



Onset Detection and Chord Recognition

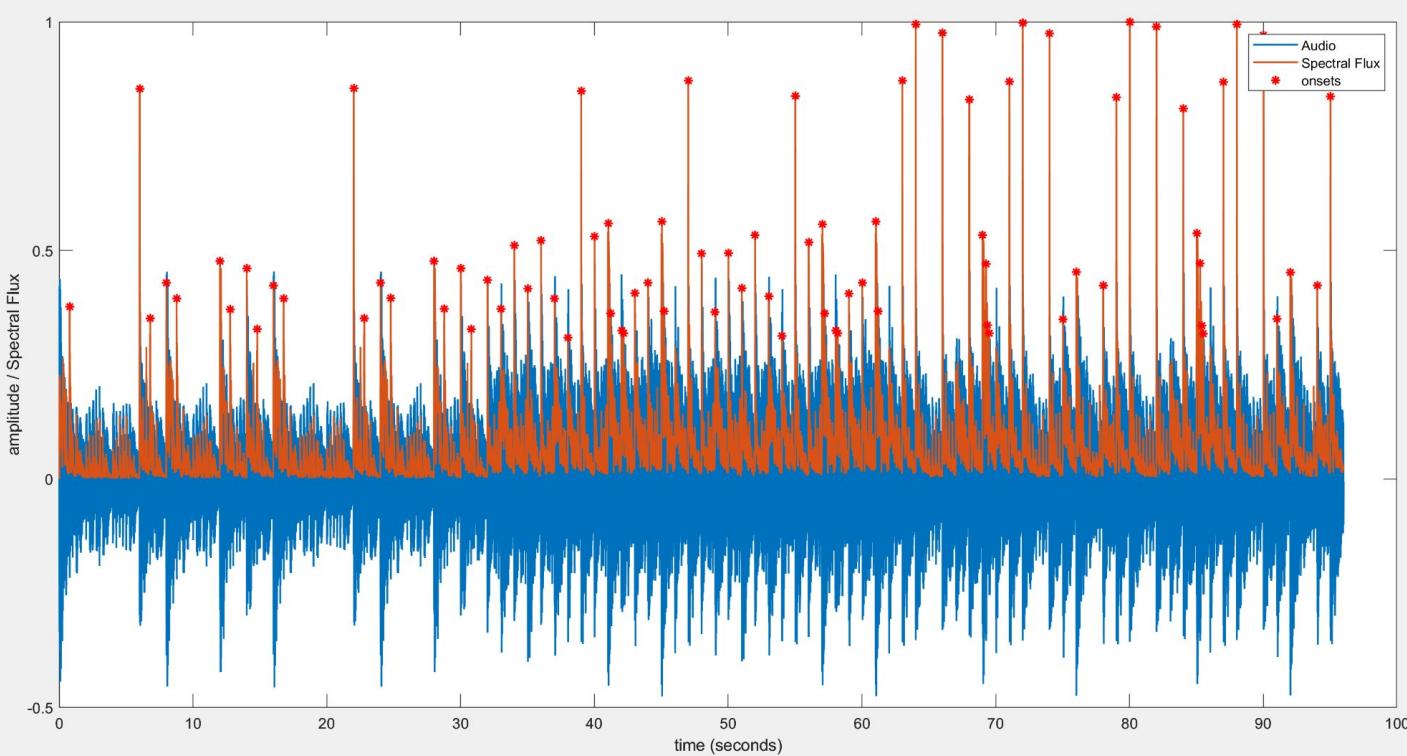
Dr. Michael McLoughlin (he/him) - AudioLab Genesis 6
Contact: michael.mcloughlin@york.ac.uk



Onset Detection

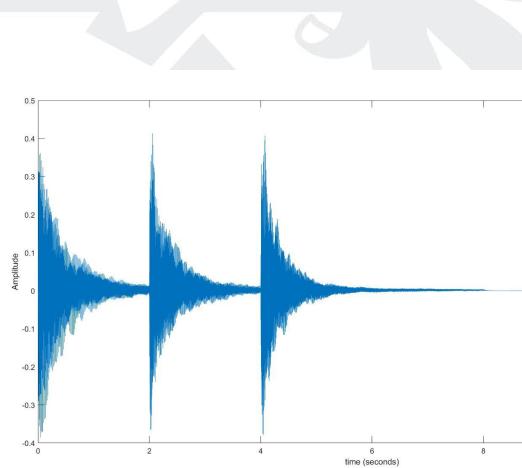
What is onset detection?

Figuring out when a sound starts

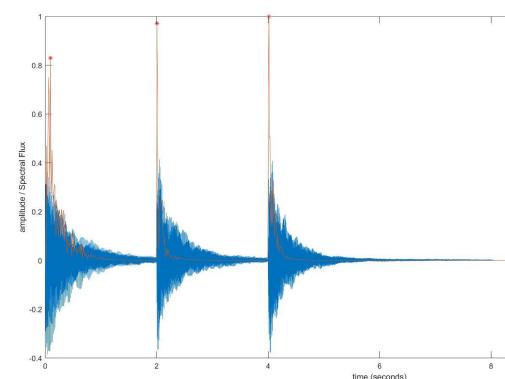


An Example of an MIR Algorithm Pipeline

Audio Input



Onset Detection



Application

Digital Audio FX (gates,
drum triggers,
sidechaining)

Chord
Recognition

Instrument
classification

Performance
Analysis

Speaker
Identification

How do we do Onset Detection?

TMTOWTDI - There is More Than One Way To Do It - For those of you who are fans of the Perl programming language

Time domain approach?

Frequency Domain Approach?

Phase based approach?

Manual Approach?!

All of the above!?!?

Machine Learning Approach?

Recommended Reading

Proc. of the 9th Int. Conference on Digital Audio Effects (DAFx'06),
Montreal, Canada, September 18-20, 2006

ONSET DETECTION REVISITED

Available on the VLE!

Simon Dixon

Austrian Research Institute for Artificial Intelligence
Freyung 6/6, Vienna 1010, Austria
simon.dixon@ofai.at

Onset detection Pipeline

Step 1 Step 2 Step 3 Step 4 Step 5 Step 6

Load Digital
Audio

FFT

Feature
Extraction* -
Spectral Flux

Rescale and
Window
feature
vector

calculate
local
maximum in
windows

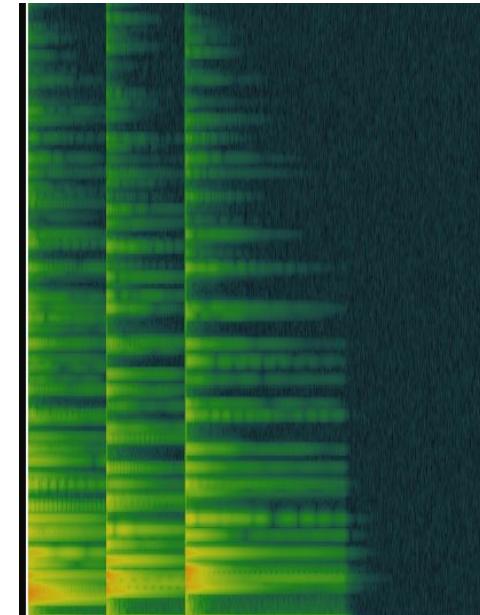
Identify
onsets by
comparing
local
maximums to
a threshold

*We'll only be covering Spectral Flux here, but take a look at the Simon Dixon paper in the Lecture 2 folder on the VLE to see what other types of features can be used

Steps 1 and 2 Recap

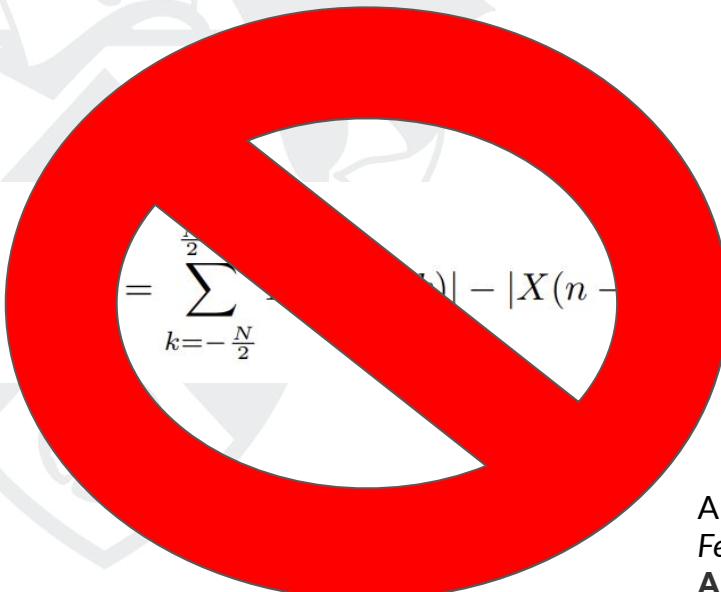


0
0.0000
0.0001
0.0000
0.0000
0
0.0001
0.0002
0.0002
0.0003
0.0002
0
-0.0001
-0.0001
-0.0003
-0.0007
-0.0010
-0.0013
-0.0014
-0.0014
-0.0013
-0.0012
-0.0014
-0.0019
-0.0027
-0.0033
-0.0034
-0.0030
-0.0030



Step 3: Spectral Flux - TMTOWTDI

$$\text{flux}(t) = \left(\sum_{k=b_1}^{b_2} |s_k(t) - s_k(t-1)|^P \right)^{1/P}$$



where

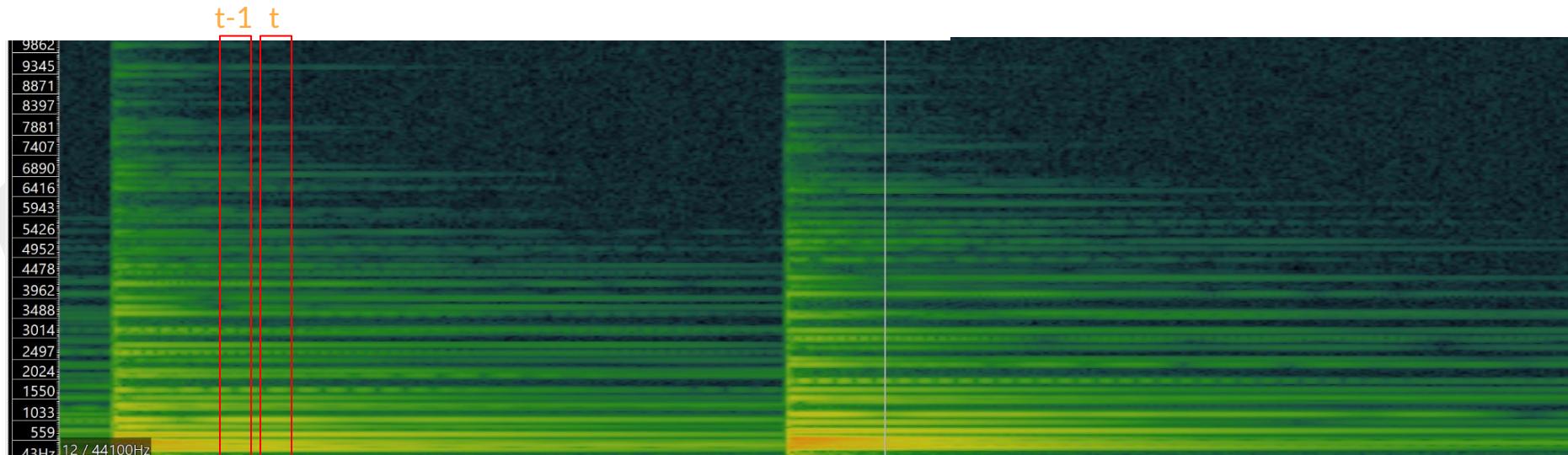
- s_k is the spectral value at bin k .
- b_1 and b_2 are the band edges, in bins, over which to calculate the spectral flux.
- P is the norm type. You can specify the norm type using [NormType](#).

A deeper explanation of spectral flux is available in the *Instantaneous Features* chapter of **An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics** by Alexander Lerch.

Spectral Flux contd.

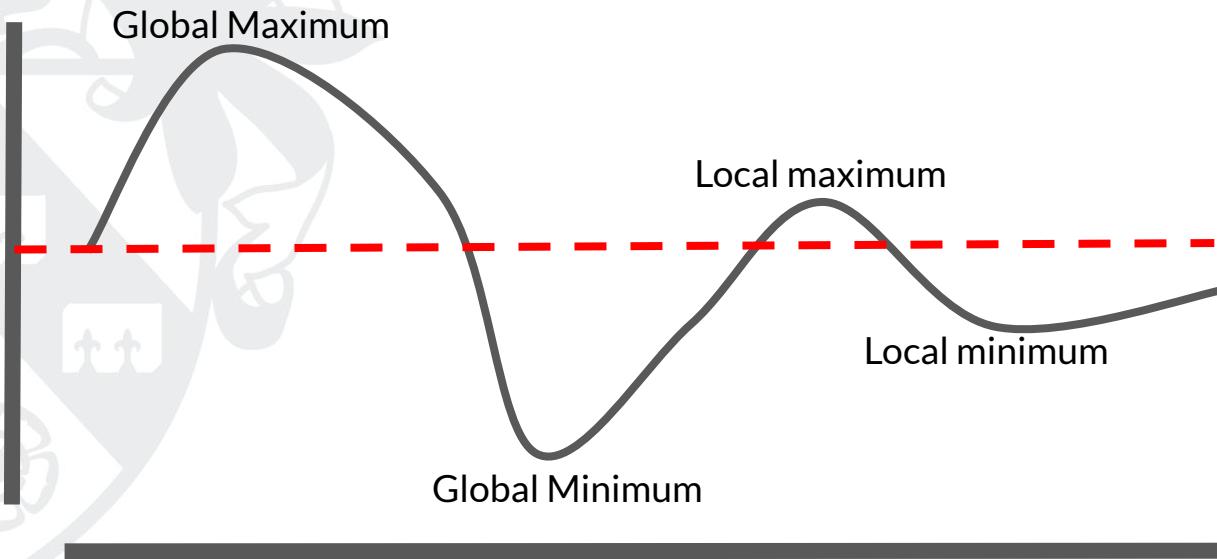


$$\text{flux}(t) = \left(\sum_{k=b_1}^{b_2} |s_k(t) - s_k(t-1)|^P \right)^{1/P}$$



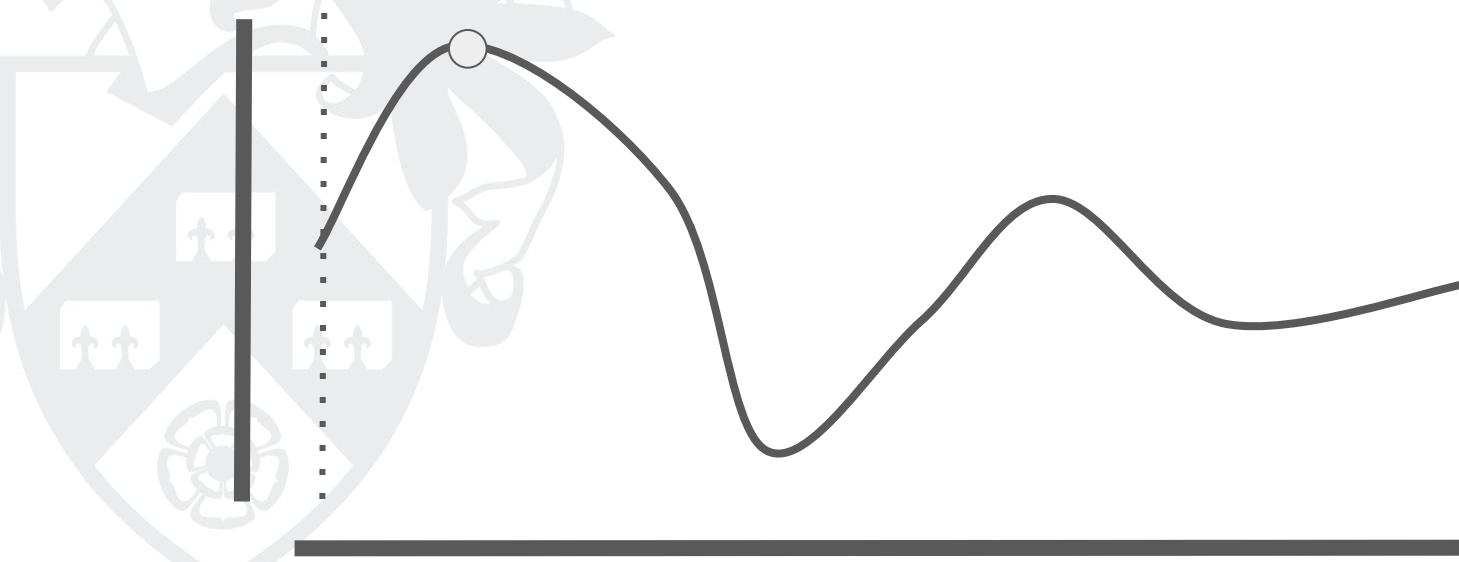
Step 4: Rescaling and Windowing
Feature Vectors - Let's jump over to
Matlab!!

Step 5: Global/Local Minima and Maxima

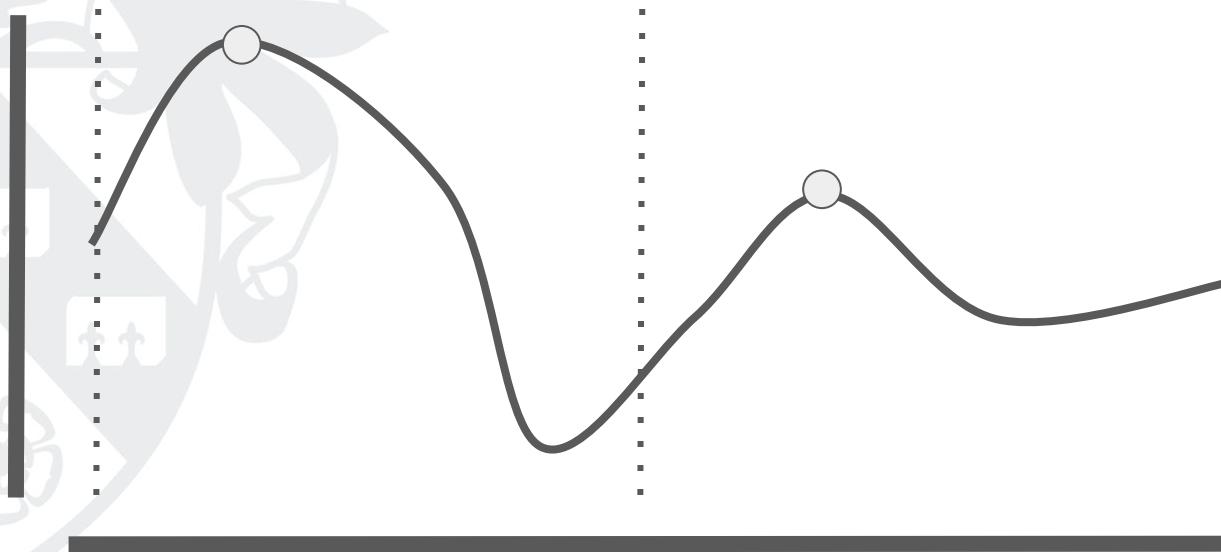


Global/Local Minima and Maxima

- Effect of Window Size



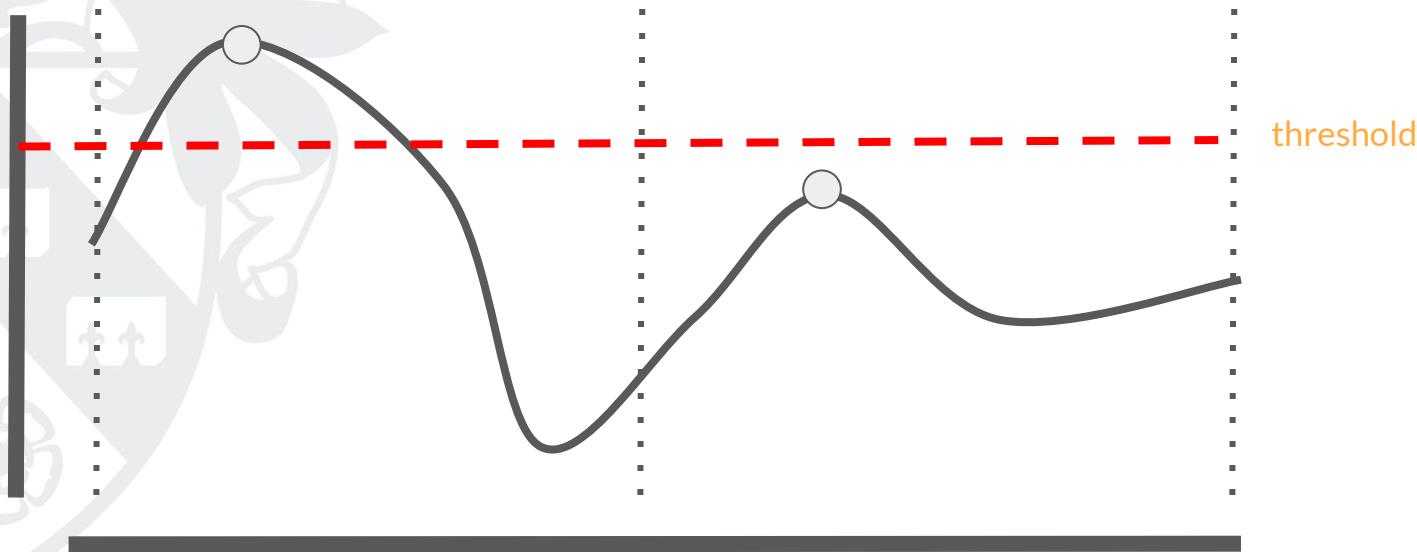
Global/Local Minima and Maxima - Effect of Window Size



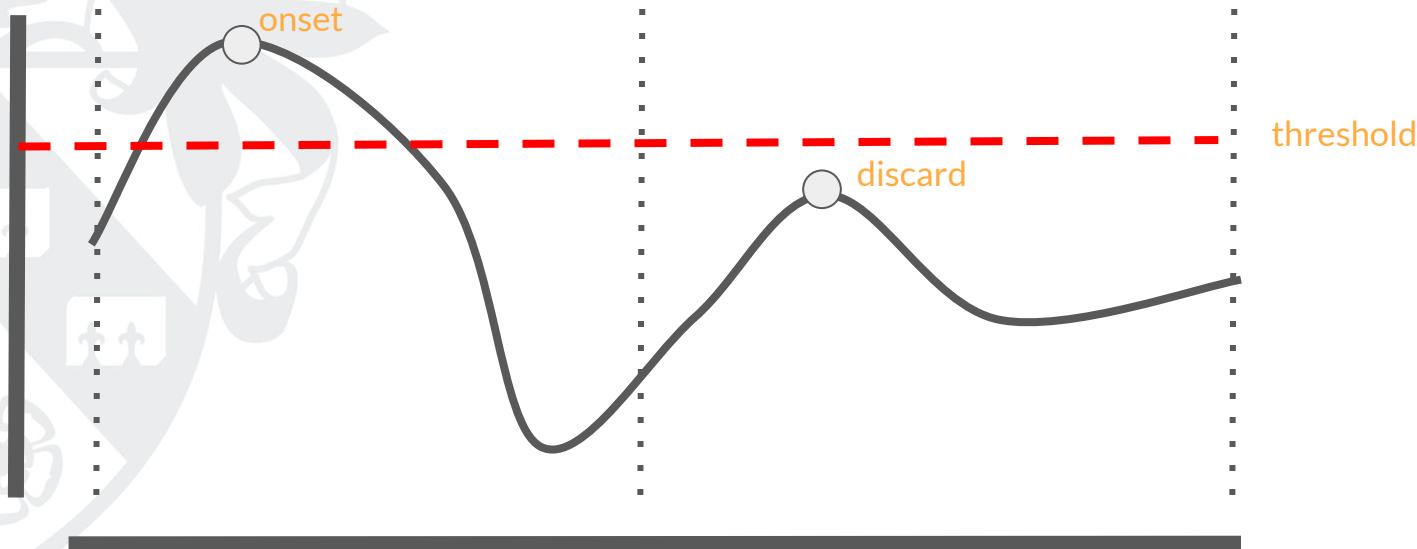


UNIVERSITY
of York

Step 6: Compare Local Maxima to threshold



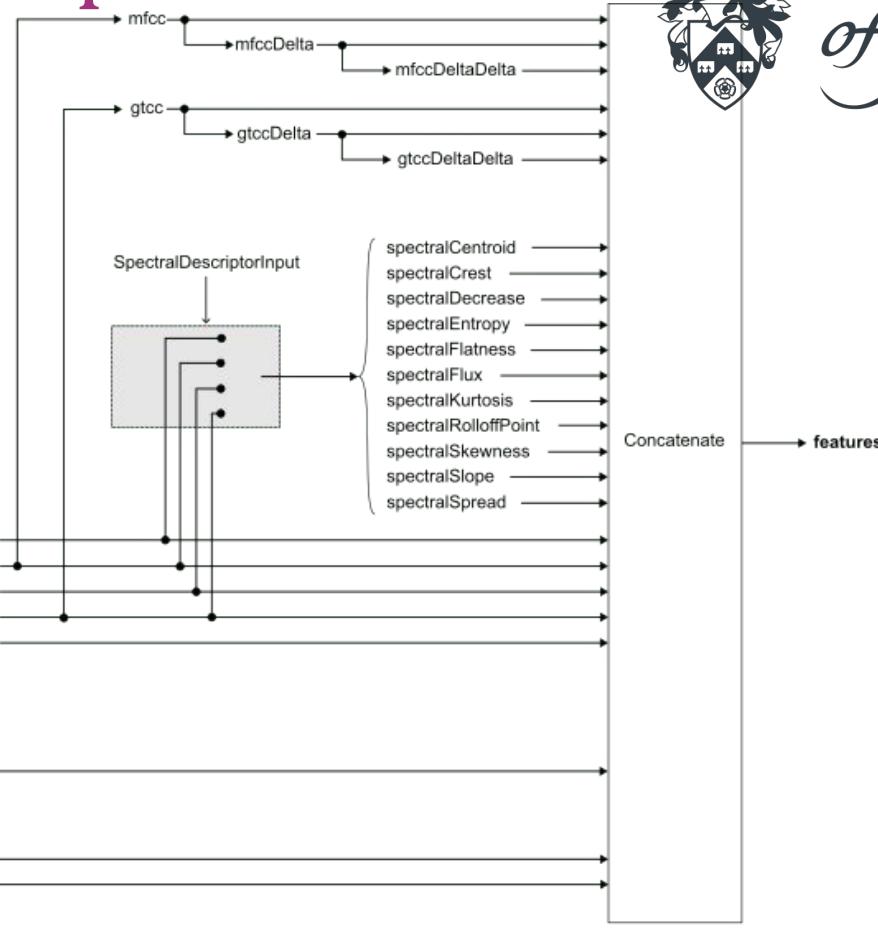
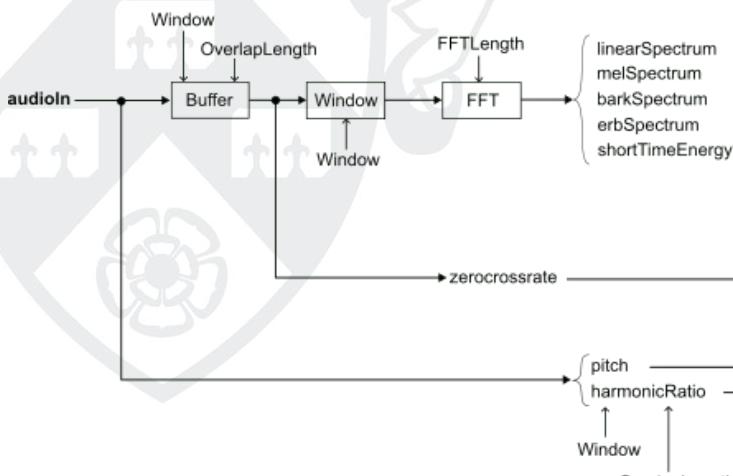
Step 6: Compare Local Maxima to threshold





Introduction to Spectral Features

Feature Extraction Pipeline in MATLAB



UNIVERSITY
of York

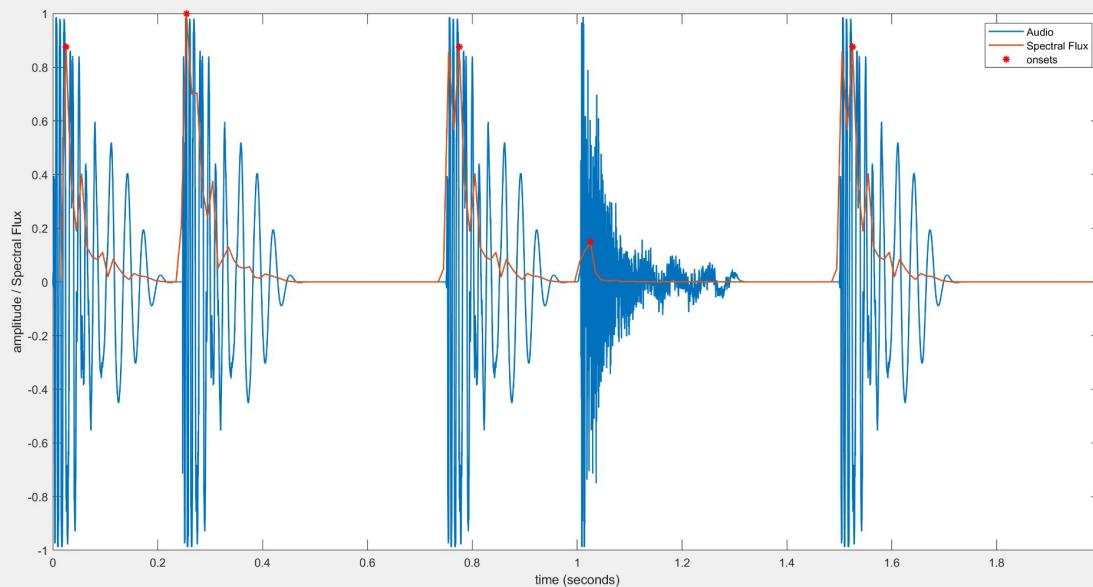
Putting it all together: A simple MIR system for drum transcription

Think about what we have done so far.
Do we have the tools to make a tool
for simple drum transcription?

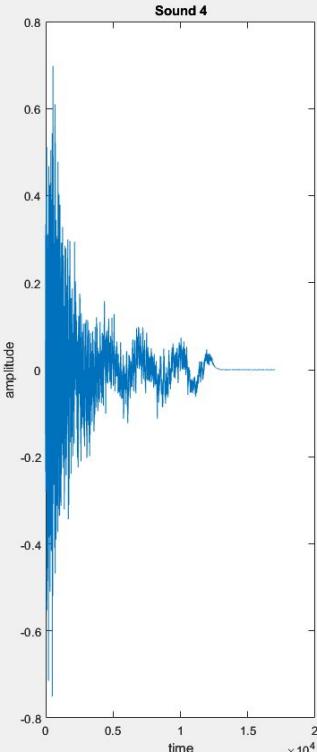
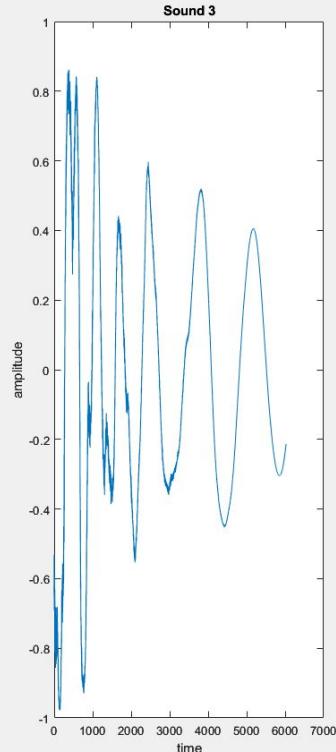
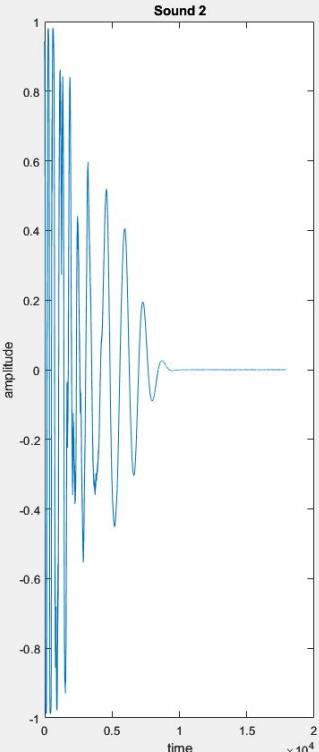
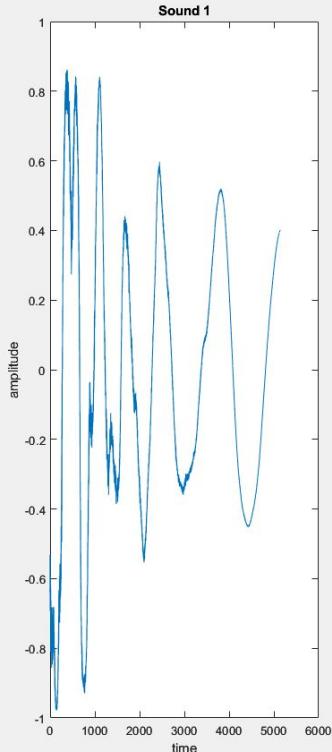
Putting it all together: A simple MIR system for drum transcription



Step 1: Use Onset detection to identify when drum hits start

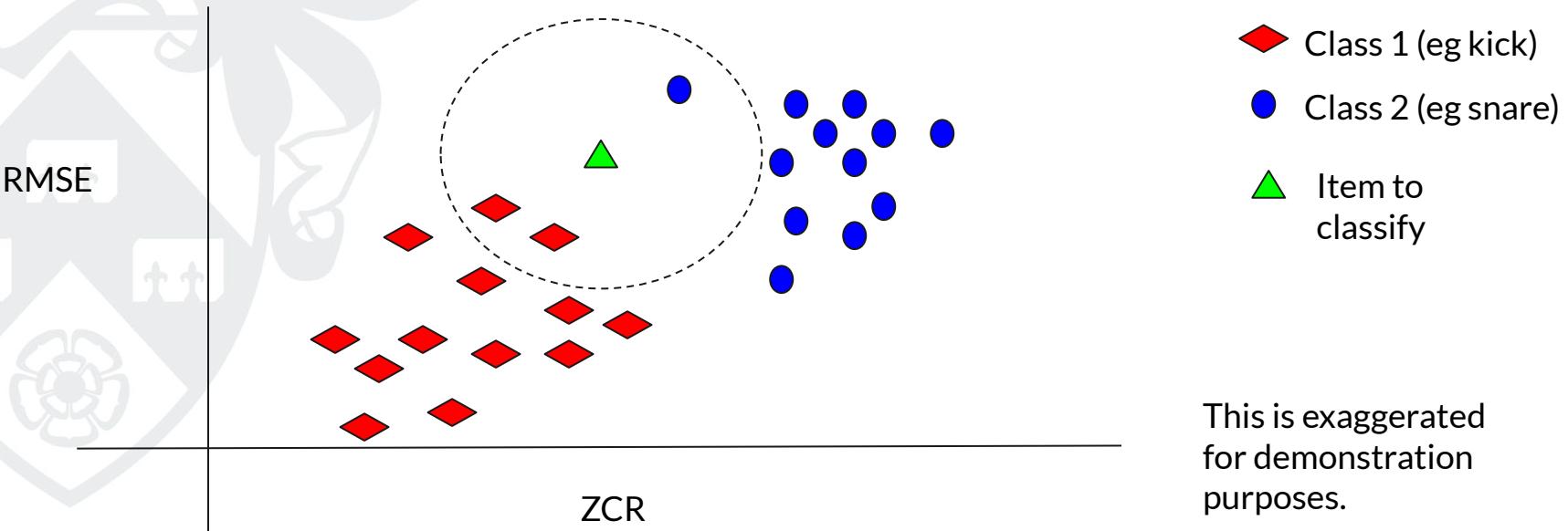


Step 2: Use Onset detection to



Step 3: Use a pre-trained k-Nearest Neighbours Algorithm to classify each sound

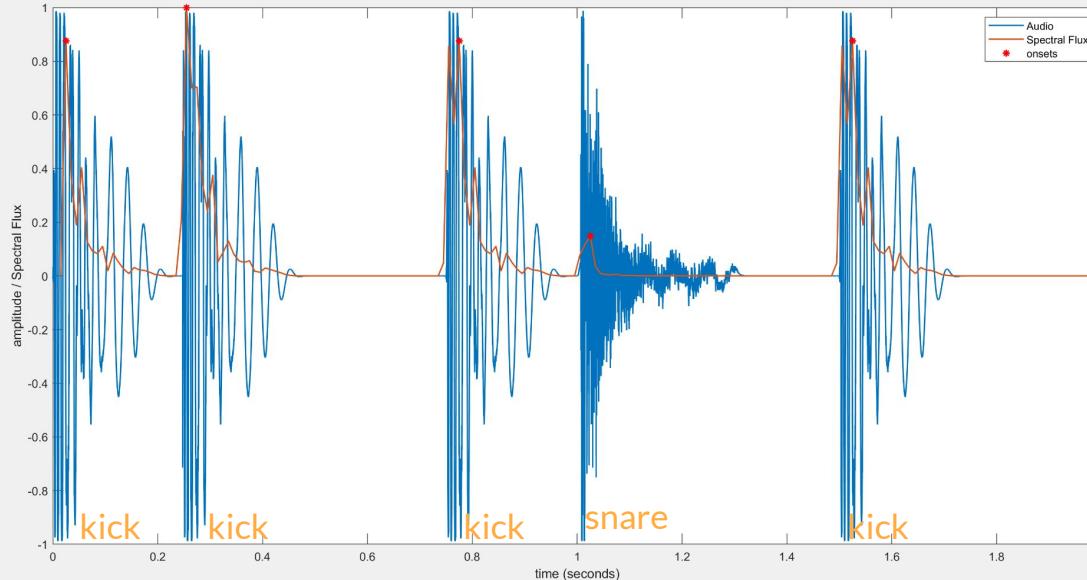
Let $k = 3$



Step 4: Connect Classification Back to Time Index Information



UNIVERSITY
of York





Chord Recognition

Tuning Systems

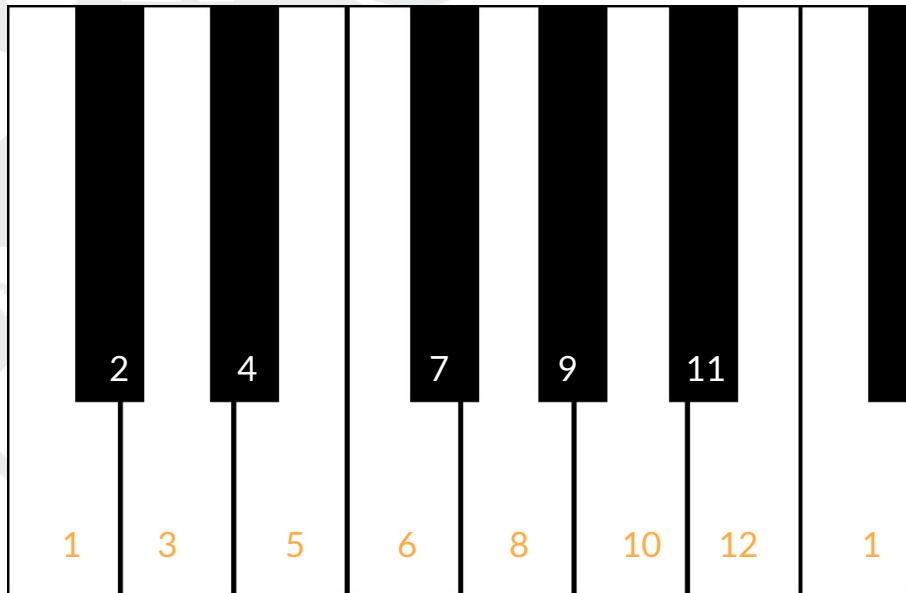
The features we are going to discuss this week make an assumption that we are dealing with instruments from *western* music - I.E European classical and baroque, jazz, rock, and pop

Other cultures use different tuning systems. For example, unique tuning systems evolved in other cultures and continents

It is important to remember this when we discuss the feature extraction methods discussed in this lecture. Applying these methods to tuning systems from other cultures probably won't work!!

12 Tone Tuning

Western Tuning systems are based on dividing the octave



Equal Tempered

$$F_{\text{pitch}}(p) = 2^{(p-69)/12} \cdot 440$$

Important things to remember:

Each note has an associated **centre frequency**. We can map notes to frequencies using the following formula:

Each note is further divided into **cents**, with 50 cents above and below each centre frequency. This will be important to remember later on when we discuss chromagrams.

Cents in Equal Temperament

B

C

C#

100 cents
between each
semitone

$$c = 1200 \log_2 \left[\frac{f_1}{f_2} \right]$$

F1 = note 1's frequency; F2 is note 2
fundamental frequency. C is cents

See appendix 2 of Howard and Angus's Acoustics and Psychoacoustics for a better explanation of this.



Pitch and Chroma Features

Chromagrams

Recall how earlier, how we calculated spectral flux by calculating the difference between frequency bins at time t and $t-1$ and summing these differences.

Each frequency bin can be mapped onto a specific frequency. We can calculate the centre frequency (of an equal tempered 12 tone tuning system!!!) of a note by using the following formula:

$$F_{\text{pitch}}(p) = 2^{(p-69)/12} \cdot 440$$

Chromagrams Cont.

```
function [freq] = ptoFreq(p)
%This function takes an input pitch number (between 0 and 127) and converts
%it to the centre frequency using the formula ( 2^( (p-69)/12 ) *440

freq = ( 2^( (p-69)/12 ) *440

end
>> ptoFreq(64)

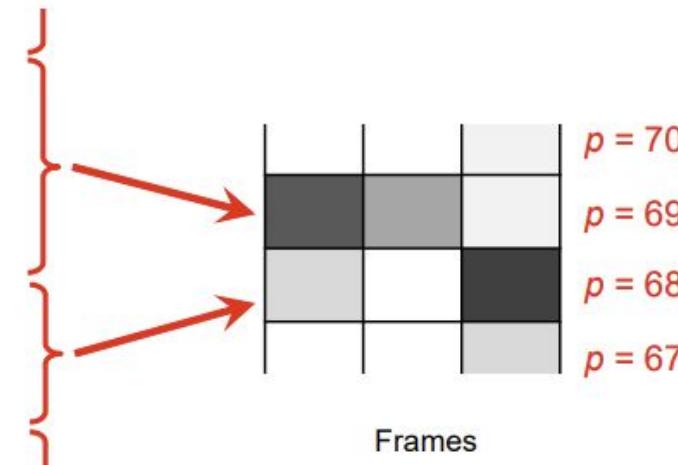
freq =
329.6276
```

Frequency Pooling

$F_{\text{coef}}(43) = 463.0$
 $F_{\text{coef}}(42) = 452.2$
 $F_{\text{coef}}(41) = 441.4$
 $F_{\text{coef}}(40) = 430.7$
 $F_{\text{coef}}(39) = 419.9$
 $F_{\text{coef}}(38) = 409.1$
 $F_{\text{coef}}(37) = 398.4$

Frames

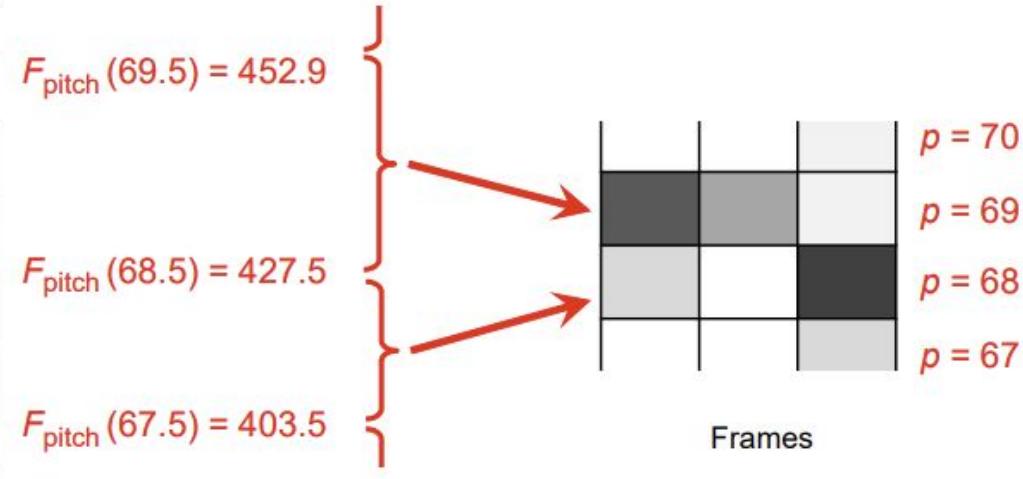
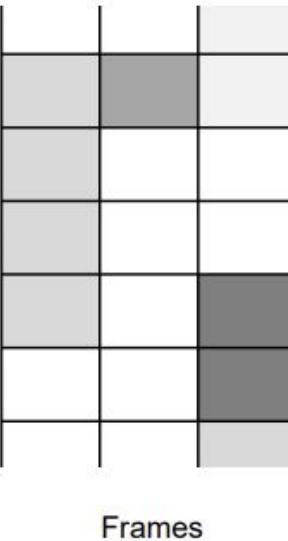
$F_{\text{pitch}}(69.5) = 452.9$
 $F_{\text{pitch}}(68.5) = 427.5$
 $F_{\text{pitch}}(67.5) = 403.5$



Frequency Pooling

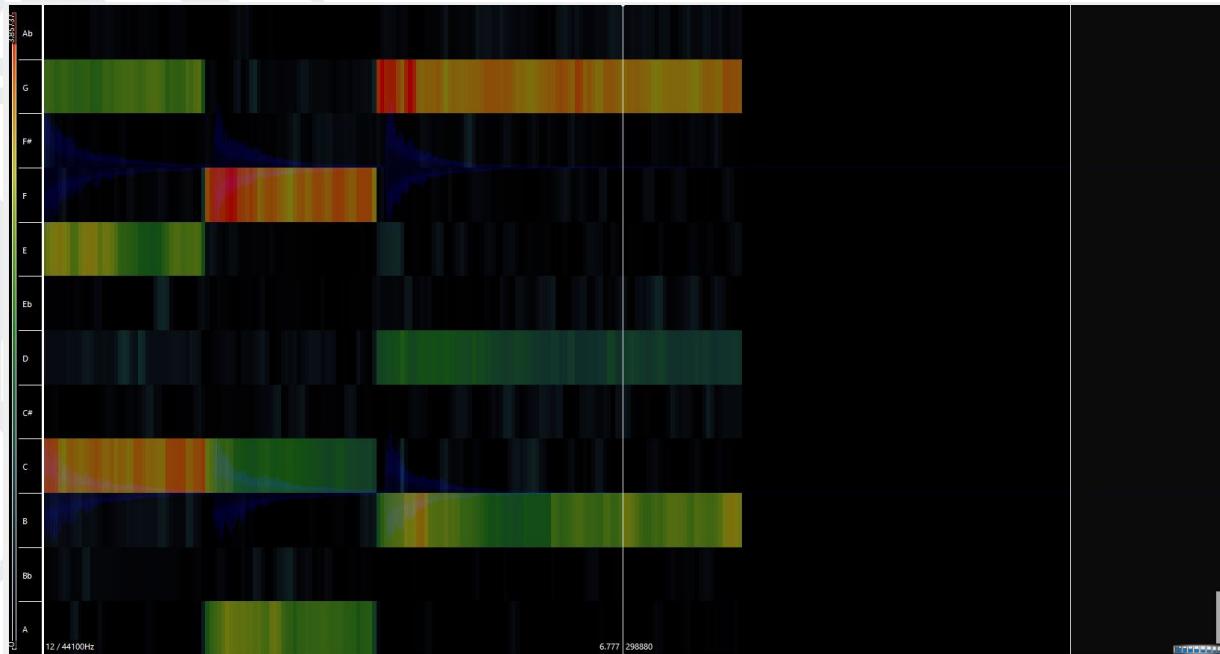


$F_{\text{coef}}(43) = 463.0$
 $F_{\text{coef}}(42) = 452.2$
 $F_{\text{coef}}(41) = 441.4$
 $F_{\text{coef}}(40) = 430.7$
 $F_{\text{coef}}(39) = 419.9$
 $F_{\text{coef}}(38) = 409.1$
 $F_{\text{coef}}(37) = 398.4$



$$F_{\text{pitch}}(p) = 2^{(p-69)/12} \cdot 440 \quad c = 1200 \log_2 \left[\frac{f_1}{f_2} \right]$$

Generating Chromagram





Chord Recognition

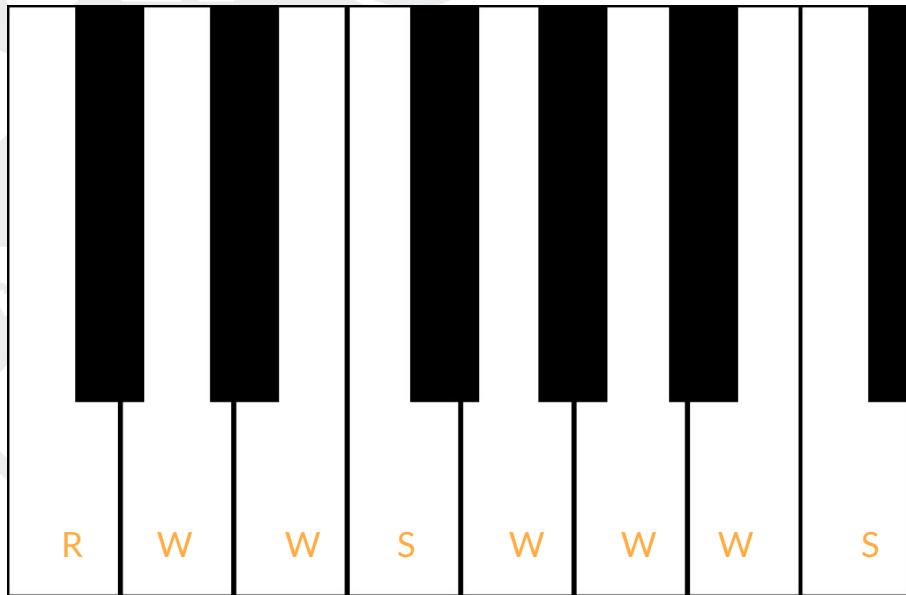
Formula for Major Scale

The major scale for any note on the chromatic scale can be constructed by following the following formula:

W, W, S, W, W, W, S

Where W corresponds to a whole tone (2 notes) and S corresponds to a semitone (1 note)

Formula for Major Scale



Basic Chord Theory



UNIVERSITY
of York

Consider the C Major Scale below:

A musical staff consisting of five horizontal lines. A treble clef is positioned at the top left. Below the staff, there are eight open circles, each representing a note. Below each note is a letter: C, D, E, F, G, A, B, and C. The notes are aligned such that the first note (C) is on the fourth line, the second (D) is on the third space, the third (E) is on the second line, the fourth (F) is on the first space, the fifth (G) is on the second line, the sixth (A) is on the third space, the seventh (B) is on the fourth line, and the eighth (C) is on the fifth line.



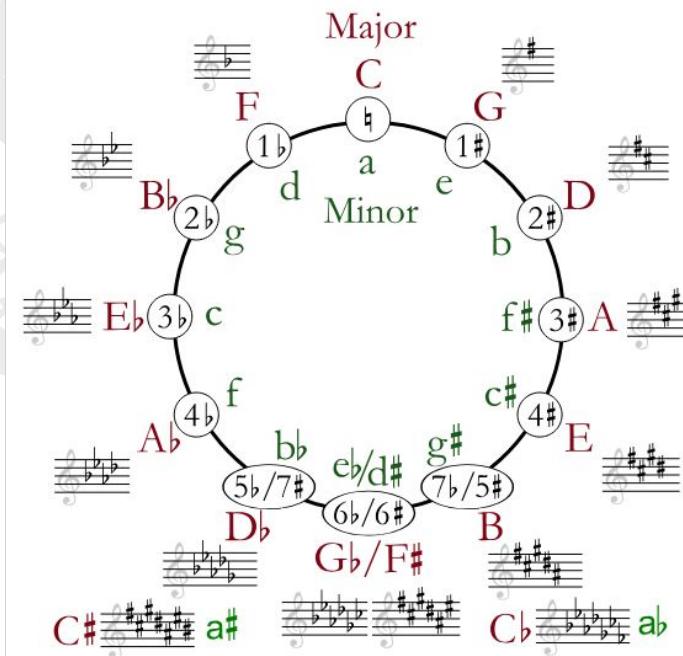
Basic Chord Theory

G	A	B	C	D	E	F
E	F	G	A	B	C	D
C	D	E	F	G	A	B
I	ii	iii	IV	V	vi	vii

Circle of Fifths



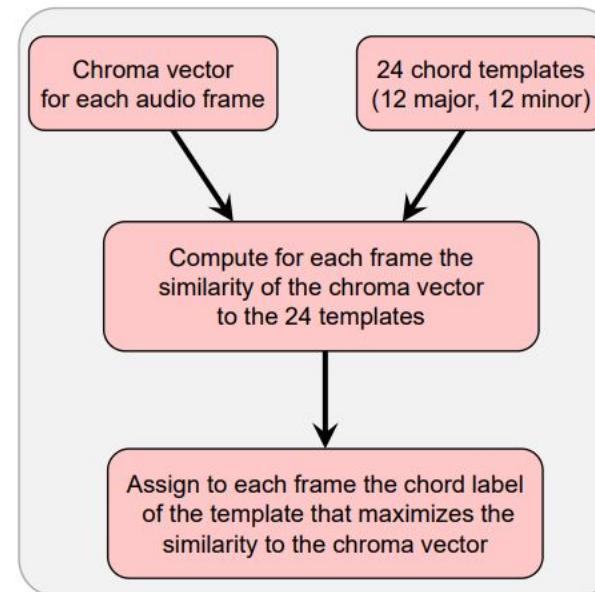
UNIVERSITY
of York



Template Based Chord Recognition

	C	C [#]	D	...	Cm	C [#] m	Dm	...
B	0	0	0	...	0	0	0	...
A [#]	0	0	0	...	0	0	0	...
A	0	0	1	...	0	0	1	...
G [#]	0	1	0	...	0	1	0	...
G	1	0	0	...	1	0	0	...
F [#]	0	0	1	...	0	0	0	...
F	0	1	0	...	0	0	1	...
E	1	0	0	...	0	1	0	...
D [#]	0	0	0	...	1	0	0	...
D	0	0	1	...	0	0	1	...
C [#]	0	1	0	...	0	1	0	...
C	1	0	0	...	1	0	0	...

Template Based Chord Recognition



Evaluating Chord Recognition Systems



Usually evaluated using a database that is assigned ground truth labels by professional musicians who have transcribed the chords in various pieces.

These ground truth labels can be combined with Precision, Recall, and F-measure metrics we discussed previously. .

Evaluating Chord Recognition Systems



$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Fscore} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Challenges in Chord Recognition - Chord Ambiguities

G	G	E	B
E	Eb	C	G
C	C	A	E
C	Cm	Am	Em

Consider the chord C. Out of the 24 chords we generated for the template based approach, it shares 2 notes with the chords Am, Cm, Em. This could cause problems when carrying out similarity matching.

Challenges in Chord Recognition - Extended Chords

B		
G	G	B
E	E	G
C	C	E
Cmaj7	C	Em

We also encounter other problems with a template based approach when we introduce extended chords suchs as a maj7.

For example, the chord Cmaj7 also includes the chords C and Em

Addressing These Issues

There are lots of interesting ways of addressing these issues (hidden markov models, extra filtering, check out the Meinard Müller book where he discusses some of these).

However, neural networks and deep learning show a lot of promise, and have become a fixture in the field of chord recognition. Check out the paper below for further discussion.

20 YEARS OF AUTOMATIC CHORD RECOGNITION FROM AUDIO

Johan Pauwels¹

Ken O'Hanlon¹

Emilia Gómez²

Mark B. Sandler¹

¹ Centre for Digital Music, Queen Mary University of London

² Music Technology Group, Universitat Pompeu Fabra

{j.pauwels, k.o.ohanlon, mark.sandler}@qmul.ac.uk, emilia.gomez@upf.edu

Proceedings of 20th annual conference of the International Society for Music Information Retrieval (ISMIR) 2019



Dynamic Time Warping

Dynamic Time Warping (DTW) - Purpose

The DTW has two purposes:

- 1) Align two signals that are related in some way to each other.
- 2) Calculate the minimum distance between these two signals after aligning them.



Dynamic Time Warping

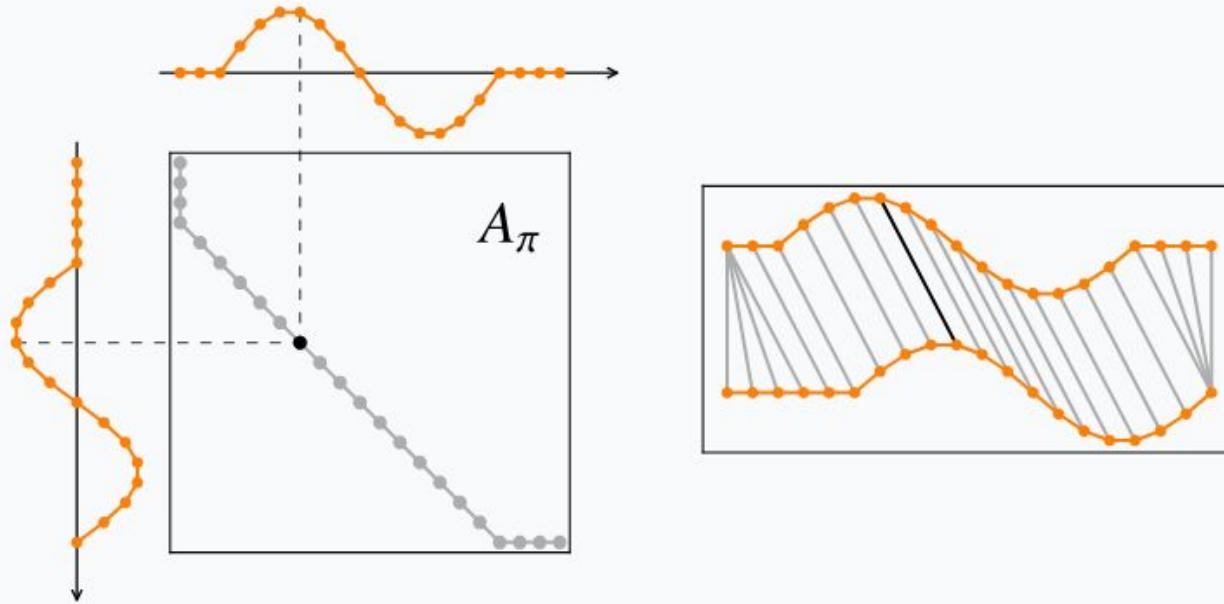


Figure from Romain Tavenard's excellent DTW Tutorial. Take a look at it!
<https://rtavenar.github.io/blog/dtw.html>

Dynamic Time Warping

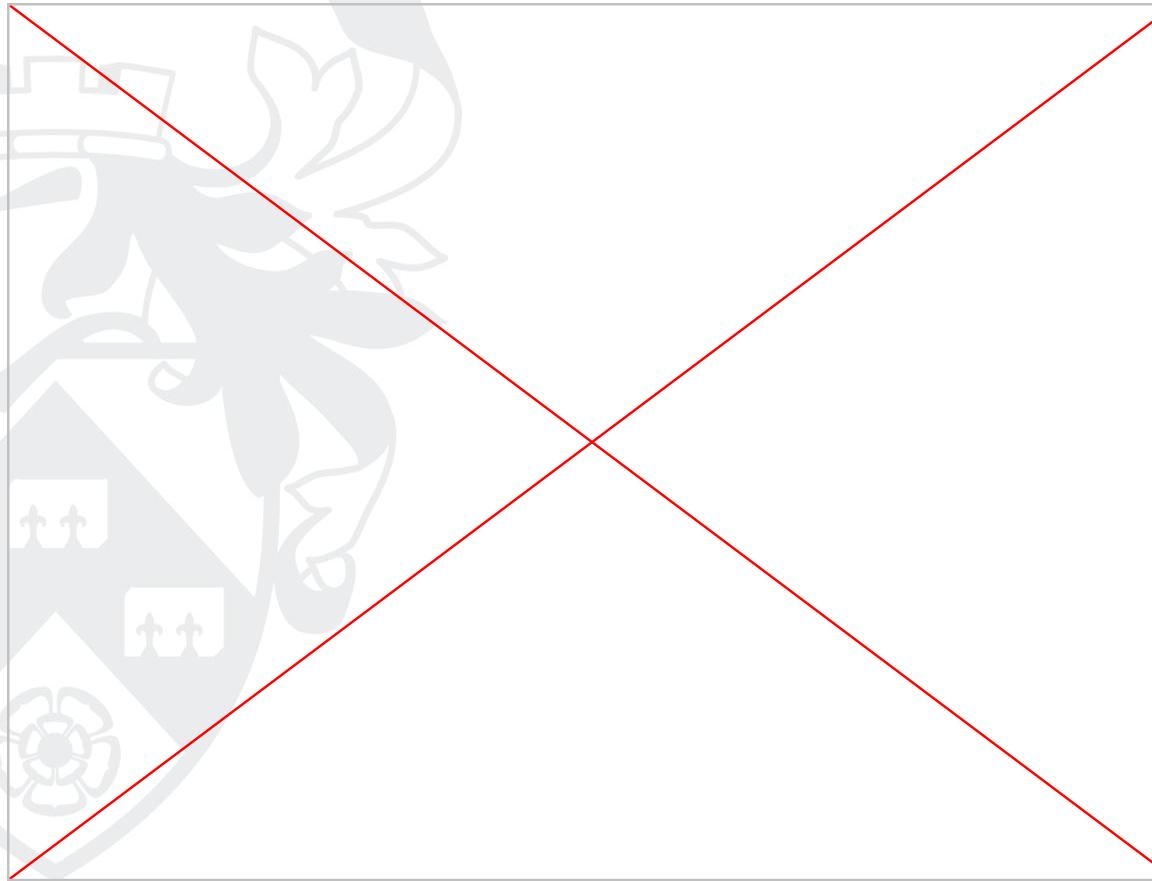


Figure from Romain Tavenard's excellent DTW Tutorial. Take a look at it! <https://rtavenar.github.io/blog/dtw.html>

Dynamic Time Warping

Try out Romain Tavenards DTW tutorial:

<https://rtavenar.github.io/blog/dtw.html>

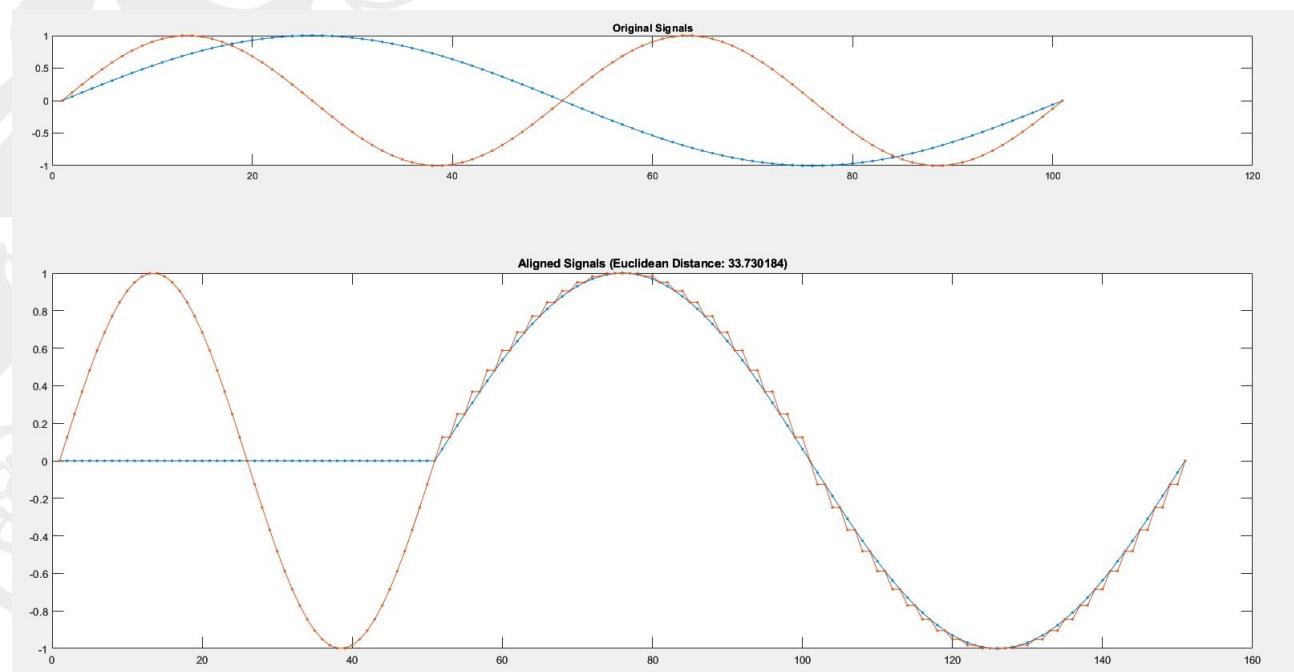
DTW in Matlab

```
1 fs= 100; %set sampling rate
2 t = 1/fs; %Set sampling interval
3 T = 0:t:1;%create time vector
4
5 f1 = 1; %frequency of our first sinusoid
6
7 % argument for angular frequency (remember, there are 2*pi radians in a
8 % full rotation)
9 w = 2*pi;
10
11 x = sin(w*f1*T) %create first sinusoid
12
13
14 f2 = 2; %frequency for our second sinusoid
15 y = sin(w*f2*T) %create second sinusoid
16
17 dtw(x,y) %carry out DTW on signals X and Y
18
```

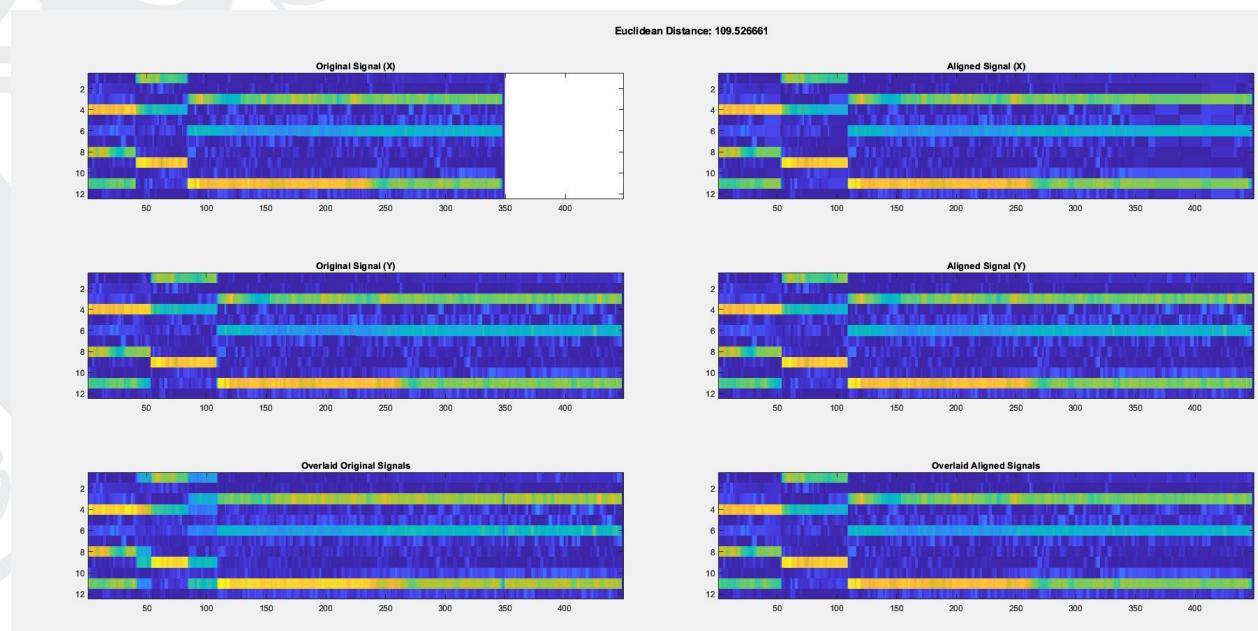


UNIVERSITY
of York

DTW in Matlab



Using the DTW with Chromagram



DTW in Matlab

```
[dist,ix,iy] = dtw(x,y) %carry out DTW on signals X and Y
```

If we look at the code above, you can see that we can calculate values called ix and iy. These are the indexes we can use to align our signals. IE, $x(ix)$ is the version of X with the lowest distance to $y(iy)$

For matrices (like our chromagrams) this would look like
 $x(:, ix)$

Summary

Western tuning systems divide the octave into 12 notes.

The most common tuning is equal tempered tuning.

Using our knowledge of tuning systems, we can convert the frequency bins of spectrogram into a chromagram.

Chromagrams are useful in chord recognition, but the use of template based recognition methods is being overtaken by deep learning models.

Dynamic Time Warping can be used to align and calculate the distance between two sets of features.



Thanks!

Any questions?

References

Howard, D., Angus, J., *Acoustics and Psychoacoustics*, 2016,
Routledge

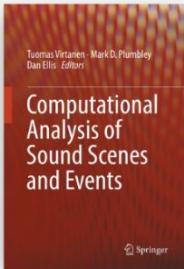
Müller, M., *Fundamentals of Using Python and Jupyter Notebooks
Second Edition Music Processing*, 2019 Springer

Romain Tavenard's excellent DTW Tutorial. Take a look at it!

<https://rtavenar.github.io/blog/dtw.html>



Recommended Reading



BOOK

Computational Analysis of Sound Scenes and Events Computer Appl. in Social and Behavioral Sciences ; Digital techniques ; Engineering Acoustics ; Signal, Image and Speech Processing ; User Interfaces and Human Computer Interaction

Virtanen, Tuomas ; Plumbley, Mark D ; Ellis, Dan Virtanen, Tuomas ; Ellis, Dan ; Plumbley, Mark D.

Cham: Springer International Publishing AG 2017

“ Presenting computational methods for extracting useful information from audio signals, this book collects the state of the art in the field of sound event and scene analysis...”

Full text available >

Chapters of this book (21) >

TOP

VIEW ONLINE

Acoustic Features for Environmental Sound Analysis

Romain Serizel, Victor Bisot, Slim Essid, Gaël Richard

Pages 71-101