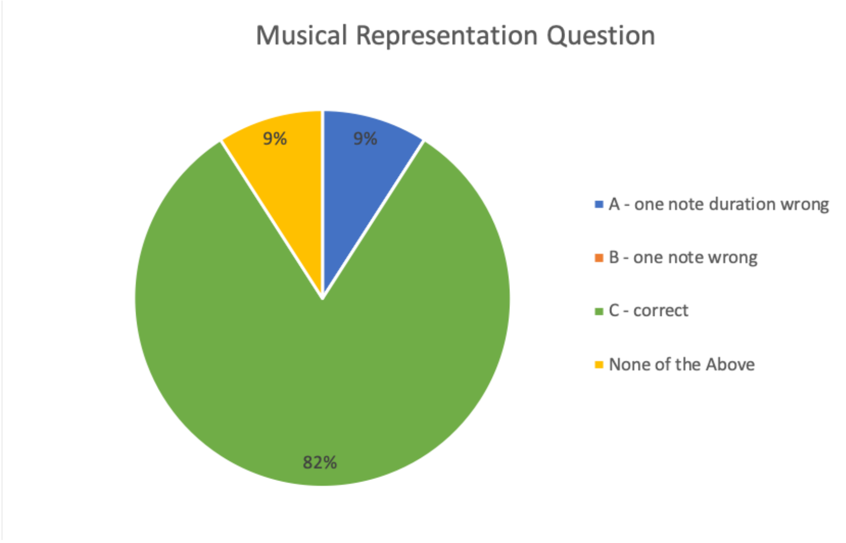


Figure 4.6: Last question results.

quality of the mixtures. To further understand the system, we focus on the harmonic and temporal structure of the reconstructed target, the use of PLCA compared to the Euclidean approach and the use of the CQT compared to the Chromagram.

In the first set of questions the results indicate that the approach with the best and most consistent results is the PLCA using the CQT, which got the highest ranking in two of the examples and the second best in the other two. This is also evidenced by the highest overall mean ranking (mean of the 4 mixes), 2.34, which can be considered an above average mix and by the lowest standard deviation which implies that the results are consistent. The other approaches however show a lower overall mean ranking, 1.83. The standard deviation also increases in both approaches by more than double which can indicate that the results, although sometimes good are not as reliable and consistent as when the PLCA CQT is used. Regarding the 2nd set of questions, the PLCA CQT obtains almost the same overall mean ranking, 2.40, and with the highest ranking in three of the four reconstructions. The PLCA Chroma approach had the worst performance, together with the lowest standard deviation which implies that the results are consistently low. The Euclidean approach improved slightly, maintaining the best rank in the reconstruction where it also got best mix rank. However, it still has the highest standard deviation.

Finally, the results obtained on the last question were very satisfactory. Almost all participants (9 out of 11) gave the correct answer to this question which shows that this type of representation could be useful for automatic melody transcription systems. One of the reasons for the very satisfying results is the use of the Beat-by-Beat CQT, that filters a lot of unnecessary information allowing the representation to be very clean and understandable.

It is also interesting to interpret the results considering the genres of the source and target songs. In the first mix, the source and target both come from the same song. The PLCA CQT in this case had a much higher score than the other approaches in both set of questions. The results offered by the PLCA Chroma and PLCA CQT are quite similar, however, the PLCA CQT is basically a loop from a small section of the source, mostly unaltered. This can show that the use of the CQT can make the PLCA find detailed harmonical relations between the source and target beats, so that less abrupt changes need to be made to the source beats, allowing for better and structured results. In mix number 2, the source is a hip-hop verse and target is a pop chorus. In this case, vocals are used to match a pop chorus. Considering both sets of questions there is only one case where the results are considerably worse, the Euclidean approach, possibly because the vocal part of the reconstruction does not sounds very coherent. However, the PLCA approaches improve this aspect. A possible explanation is that the simultaneous activations increase the probability of small vocals samples being in sequence causing to the result to sound more logical and musical. In the third mix the opposite happens, an indie instrumental verse is the source and the same hip-hop verse is the target. The Euclidean approach offered a very compelling result as shown by the results. However, the other chroma approach got bad results meaning that the use of the Chromagram was not what impacted the final result. This can demonstrate that the Euclidean approach, although simpler, can also sometimes offer valid results even when compared with more complex approaches. Finally, in the last mix, where the source and target are from a similar genre, the results were very evenly distributed, with none of the approaches standing out, even though the CQT PLCA had the best ranking in both sets of questions.

As stated, some feedback on the mixtures and reconstructions was gathered from the participants. While some feedback revealed an overall appreciation and curiosity regarding the results obtained there was also some more specific feedback. On the first mix there was an overall agreement that one of the mixes/reconstructions sounded a lot better than the others, the PLCA CQT one, having a very tight structure, almost like it had been arranged by a human. For the example where the Euclidean approach got the best ranking both mix and structure wise the feedback received indicated that the participants enjoyed the "choppy sampling" flow provided by the Euclidean approach, being also described as more danceable and therefore more enjoyable than the other approaches which offered a more stale result. Another mix that got some comments was the last one, which got very close results between all of the approaches. While some participants stated that they liked more the drum beat offered by the Euclidean approach others preferred the bass line present in the CQT PLCA, which can, to some extent explain, the almost even ranking values between all the approaches. In the wider context, this can imply that multiple good solutions can exist which draw upon different musical properties between the source and the target.

One last graphic that can give some more insight into the results is the scatter graphic presented in Figure 4.7, where the two variables are the mean ranking of the mixes and of the reconstructions. Considering all the points, independently of the approach, a strong relationship between the structural ranking and the mix quality can be found. Only in two cases the values are not closely related.

Although the results obtained in the PLCA CQT are better and more consistent overall, no obvious conclusion can be taken regarding the use of the PLCA when compared with the Euclidean approach. The most direct conclusion that can be taken from the results is that the use of the CQT is what tends to improve the results when compared to the Chromagram. If both PLCA

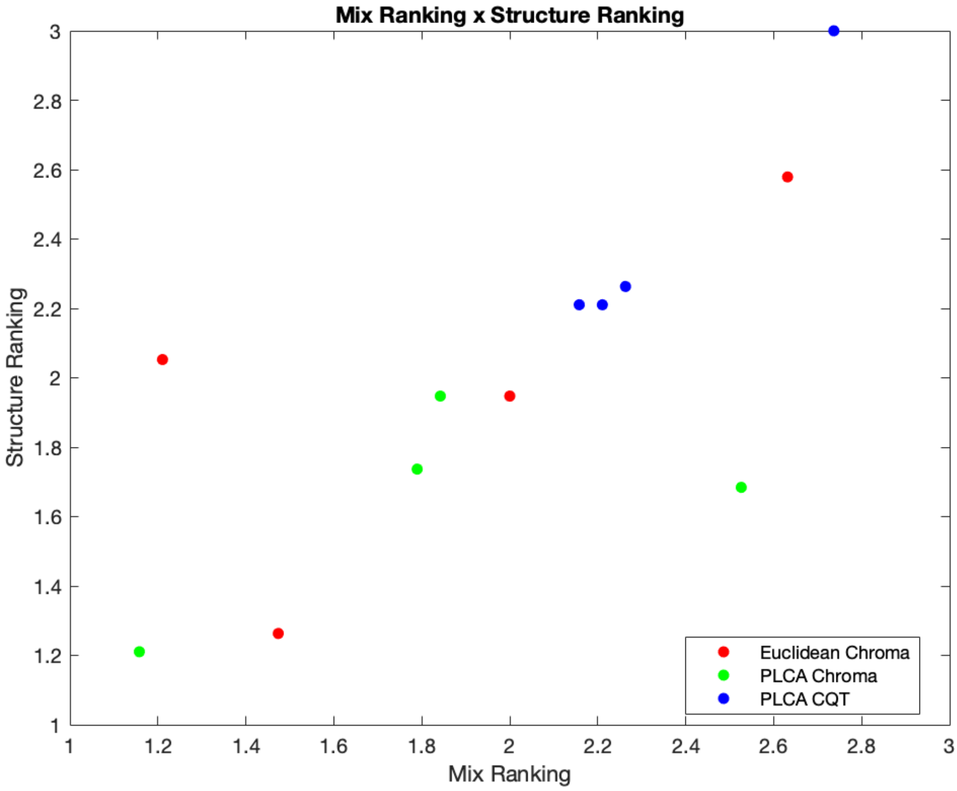


Figure 4.7: Relationship between the mix ranking and the structure ranking.

approaches are compared, the CQT version performs better in almost every case than the Chromagram version, sometimes with a quite high difference between the two. However if we compare the two Chromagram approaches, there is no observable pattern that demonstrates that the use of the PLCA improves the results. This can be because the Chromagram is a very simple representation, that may not take full advantage of a complex algorithm like the PLCA. The CQT, being a lot more complete and complex (every note is represented separately instead of being basically an histogram of notes like the Chromagram) may take more advantage of the PLCA. Also, the better tuning correction implied with the CQT can also subjectively improve the results, even if this was not explicitly tested in the experiment.

# Chapter 5

# Conclusion and Future Work

## 5.1 Summary

In this dissertation a novel approach to manipulate audio signals with the purpose of cross-synthesis has been presented. It has a different goal than most state-of-the-art cross-synthesis systems presented in Chapter 2, offering the users more musical results than systems like the Factorsynth [44] as this system does not focus on precisely recreating the target using snippets of the source but on transforming the source in order to sound good when combined with the target. This system offers a flexible framework that allows for simple user interaction, through a GUI, that impacts the final result. It allows the user to choose any audio files he/she wants as the source or target due to beat calculation, correction and time stretching algorithms being implemented in the system. For the input representation it uses either the CQT, which allows for a detailed representation of the system and better tuning correction, or the Chromagram, that although simpler, can also obtain interesting results sometimes at a fraction of the time when compared to the CQT. It takes advantage of PLCA to enhance the cross-synthesis process, by allowing the layering of multiple basis functions (beat-synchronised vectors from the source) to better match a target beat. Finally, to create an audio file comparable with the target in terms of loudness, a beat-by-beat RMS normalization algorithm was developed and is applied to the result.

The evaluation of the system was done by performing a subjective listening test to evaluate the impact of the use of the PLCA compared against a Euclidean distance approach as well as the use of the CQT against the Chromagram. Simple statistical measures like the mean and standard deviation were used and showed that when using the PLCA in conjunction with the CQT the results are more satisfactory both in terms of fitting the target and of harmonic and temporal structure. However, the other approaches presented in the test also obtained good results in some specific cases. Another interesting result consisted in a musical representation for monophonic melodies that could be potentially used in music transcription systems which was also evaluated and very good results were obtained.

## 5.2 Future Work

Despite the promising results described in chapter 4, there are some features that can be added to this system in order to improve its usability and performance. Due to time restrictions and the considerable amount of work some of these features needed, they are not yet implemented or tested enough to be included in the system. Some of the these planned improvements are:

- **Improving the pitch-shifting:** As mentioned before, when using the CQT the tuning of the reconstructed target is changed to match that of the target but sometimes the different pitch-shifts for each beat obtained are not in the same tuning, e.g, instead of all the pitch-shifts being apart by semitones they are apart by divisions of a semitone which can cause the reconstructed track to not have a constant tuning. A method of finding the best compromise between tuning correction and having all the pitch-shifts as close to the same tuning as possible would likely improve the obtained results in this aspect.

- **More Listening Tests:** More testing should be done in order to find if there is a type of input that causes the PLCA CQT approach to offer better results. Some correlation between the use of audio snippets from the same song as the source and target and considerably better results regarding the mix and especially the structure when the PLCA CQT is used has already been found. Listening tests focused on these kind of examples would confirm if this was just an isolated case or if there is an observable pattern.

- **Improve the musical representation result:** Exploring ways of transcribing not just monophonic melodies but also polyphonic ones. This was tried and while some information about the harmonic movement such as the musical intervals present in a certain beat can be retrieved the results were not very reliable and couldn't yet be used in a music transcription context. Some related work that could be used as a starting point is presented in [47].

- **More control over the basis functions:** As of now the GUI only allows the user to have simple access to the algorithm, but some features, especially regarding the basis functions, could really make the process of interacting with the system more engaging. Some of the considered features consist in having a simple way of combining the basis functions of multiple sources (as of now, a maximum of two sources can be combined only when the CQT is used as the representation), allowing the user to listen to the basis functions so that he/she can subsequently select which ones he/she wants to use to create the reconstructed target and combining the activation matrix from a certain source/target with the basis functions of another source.

- **Include the Euclidean approach in the GUI:** As shown the Euclidean approach can also obtain musically interesting results, so adding this feature to the system would further extend its flexibility.

- **Better audio file player:** Add to the GUI the possibility of saving and playing back results so that they are easily comparable between each other.

- **Developing an audio-fingerprinting system:** Some informal testing showed that state-of-the-art audio fingerprinting tools did not recognise the source song used to create the reconstructed target. However, for humans this task is easily performed, so, an interesting next step would be creating a system that could identify the source file used even in its pitch-shifted and rearranged form.

## 5.3 Perspective on the project

Although this project started with a different goal that followed a more common cross-synthesis approach of reconstructing a target song using a source song so that both were easily recognisable, it quickly developed into what has been presented so far, mostly because early versions of the algorithm showed that the use of larger temporal windows (i.e. beats) created better musical results, and with different application possibilities.

To show the participants and other people the results obtained with this project was a rewarding experience as many showed interest in using the system themselves. This project also provided me with a lot more insight in computer processing of musical signals as well as the use of machine learning techniques in this context.

A different point of discussion, a more ethical one, is about the ownership of the musical results. As the results were uploaded into *Soundcloud* to be included in the listening test, *Soundcloud* copyright system did not flag the files as being copyrighted songs. After noticing this, the same was tested with state-of-the-art audio fingerprint tools such as *Shazam* that claims to identify songs almost instantly. *Shazam* also could not recognise any of the results that were played to it. This raises an interesting question. What degree of manipulation is sufficient to have some kind of ownership over the musical result? Although humans can, in most cases, easily identify the manipulated song if they are familiar with the original version, these algorithms that tend to prevent the usage of copyrighted content cannot do it. This topic of discussion will become more relevant over the next decades, as the use of machine learning is starting to be integrated in the music creation process. Who owns the music when it is an algorithm that makes almost all the composition choices? Even from a commercial perspective, will labels even require artist to create music or will artists become an outdated commodity?

# References

[1] Alexandre Papadopoulos, Pierre Roy, and François Pachet. Assisted Lead Sheet Composition Using FlowComposer. In *Principles and Practice of Constraint Programming*, pages 769–785, Cham, 2016. Springer International Publishing.

[2] Gaëtan Hadjeres and François Pachet. Deepbach: a steerable model for bach chorales generation. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pages 1362–1371, 2017.

[3] Fiammetta Ghedini, François Pachet, and Pierre Roy. *Creating Music and Texts with Flow Machines*, pages 325–343. Springer Singapore, Singapore, 2016.

[4] Christopher C. Johnson. Logistic Matrix Factorization for Implicit Feedback Data. In *Advances in Neural Information Processing Systems*, 2014.

[5] Jonathan Driedger, Thomas Prätzlich, and Meinard Müller. Let It Bee – Towards NMF-Inspired Audio Mosaicing. In *ISMIR*, pages 350–356, 2015.

[6] S. Su, C. Chiu, L. Su, and Y. Yang. Automatic conversion of pop music into chiptunes for 8-bit pixel art. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 411–415, March 2017. doi:10.1109/ICASSP.2017.7952188.

[7] Peter Manning. *Electronic and Computer Music*. Oxford University Press, Inc., New York, NY, USA, 2004.

[8] J.W. Cooley and J.W. Tukey. An Algorithm for the Machine Calculation of Complex Fourier Series. *Mathematics of Computation*, 19:297–301, 1965. doi:10.1090/S0025-5718-1965-0178586-1.

[9] Martin Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, Manjunath Kudlur, Josh Levenberg, Rajat Monga, Sherry Moore, Derek G. Murray, Benoit Steiner, Paul Tucker, Vijay Vasudevan, Pete Warden, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.

[10] E. T. Jaynes. *The Physical Basis Of Music*. Washington University, 1996.

[11] Werner Meyer-Eppler. *Elektrische Klangerzeugung : elektronische Musik und synthetische Sprache*. Bonn , Dümmler, 1949.

[12] M. Brend. *Strange Sounds: Offbeat Instruments and Sonic Experiments in Pop*. A Backbeat book. Backbeat, 2005. URL: https://books.google.pt/books?id=m6KRDxYOp4UC.

[13] John Chowning. Synthesis of complex audio spectra by means of frequency modulation. *AES: Journal of the Audio Engineering Society*, 21:526–534, 1973.

[14] Juan Chattah. Music instrument digital interface (midi) - music in the social and behavioral sciences. *Encyclopedia of Music in the Social and Behavioral Sciences*, II:789–791, 08 2014.

[15] Adam Behr, Keith Negus, and John Street. The sampling continuum: musical aesthetics and ethics in the age of digital production. *Journal for Cultural Research*, 21(3):223–240, 2017.

[16] Markus Schedl, Emilia Gómez, and J. Urbano. Music information retrieval: Recent developments and applications. *Foundations and Trends in Information Retrieval*, 8:127–261, 2014.

[17] E.J. Humphrey, D. Turnbull, and T. Collins. A brief review of creative MIR. In *Late-Breaking News and Demos presented at the Int. Conf. on Music Information Retrieval.*, 2013.

[18] H.K.G. Andersen and Peter Knees. Conversations with expert users in music retrieval and research challenges for creative MIR. In *17th International Society for Music Information Retrieval Conference (ISMIR), 7-11 August 2016, New York, New York*, pages 122–128. International Society for Music Information Retrieval, 2016.

[19] Matthew Davies, Philippe Hamel, Kazuyoshi Yoshii, and Masataka Goto. AutoMashUpper: Automatic Creation of Multi-Song Music Mashups. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 22:1726–1737, 2014.

[20] Garth Griffin, Youngmoo E. Kim, and Douglas Turnbull. Beat-sync-mash-coder: A web application for real-time creation of beat-synchronous music mashups. *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 437–440, 2010.

[21] Gérard Assayag, Georges Bloch, Marc Chemillier, Arshia Cont, and Shlomo Dubnov. Omax brothers: a dynamic yopology of agents for improvization learning. In *AMCMM*, 2006.

[22] Shlomo Dubnov, Gérard Assayag, and Arshia Cont. Audio oracle analysis of musical information rate. *2011 IEEE Fifth International Conference on Semantic Computing*, pages 567–571, 2011.

[23] Rebecca Fiebrink, Dan Trueman, and Perry Cook. A meta-instrument for interactive, on-the-fly machine learning. In *Proceedings of the International Conference on New Interfaces for Musical Expression*, pages 280–285, 2009.

[24] Markus Schedl, Arthur Flexer, and Julián Urbano. The neglected user in music information retrieval research. *Journal of Intelligent Information Systems*, 41(3):523–539, Dec 2013.

[25] Markus Schedl and Arthur Flexer. Putting the user in the center of music information retrieval. In *International society for music information retrieval conference*, page 385–390, 2012.

[26] Judith C. Brown. Calculation of a constant Q spectral transform. *The Journal of the Acoustical Society of America*, 89(1):425–434, 1991.

[27] Gino Angelo Velasco, Nicki Holighaus, Monika Doerfler, and Thomas Grill. Constructing an invertible constant-Q transform with nonstationary Gabor frames. In *Proceedings of the 14th International Conference on Digital Audio Effects*, pages 93–99, 2011.

[28] Florent Jaillet, Peter Balazs, and Monika Doerfler. Nonstationary Gabor Frames. In *Proceedings of the 8th international conference on Sampling Theory and Applications (SAMPTA'09)*, 2009.

[29] Christian Schörkhuber, Anssi Klapuri, and Alois Sontacchi. Audio Pitch Shifting Using the Constant-Q Transform. *Journal of the Audio Engineering Society*, 61:562–572, 2013.

[30] Mark A. Bartsch and Gregory H. Wakefield. Audio Thumbnailing of Popular Music Using Chroma-Based Representations. *Multimedia, IEEE Transactions on*, 7:96–104, 2005. doi:10.1109/TMM.2004.840597.

[31] Roger N. Shepard. Circularity in Judgments of Relative Pitch. *The Journal of the Acoustical Society of America*, 36(12):2346–2353, 1964.

[32] Matthias Mauch and Simon Dixon. Approximate Note Transcription for the Improved Identification of Difficult Chords. In *Proceedings of the 11th International Society for Music Information Retrieval Conference, ISMIR 2010, Utrecht, Netherlands, August 9-13, 2010*, pages 135–140, 2010.

[33] Daniel Lee and H. Sebastian Seung. Learning the Parts of Objects by Non-Negative Matrix Factorization. *Nature*, 401:788–91, 1999.

[34] Daniel Lee and H.Sebastian Seung. Algorithms for Non-negative Matrix Factorization. In *Proceedings of the 13th International Conference on Neural Information Processing Systems*, NIPS'00, pages 535–541, Cambridge, MA, USA, 2000. MIT Press.

[35] Tran Hien, Do Tuan, Pham Van At, and Le Hung Son. Novel Algorithm for Non-Negative Matrix Factorization. *New Mathematics and Natural Computation*, 11:121–133, 2015.

[36] Paris Smaragdis and Bhiksha Raj. Shift-Invariant Probabilistic Latent Component Analysis. Technical report, Mitsubishi Electric Research Laboratories, 2007.

[37] Cédric Févotte, Emmanuel Vincent, and Alexey Ozerov. Single-channel audio source separation with NMF: divergences, constraints and algorithms. In *Audio Source Separation*. Springer, March 2018. URL: https://hal.inria.fr/hal-01631185.

[38] Mikkel N. Schmidt and Morten Morup. Nonnegative Matrix Factor 2-D Deconvolution for Blind Single Channel Source Separation. In *International Conference on Independent Component Analysis and Signal Separation*, pages 700–707, 2006.

[39] Sarver Ryan and Anssi Klapuri. Applications of non-negative matrix factorization to signal-adaptative audio effects. In *Proceedings of the 14th International Conference of Digital Audio Effects (DAFx-11)*, pages 249–252, 2011.

[40] Diemo Schwarz. Current Research in Concatenative Sound Synthesis. In *International Computer Music Conference (ICMC)*, pages 1–4, Barcelona, Spain, Sep 2005.

[41] Diemo Schwarz, Grégory Beller, Bruno Verbrugghe, and Sam Britton. Real-Time Corpus-Based Concatenative Synthesis with CataRT. In *Proceedings of the 9th International Conference on Digital Audio Effects, DAFx 2006*, pages 279–282, 2006.

[42] Gilberto Bernardes, Carlos Guedes, and Bruce Pennycook. EarGram: An Application for Interactive Exploration of Concatenative Sound Synthesis in Pure Data. In *From Sounds to Music and Emotions*, pages 110–129, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[43] Christopher J. Tralie. Cover Song Synthesis by Analogy. In *ISMIR*, pages 197–203, 2018.

[44] Juan José Burred. Factorsynth: A max tool for sound analysis and resynthesis based on matrix factorization. In *Proceedings of the Sound and Music Computing Conference 2016, SMC 2016, Hamburg, Germany*, 2016.

[45] Christian Schörkhuber, Anssi Klapuri, Nicki Holighaus, and Monika Doerfler. A Matlab Toolbox for Efficient Perfect Reconstruction Time-Frequency Transforms with Log-Frequency Resolution. In *Proceedings of the AES International Conference*, 2014.

[46] Gilberto Bernardes, Matthew Davies, and Carlos Guedes. Automatic Musical Key Estimation with Adaptive Mode Bias. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 316–320, March 2017.

[47] Paris Smaragdis and Judith C. Brown. Non-negative matrix factorization for polyphonic music transcription. In *2003 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (IEEE Cat. No.03TH8684)*, pages 177–180, Oct 2003.