ECS7006 Music Informatics

Week 6 – Harmony and Melody

School of Electronic Engineering and Computer Science
Queen Mary University of London

prepared by Simon Dixon
using material by Juan Bello, Emmanuel Vincent and Meinard Müller

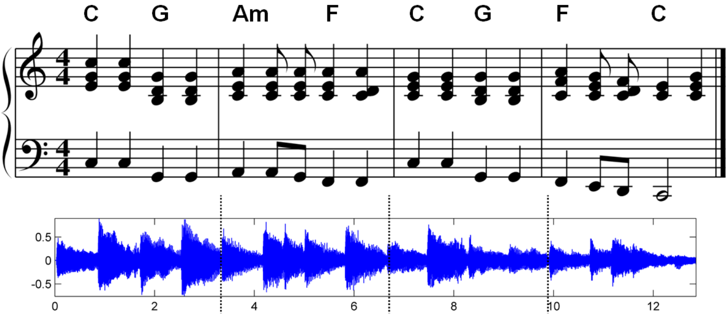
s.e.dixon@qmul.ac.uk

2023

Harmony: Chord and Key Estimation

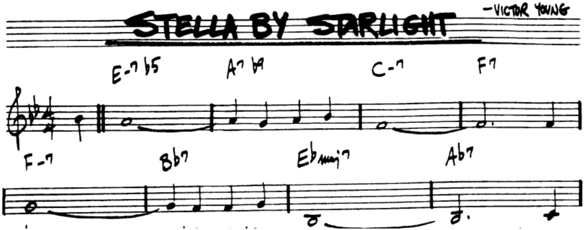
Harmony

- **Harmony**: refers to relationships involving simultaneous pitches (chords) and sequences of chords (and notes)
- **Chord**: two (usually three) or more notes played simultaneously
- **Key**: the overall tonal centre and mode
- Features that describe harmony are important for music retrieval tasks (e.g. cover song recognition) and for analysing musical structure



Chord Representations

- In many styles of Western music, a musical work can be summarised by a “partial score”, containing only an outline of the work, such as lists of symbols for the harmony, representing the progression and timing of chords



- Chord symbols can be represented in Roman numeral notation, figured bass (relative to key or bass), or pop/jazz notation (absolute)
- For any chord representation, we need to specify the temporal resolution (e.g. note, beat or bar level), the level of abstraction (e.g. function vs specific instance), and the chord vocabulary

Chord Symbols

- A chord symbol expresses the *root* pitch class, the basic *chord type* (e.g. major, minor, augmented), and any *extensions* (e.g. 7th, 9th), modifications (e.g. $\flat 5$) or extra notes (e.g. $\sharp 9$), and possibly voicing information (inversion, bass note e.g. G/F)
- There are many genre-specific assumptions about the interpretation of these symbols
- If pitch is expressed in terms of a pitch class and octave number (e.g. C4), one simple approach is to consider chords as sets of pitch classes
- Then each chord symbol can be mapped to a pitch class set, e.g.

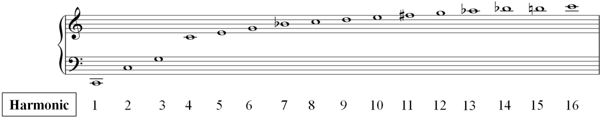
  C = C major $\rightarrow \{$C, E, G$\}$
  Gm7 = G minor 7th $\rightarrow \{$G, B$\flat$, D, F$\}$

Template Matching

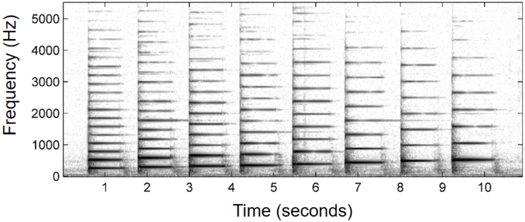
- A baseline chord transcription approach involves feature extraction followed by template matching
- Given an audio feature that tells the presence or absence of each pitch class at any point in time, we can match to the nearest chord symbol in our chord vocabulary
  - Frequency detection and mapping to pitch are hard (why?)
  - Mapping from pitch to pitch class is trivial
  - The goal is to capture pitch class and be invariant to instrument and timbre
  - The *chroma* feature (also *harmonic pitch class profile*) is widely used (and misused)
- Possible postprocessing steps include temporal smoothing and language modelling

Chroma Features

- Basic idea: assign energy in spectrum to pitch classes
- Naïve approach: assume that energy at any frequency f is due to a pitched sound with fundamental frequency f
- Consider the harmonic series to see one reason why this is wrong



- Other problems are spectral leakage and that percussive onsets have high energy across a broad range of frequencies



Chroma Features

- Start with a spectrogram $X(n,k)$, computed with window size N and hop size H from a signal with sampling rate f_s
- Then the log frequency spectrogram (Lecture 2) is given by:

$$Y(n,p) = \sum_{\{k:\ p-0.5\ \leq\ P(k)\ <\ p+0.5\}} |X(n,k)|^2$$

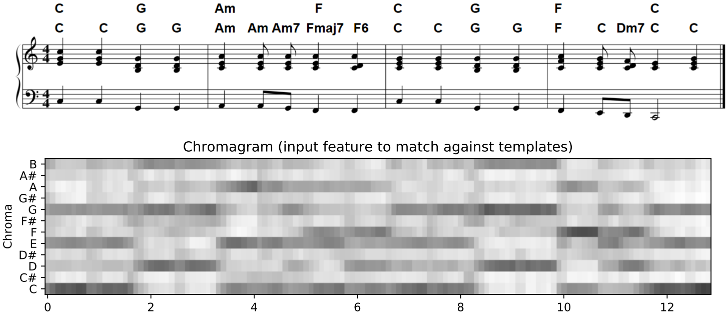
where

$$P(k) = 69 + 12\log_2 \frac{k f_s}{440N}$$

- Assume that the energy for each pitch p is solely at the corresponding fundamental frequency f_p
- Then the chroma representation sums across octaves the power at each f_p corresponding to pitch class $c = p \text{ mod } 12$:

$$C(n,c) = \sum_{\{p:\ p \text{ mod } 12\ =\ c\}} Y(n,p)$$

Chromagram Example

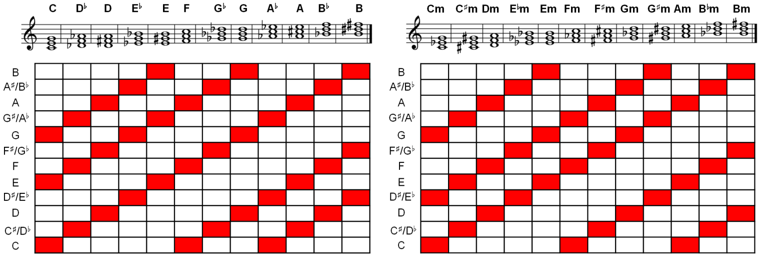


Chroma features: Pre- and post-processing

- Various other processing steps can be performed
- Log magnitude scaling: $C'(n,c) = \log(1 + \gamma C(n,c))$
- Normalisation: $C''(n,c) = \frac{C'(n,c)}{(\sum_{i=0}^{11} C'(n,i)^2)^{1/2}}$
  - Note that if the denominator is very small, it is best to replace C'' by a uniform vector of unit norm
- Temporal smoothing can be helpful, as chords change more slowly than notes
- Recalibration (tuning) should be performed if the reference frequency is not A4=440Hz
- Harmonic-percussive separation (Week 8) can be used to attenuate the non-pitch content before converting to a chroma representation
- An approximate transcription or pitch salience function can be used to assign partials to the correct pitch class

Template-Based Chord Transcription

- Now we can complete a basic chord transcription system
- For each chord symbol λ in the vocabulary Λ, compute a template representation $t_\lambda \in \mathbb{R}^{12}$
- One simple approach is to use binary vectors for t_λ with ones at the positions of pitch classes occurring in λ; e.g. major and minor triads:



	C	D♭	D	E♭	E	F	G♭	G	A♭	A	B♭	B
B					🟥			🟥				🟥
A♯/B♭				🟥			🟥				🟥	
A			🟥			🟥				🟥		
G♯/A♭		🟥			🟥				🟥			
G	🟥			🟥				🟥				
F♯/G♭			🟥				🟥					🟥
F		🟥				🟥					🟥	
E	🟥				🟥					🟥		
D♯/E♭				🟥					🟥			🟥
D			🟥					🟥			🟥	
C♯/D♭		🟥					🟥			🟥		
C	🟥					🟥			🟥			

	Cm	C♯m	Dm	E♭m	Em	Fm	F♯m	Gm	G♯m	Am	B♭m	Bm
B					🟥				🟥			🟥
A♯/B♭				🟥				🟥			🟥	
A			🟥				🟥			🟥		
G♯/A♭		🟥				🟥			🟥			
G	🟥				🟥			🟥				
F♯/G♭				🟥			🟥					🟥
F			🟥			🟥					🟥	
E		🟥			🟥					🟥		
D♯/E♭	🟥			🟥					🟥			
D			🟥					🟥				🟥
C♯/D♭		🟥					🟥				🟥	
C	🟥					🟥				🟥		

- Given a similarity measure $s : \mathbb{R}^{12} \times \mathbb{R}^{12} \to \mathbb{R}$, the detected chord for input chroma vector c_n is $\lambda_n = \arg\max_{\lambda\in\Lambda} s(t_\lambda, c_n)$

Template Matching: Example Output

- Using a normalised inner product as similarity measure:

$$s(x,y) = \frac{\langle x, y \rangle}{||x|| \cdot ||y||}$$

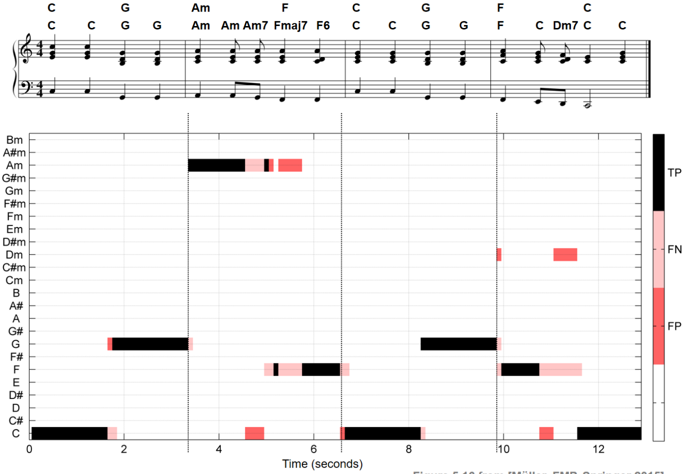
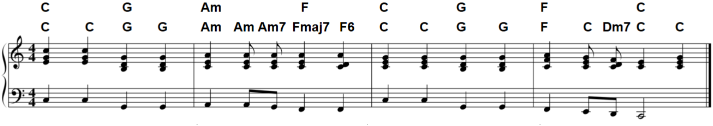


Figure 5.16 from [Müller, FMP, Springer 2015]

Key Detection

- Key detection can be seen as seeking a higher level of abstraction of the harmonic content compared to finding chord sequences
- A key is defined by a central pitch class and a mode (major, minor)
- e.g. the key is C major for this whole excerpt:



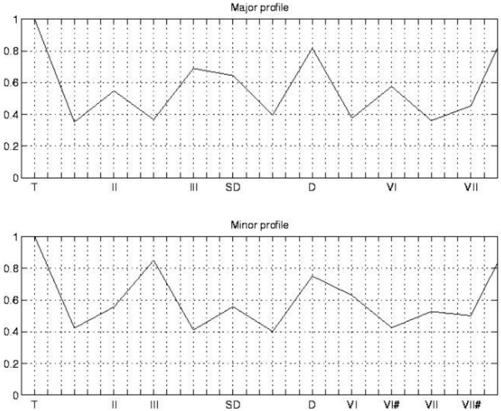
- Key can be difficult to define (ambiguity, temporal scale)
- Often the first and last chords of a piece indicate the key
- Chords are good indicators: a chord “belongs” to a key if all of its pitch classes are in the scale of the key
    - Based on this definition, many chords belong to multiple keys

Probe Tone Profiles for Key Detection

- A simple template-based approach can be defined based on Krumhansl's (1990) probe tone profiles
- Probe-tone method: first used by Krumhansl and Shepard (1979) for quantifying listeners' responses to tonality
- Participants listen to a short sequence of notes, chords, scale or melody (the *context*) and then rate a following tone, the *probe tone*, for degree of completion or goodness-of-fit with the context
- The set of ratings, one for each of the 12 possible probe tones, is called the *probe-tone profile* for the context, and reflects the stability of tones in the context
- If the context defines a key, the profile reflects the relationship between each probe tone and the key

Key Finding with Probe Tone Profiles

- For MIDI data, the key can be estimated by correlating the probe tone profiles (below) with a vector of the relative duration of each pitch class (Gómez and Herrera, 2004)
- For audio data, chroma features can be used instead of durations

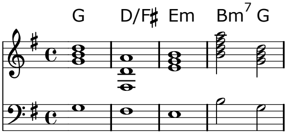


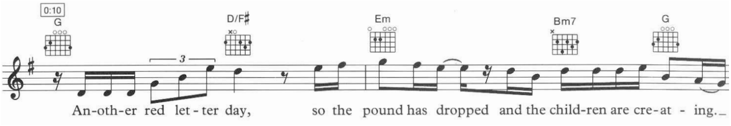
A Probabilistic Model for Chord Transcription

- Motivation: intelligent chord transcription
  - Modern popular music
- Front end (low-level) processing
  - Approximate transcription (Mauch & Dixon ISMIR 2010)
- Dynamic Bayesian network (IEEE TASLP 2010)
  - Integrates musical context (key, metrical position) into estimation
- Utilising musical structure (ISMIR 2009)
  - Clues from repetition
- Full details in Matthias Mauch's PhD thesis (2010): *Automatic Chord Transcription from Audio Using Computational Models of Musical Context*

The Problem: Chord Transcription

- Different to polyphonic note transcription
- Abstractions
  - Pitch height is disregarded (except for bass notes)
  - Notes are integrated across time
  - Non-harmony notes are disregarded
- Aim: output suitable for musicians




0:10
G
D/F♯
Em
Bm7
G
3
An-oth-er red let - ter day, so the pound has dropped and the child-ren are cre-at - ing._

Signal Processing Front End

- Preprocessing steps
  - Find reference tuning pitch
  - Perform noise reduction and normalisation
  - Beat tracking for beat-synchronous features
- Usual approach: chromagram
  - FFT frequency bins mapped to pitch classes (A,B♭,B,...)
  - Advantage: data reduction (one 12-D feature per time frame)
  - Disadvantage: frequency $\neq$ pitch
- Approximate transcription with non-negative least squares (NNLS)
  - Consider spectrum X as a weighted sum of note profiles
  - Dictionary T: fixed spectral shape for all notes
  - Find note activation pattern z:
  - $X \approx Tz$ (similar to NMF: $V \approx WH$)
  - NNLS: minimise $||X - Tz||$ for $z \geq 0$

Signal Processing Front End

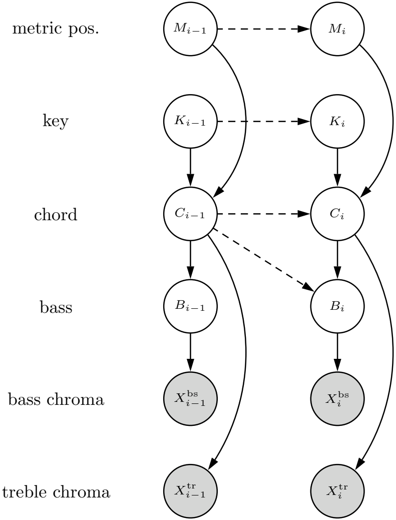
- Preprocessing steps
  - Find reference tuning pitch
  - Perform noise reduction and normalisation
  - Beat tracking for beat-synchronous features
- Usual approach: chromagram
  - FFT frequency bins mapped to pitch classes (A,B♭,B,...)
  - Advantage: data reduction (one 12-D feature per time frame)
  - Disadvantage: frequency $\neq$ pitch
- Approximate transcription with non-negative least squares (NNLS)
  - Consider spectrum X as a weighted sum of note profiles
  - Dictionary T: fixed spectral shape for all notes
  - Find note activation pattern z:
  - $X \approx Tz$ (similar to NMF: $V \approx WH$)
  - NNLS: minimise $||X - Tz||$ for $z \geq 0$

Signal Processing Front End

- Preprocessing steps
  - Find reference tuning pitch
  - Perform noise reduction and normalisation
  - Beat tracking for beat-synchronous features
- Usual approach: chromagram
  - FFT frequency bins mapped to pitch classes (A,B♭,B,...)
  - Advantage: data reduction (one 12-D feature per time frame)
  - Disadvantage: frequency $\neq$ pitch
- Approximate transcription with non-negative least squares (NNLS)
  - Consider spectrum X as a weighted sum of note profiles
  - Dictionary T: fixed spectral shape for all notes
  - Find note activation pattern z:
  - $X \approx Tz$ (similar to NMF: $V \approx WH$)
  - NNLS: minimise $||X - Tz||$ for $z \geq 0$

Musical Context in a Dynamic Bayesian Network (DBN)

- Key, chord, metrical position and bass note are estimated simultaneously
  - Chords are estimated in context
  - Useful details for *lead sheets*
- Graphical model with two temporal slices: initial and recursive slice
  - Nodes represent random variables
  - Directed edges represent dependencies
  - Observed nodes are shaded



DBN Model Details

- Node dependencies derived from an “expert” model (musical knowledge)
- Metric position: cyclic pattern 1, 2, 3, 4, 1, 2, 3, 4, ...
- Key: stable (low probability of change)
- Chord: fits key and changes at strong metrical positions
- Bass note: fits chord (particularly on first beat)
- Observations: bass and treble chroma emission model based on bass and chord pitch classes respectively

Modelling Musical Structure

- Popular music is repetitive
    - High-level segments such as the verse and chorus occur multiple times in a song, usually with no change in harmony
- Markovian assumption of DBNs disallows modelling of long-term dependencies (global structure)
- Idea: recognise the high-level structure and combine the low-level features of repeated sections
    - Attenuate non-systematic deviations and noise
    - Provide consistent (and shorter) transcriptions
- Segments are found with a greedy algorithm searching for the best “diagonal lines” in the self-similarity matrix (see Week 8)
- Chroma vectors are averaged for corresponding beats of each instance of a segment
- **Results**: Improves chord recognition significantly
- Using manual (ground-truth) segmentation and/or beats does not improve results significantly

Evaluation

Methodology

- Tested against human annotations of chords
- Relative correct overlap (MIREX[a])

$$\text{RCO} = \frac{\text{Sum of durations of correct chords}}{\text{Total duration}}$$

- Varied number of chord categories
  - `maj, min` (MIREX)
  - `maj, min, maj/3, maj/5, maj6, 7, maj7, m7, dim, aug, NC`

[a] `https://www.music-ir.org/mirex/wiki/2019:Main_Page`

Data

- 174 Beatles songs, 18 Queen songs, 18 Zweieck songs
- Same data as used in MIREX evaluation
- Annotations freely available

Evaluation

Methodology

- Tested against human annotations of chords
- Relative correct overlap (MIREX[a])

$$\text{RCO} = \frac{\text{Sum of durations of correct chords}}{\text{Total duration}}$$

- Varied number of chord categories
  - `maj, min` (MIREX)
  - `maj, min, maj/3, maj/5, maj6, 7, maj7, m7, dim, aug, NC`

[a] `https://www.music-ir.org/mirex/wiki/2019:Main_Page`

Data

- 174 Beatles songs, 18 Queen songs, 18 Zweieck songs
- Same data as used in MIREX evaluation
- Annotations freely available

Results

MIREX-style evaluation results

Model	RCO
Plain	65.5
Add metric position	65.9
Best MIREX'09 (pretrained)	71.0
Add bass note	72.0
Add key	73.0
Best MIREX'09 (test-train)	74.2
Add structure	75.2
Use NNLS front end	80.7

Conclusions

- Modelling musical context and structure **is** beneficial
- Further work: separation of high-level (note-given-chord) and low-level (features-given-notes) models

Results

MIREX-style evaluation results

Model	RCO
Plain	65.5
Add metric position	65.9
Best MIREX’09 (pretrained)	71.0
Add bass note	72.0
Add key	73.0
Best MIREX’09 (test-train)	74.2
Add structure	75.2
Use NNLS front end	80.7

Conclusions

- Modelling musical context and structure **is** beneficial
- Further work: separation of high-level (note-given-chord) and low-level (features-given-notes) models

Predominant Melody Estimation

Melody

- A sequence of pitches
  - when notes are structured in succession so as to make a unified and coherent whole (“horizontal” organisation)
- The “tune” of the piece of music
  - the part of a polyphonic piece that you would hum or sing if asked to reproduce the piece
  - the most *salient* note sequence
- Melody perception is based on the *intervals* between successive notes (relative pitch) rather than their absolute pitches
- Includes the (relative) timing
- Why “predominant” melody? In MIR, this just reminds the reader that we are dealing with a complex mixture, not just a monophonic melody

Applications of Melody Estimation

- Retrieval: query by humming (QBH), or query by example (QBE)
- Cover song identification
- Transcription (e.g. of solos in jazz and oral traditions)
- Studying musical similarity, influence, style, intonation
- With separation: karaoke, music-minus-one

Predominant Melody Estimation

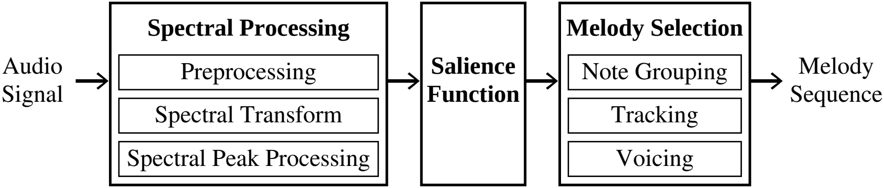
- The task can be viewed as a transcription problem, and broken into two steps
  - Polyphonic transcription or estimation of pitch salience
  - Melody note selection
- Various cues can be used to select melody notes
  - Dynamics: the main melody tends to be louder than accompaniment
  - Pitch: melody notes tend to have higher pitch than accompaniment
  - Timbre: all the melody notes usually come from the same instrument (sometimes assumed to be the human voice)
  - Continuity in the time-frequency plane: melody notes tend to start near the time and pitch of the end of the previous melody note

Predominant Melody: Baseline Approaches

- Monophonic transcription: a system designed for monophonic transcription might in fact return the most salient notes if given a polyphonic mixture as input
- Polyphonic transcription with *skylining*: this approach outputs the highest pitch detected at each timepoint by a polyphonic AMT system
- These approaches fail for several reasons:
    - they do not address the *voicing* problem: determining whether the melody is active or not at each point in time
    - they are based on invalid assumptions: that every melody note is higher in pitch or more salient than simultaneous accompaniment notes
    - they do not consider continuity or timbre/instrumentation

Predominant Melody: Salience-Based Approaches

- These approaches extend the monophonic transcription approach with a salience function that is appropriate for mixtures, combined with sequential constraints



Ref: Salamon et al., IEEE Signal Processing Magazine, 2014

Predominant Melody: Salience-Based Approaches

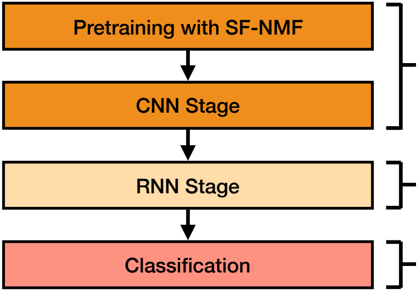
- Preprocessing: filtering (emphasise frequencies of interest; equalise loudness; separation e.g. of harmonic and percussive content)
- Spectral transform: STFT, CQT or multi-resolution FFT
- Spectral processing: normalisation, peak-picking, reassignment, instantaneous frequency
- Salience function: weighted harmonic summation
- Reduce false detections at multiples and submultiples of the true f_0: spectral smoothness (Klapuri 2004), detection of duplicate contours (Paiva 2006), penalising large jumps
- Pitch tracking: hidden Markov models (HMMs), dynamic programming, agent-based approaches, clustering, successive deletion
- Voicing: salience, timbre modelling, silence modelling

Predominant Melody: Other Approaches

- Separation-based approaches attempt to split the signal first into lead voice and accompaniment, e.g. using:
    - Source-filter model for lead instrument (could be instrument-specific)
    - Harmonic-percussive separation (sometimes with a longer window, to remove long notes or chords that occur in the harmony part)
    - Repeated structure estimation (assuming the melody varies over repeated structural units, where accompaniment does not vary)
- Classification-based approaches learn a function from data that estimates the melody pitch directly from the spectrum
    - Using traditional machine learning (e.g. support vector machines (SVMs) and HMMs, Poliner and Ellis, ISMIR 2005)
    - Using deep learning (Başaran et al., ISMIR 2018)

Başaran et al.: CRNN with SF-NMF pretraining

- A neural network is trained to recognise main melody notes



Pretraining with SF-NMF

CNN Stage

RNN Stage

Classification

Salience representation

Initial salience $\mathbf{H}^{F_0}$ + Deep CNN for enhanced salience

Temporal pitch tracking

Bidirectional GRU

Melody/non-melody F_0 estimation

62 classes: 1 non-melody, 61 target F_0s

- Results: state of the art on Medley-DB dataset
- Some missed & extra notes, octave & semitone errors

Resources

- M. Müller, *Fundamentals of Music Processing*, Section 3.1 and Chapter 5
- M. Mauch, *Automatic Chord Transcription from Audio Using Computational Models of Musical Context*, PhD Thesis, QMUL, 2010
- J. Salamon and E. Gómez, *Melody Extraction from Polyphonic Music Signals Using Pitch Contour Characteristics*, IEEE Transactions on Audio, Speech and Language Processing, vol. 20, no. 6, pp. 1759–1770, 2012
- J. Salamon, E. Gómez, D. Ellis and G. Richard, *Melody Extraction from Polyphonic Music Signals: Approaches, Applications and Challenges*, IEEE Signal Processing Magazine, vol. 31, no. 2, pp. 118–134, 2014