ECS7006 Music Informatics

Week 5 – Pitch Estimation and Transcription

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

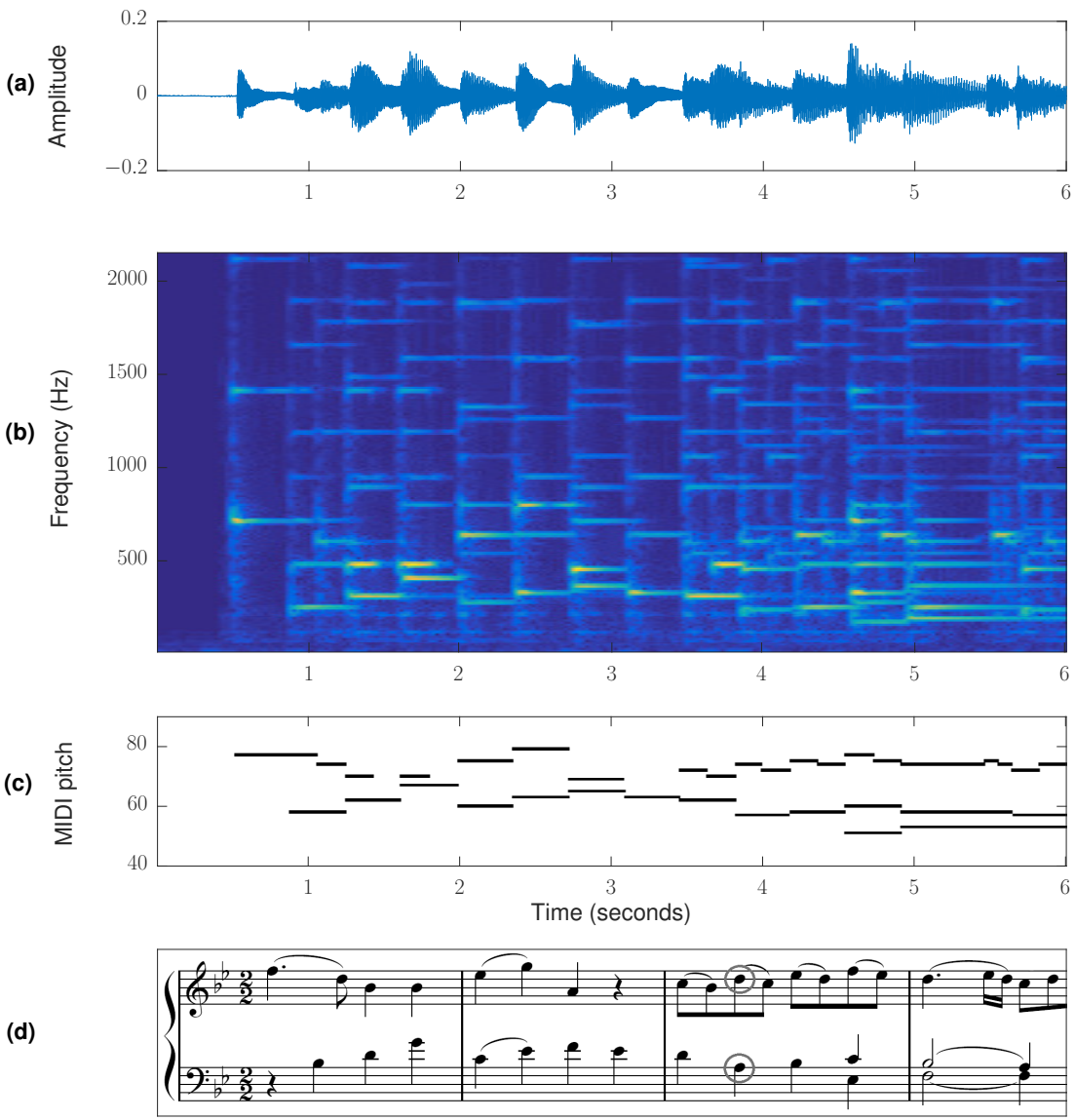
prepared by Simon Dixon
using material by Juan Bello, Emmanuel Vincent and Meinard Müller

s.e.dixon@qmul.ac.uk

2023

Queen Mary University of London

Automatic Music Transcription (AMT): Example



Input: audio signal

Time-frequency representation

Piano-roll representation

Output: musical score

Automatic Music Transcription (AMT)

- Aim: describe music signals at the note level, in terms of:
  - discrete pitch
  - onset time
  - duration
  - possibly other parameters (instrument, dynamics, expression, etc.)
- Much work on transcription (including this lecture) is limited to multiple pitch (or *multi-* f_0) detection
- A full transcription system would also need to include:
  - recognition of instruments
  - transcription of lyrics
  - rhythmic parsing
  - key estimation and pitch spelling
  - structure recognition (i.e. repetition)
  - layout of notation

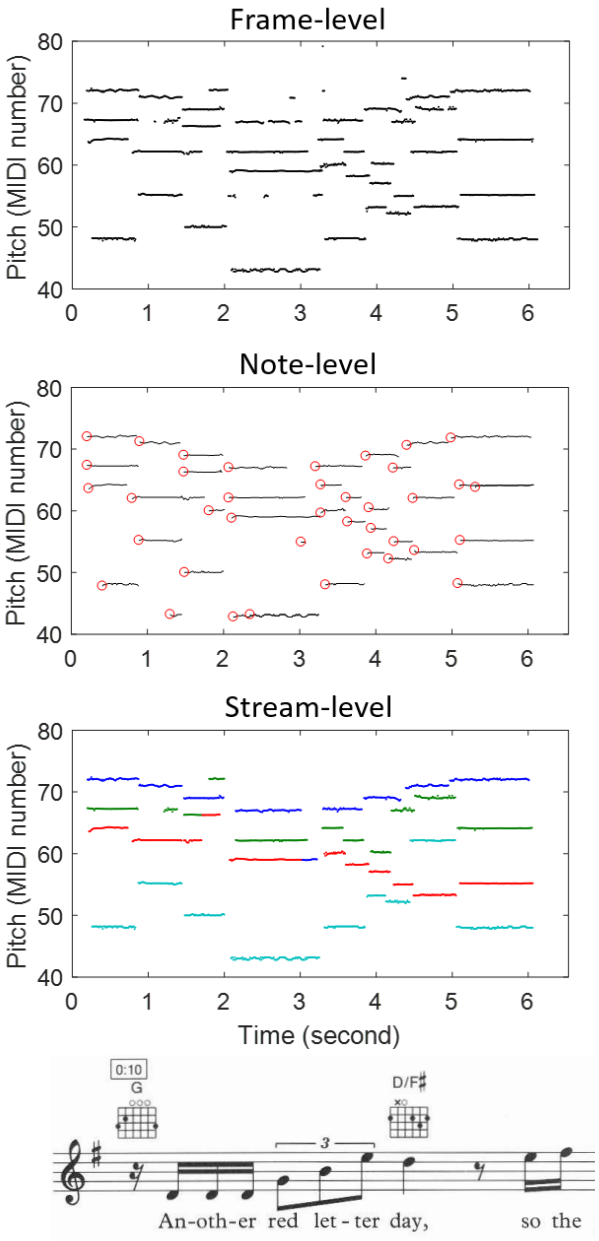
Applications of AMT

- Score creation from recordings or performances
- Music education / pedagogy
- Performance research
- Ethnomusicology, jazz studies
- Automatic accompaniment
- Indexing of music databases for search/retrieval
- Query-by-humming / Query-by-example
- Low bitrate compression (object coding, e.g. MPEG4)
- Audio editing, source separation, denoising, etc.

Variants of AMT

- Which approach is best depends on the data and use case
- Transcription difficulty increases with the number of simultaneous notes and instruments
  - The simplest case is transcription of *monophonic* music, where one instrument plays one note at a time
  - A more difficult case is transcription of *polyphonic* music, where multiple simultaneous notes are played by one (*monotimbral*) or more (*polytimbral*) types of instruments
- In some use cases, only a subset of notes needs to be transcribed:
  - in *predominant melody estimation* (likewise *bass-line estimation*) a monophonic line is transcribed from a polyphonic mixture
  - *chord transcription* summarises the harmony with chord symbols
  - *lyrics transcription* transcribes the words sung by the singer (like speech recognition)
  - a *lead sheet* usually contains melody, chords, and possibly lyrics

AMT Systems: Level of Detail



- *Frame-level transcription* provides only an estimate of which pitches are present at each time-point
- *Note-level transcription* groups the frame-level estimates into continuous pitch tracks or note objects
- *Stream-level transcription* also assigns each note object to an instrumental source or voice
- *Score-level transcription* interprets pitches and timing in musical units and symbols for human use

Background

Pitch and Melody

- **Pitch**: the perceived (fundamental) frequency f_0 of a musical note
  - related to the frequency spacing of a harmonic series in the frequency-domain representation of the signal
  - perceived logarithmically
  - one octave corresponds to a doubling of f_0
  - octaves are divided into 12 semitones
  - semitones are divided into 100 cents
  - standard reference: f_0 of A4 is 440 Hz
- **Melody**: a sequence of pitches, the "tune" of the piece of music
  - when notes are structured in succession so as to make a unified and coherent whole ("horizontal" organisation)
  - melody can be perceived without knowing the actual notes involved, based on the *intervals* between successive notes
  - melody is translation (transposition) invariant (in log domain)

Harmony

- **Harmony**: refers to relationships between simultaneous pitches (chords), and to sequences of chords ("vertical" organisation)
- Harmony is also perceived relatively (i.e. as intervals)
- Common intervals in Western music:
  - octave (12 semitones, f_0 ratio of 2)
  - perfect fifth (7 semitones, f_0 ratio approximately $\frac{3}{2}$)
  - major third (4 semitones, f_0 ratio approximately $\frac{5}{4}$)
  - minor third (3 semitones, f_0 ratio approximately $\frac{6}{5}$)
- **Consonance**: corresponds to simple fundamental frequency ratios

$$\frac{f_A}{f_B} = \frac{m}{n}$$

where m and n are small positive integers

  - Every nth partial of sound A overlaps every mth partial of sound B
  - What impact does this have on transcription?

Pitch and Harmonicity

- The pitch range of a standard piano is from A0 (f_0 = 27.5 Hz, MIDI note 21) to C8 (f_0 = 4186 Hz, MIDI note 108)
- Non-percussive instruments usually produce notes with *harmonic* sinusoidal partials, i.e. with frequencies:

$$f_k = k f_0$$

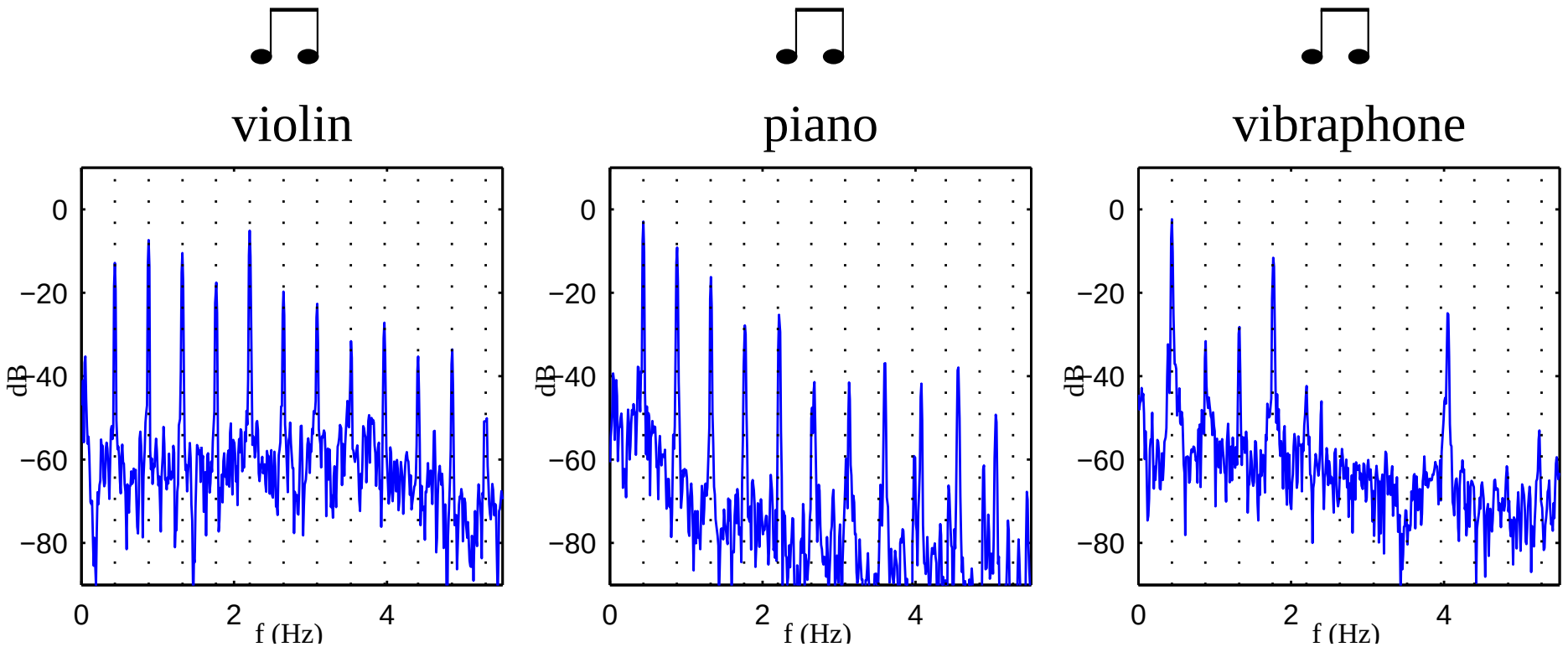
  where $k \geq 1$ and f_0 is the *fundamental frequency*
- Partials produced by struck or plucked string instruments are slightly *inharmonic*:

$$f_k = k f_0 \sqrt{1 + Bk^2}, \quad \text{with } B = \frac{\pi^3 E d^4}{64 T L^2}$$

  for a string with Young's modulus E (inverse elasticity), diameter d, tension T and length L

Harmonicity

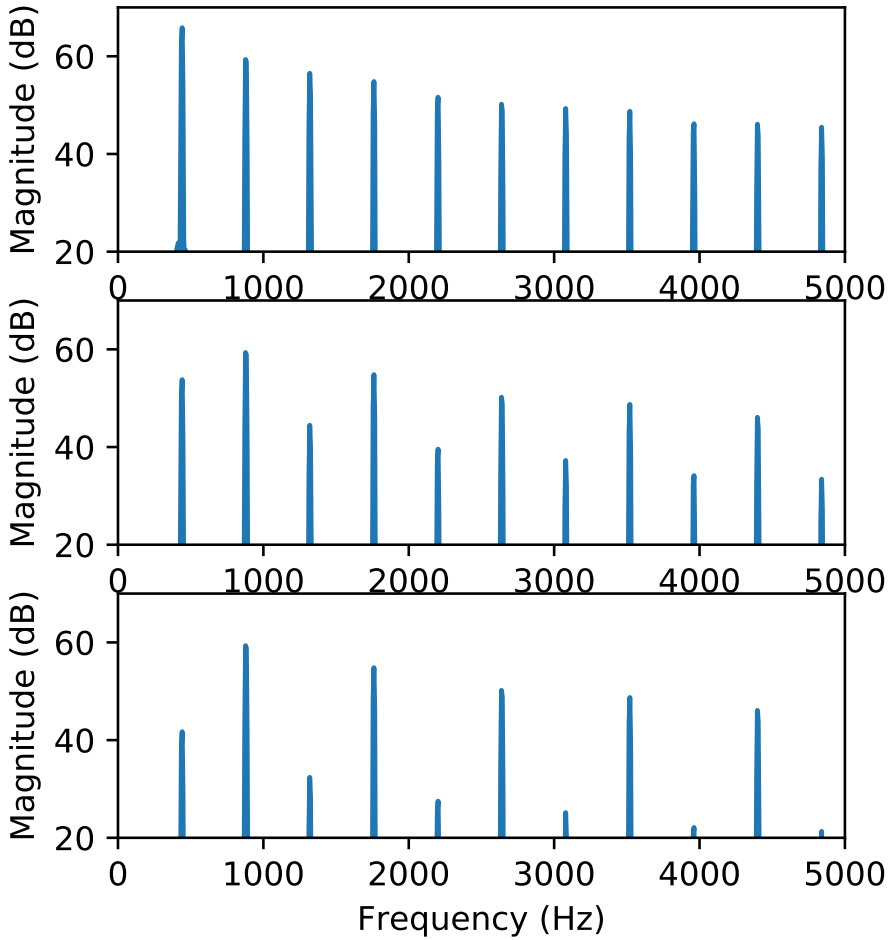
- Magnitude spectra for 3 acoustic instruments playing the note A4 ($f_0 = 440$ Hz)



- Note: the frequency axis should be in kHz

Pitch Ambiguity

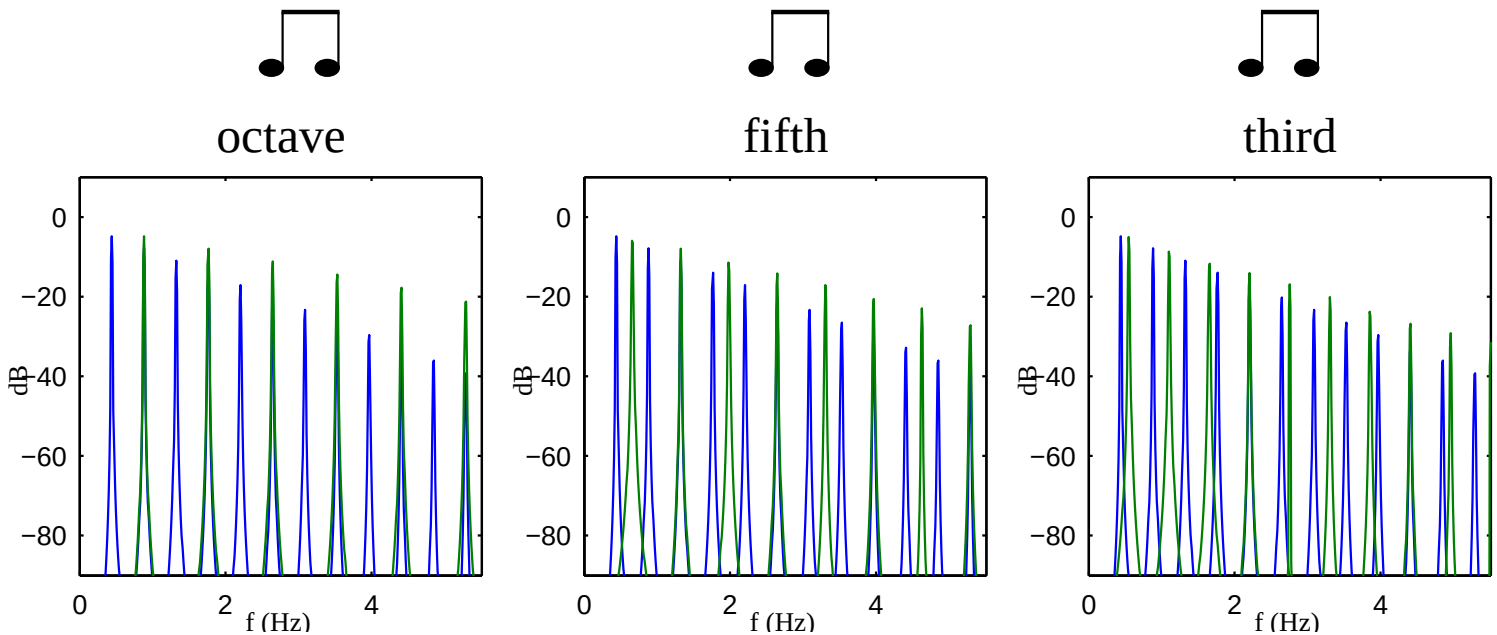
- Periodicity information alone does not suffice to determine pitch
- Harmonic partials with frequencies kf_0 for integer k can also be considered as the even-numbered partials of a harmonic complex with fundamental frequency $\frac{f_0}{2}$ since $kf_0 = (2k)\frac{f_0}{2}$ (called an *octave error*) 



Overlapping Partials

- Several notes having simple rational fundamental frequency ratios may be mistaken for a single note
- If two notes have fundamental frequencies f_1 and $f_2 = \frac{p}{q} f_1$, where p and q are small integers, every kq-th partial of the second note overlaps with the kp-th partial of the first note:

$$kqf_2 = kq\frac{p}{q}f_1 = kpf_1$$

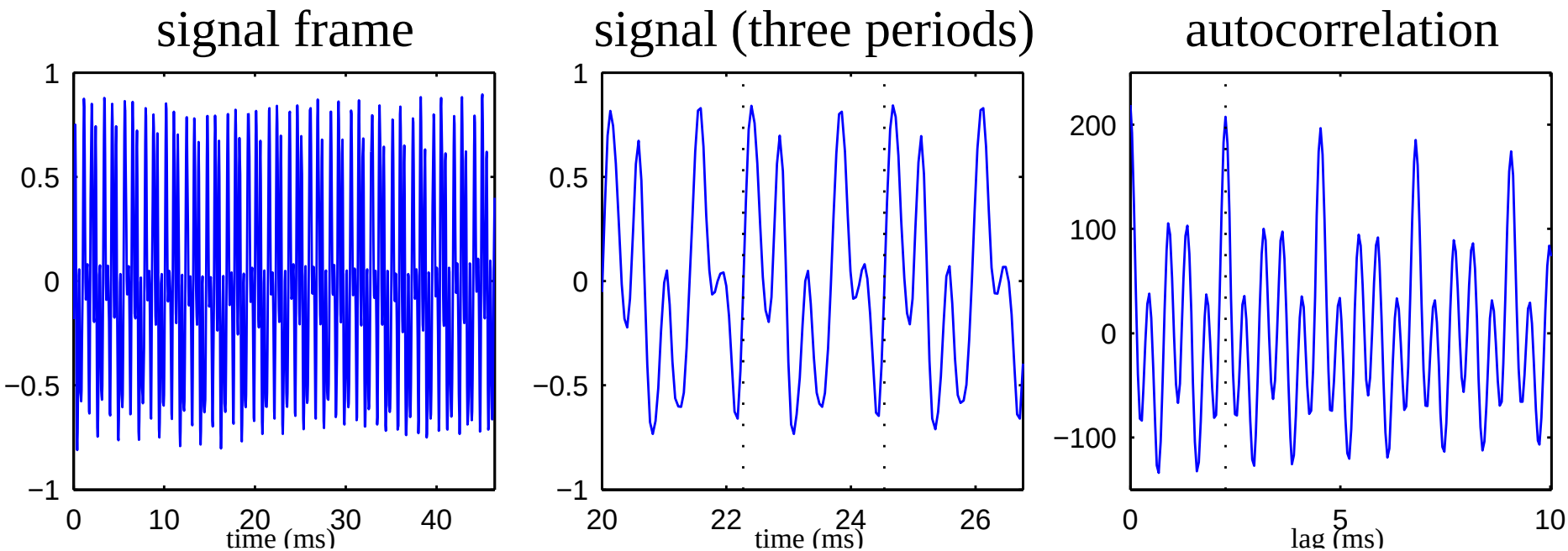


Periodicity-Based Pitch Estimation

Autocorrelation

- For monophonic signals, pitch can be estimated from periodicity in the signal
- The *Auto-Correlation Function* (ACF) of a signal frame $x(t)$ is

$$r(\tau) = \frac{1}{T} \sum_{t=0}^{T-\tau-1} x(t)x(t+\tau)$$



Autocorrelation

- Generally, for a monophonic signal, the highest peak of the ACF for positive lags τ corresponds to the fundamental period $\tau_0 = \frac{1}{f_0}$
- However other peaks always appear:
  - peaks of similar amplitude at integer multiples of the fundamental period (subharmonics)
  - peaks of lower amplitude at simple rational multiples of the fundamental period
- The autocorrelation is biased by the number of values in the sum; an unbiased estimate is obtained by replacing $\frac{1}{T}$ by $\frac{1}{T-\tau}$

YIN

- The ACF decreases for large values of τ, leading to harmonic errors when the target period τ_0 is not much smaller than frame length T
- An alternative approach called YIN is to consider the difference function:

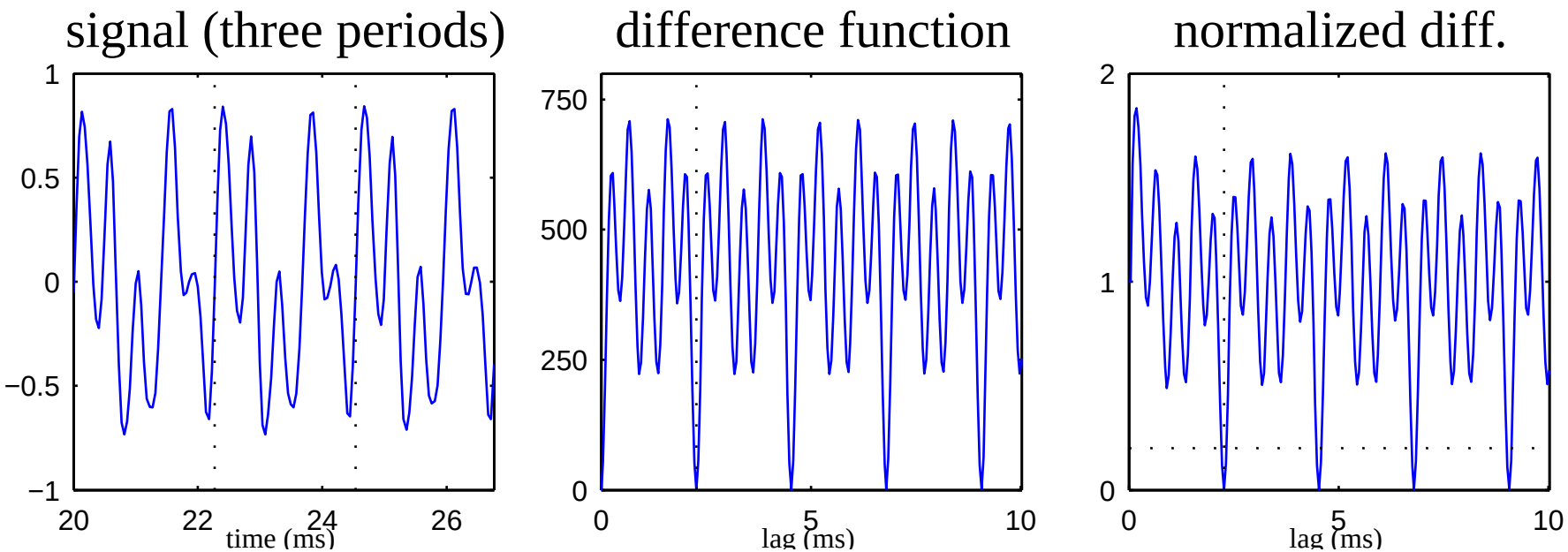
$$d(\tau) = \sum_{t=0}^{T-\tau-1} (x(t) - x(t+\tau))^2$$

  which measures the amount of energy in the signal which cannot be explained by a periodic signal of period τ
  (de Cheveigné & Kawahara, JASA 2002)
- The difference function is then normalised:

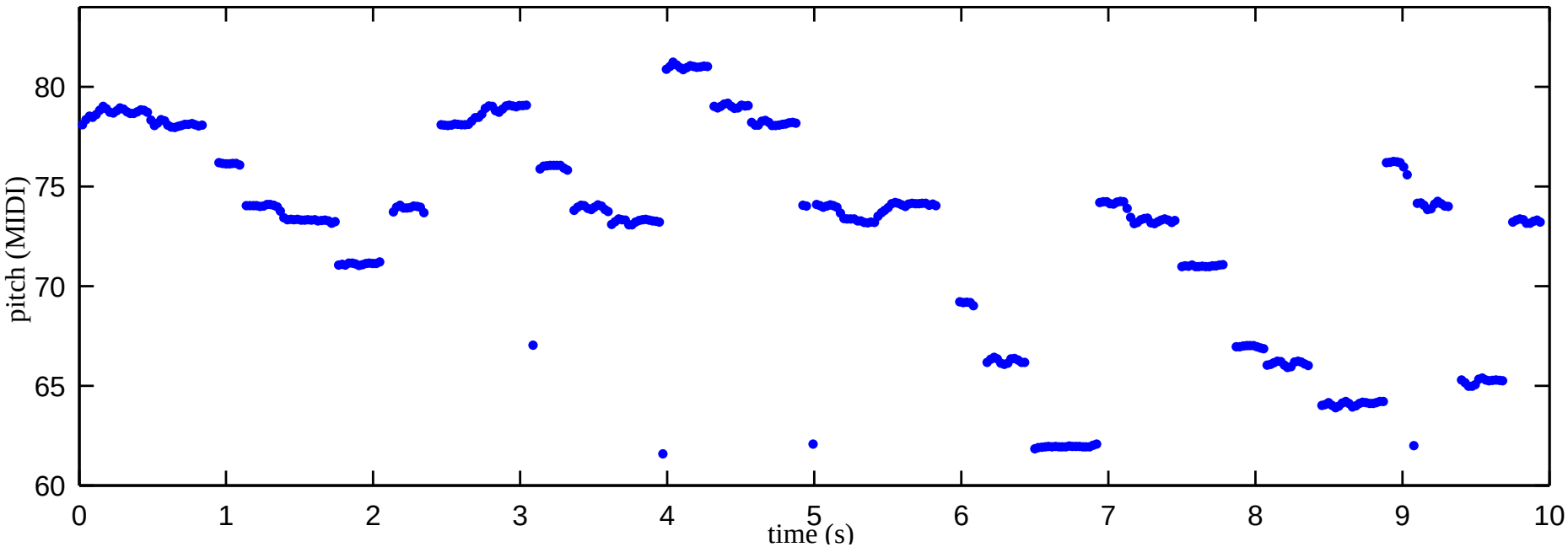
$$d'(\tau) = \frac{d(\tau)}{\frac{1}{\tau}\sum_{t=1}^{\tau} d(t)}$$

YIN

- The first minimum of d' below a fixed non-periodicity threshold corresponds to $\tau_0 = \frac{1}{f_0}$
- τ_0 is estimated precisely by quadratic interpolation
- The value $d'(\tau_0)$ gives a measure of how periodic the signal is: $d'(\tau_0) = 0$ if and only if the signal is periodic with period τ_0



YIN: Example



- YIN performs well on monophonic signals and runs in real-time
- Post-processing is needed to segment the output into discrete note events and remove erroneous pitches (mostly at note transitions)

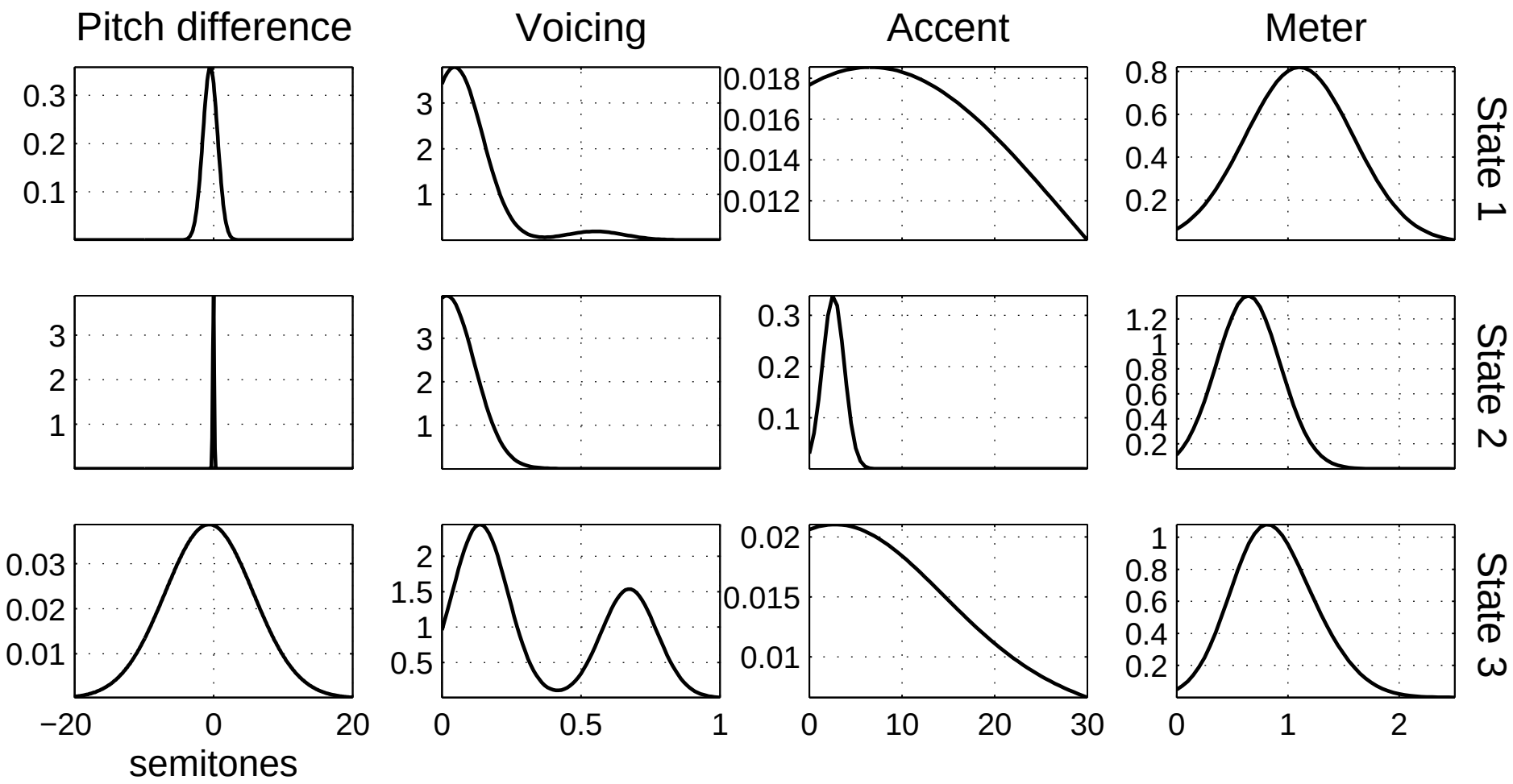
YIN: Monophonic Post-Processing

- The output from YIN can be improved by considering the following cues:
  - voicing (pitch information is less reliable on unvoiced frames, i.e. when $d'(\tau_0) >> 0$)
  - onsets (notes are more likely to begin at large values of an onset detection function)
  - tempo and meter (notes are more likely to begin on a beat or other metrically strong position)
  - musical context, e.g. key (pitches are not equally probable)
- These cues can be built into probabilistic note and phrase models (Ryynänen & Klapuri 2004)

YIN: Post-Processing with Probabilistic Note Models

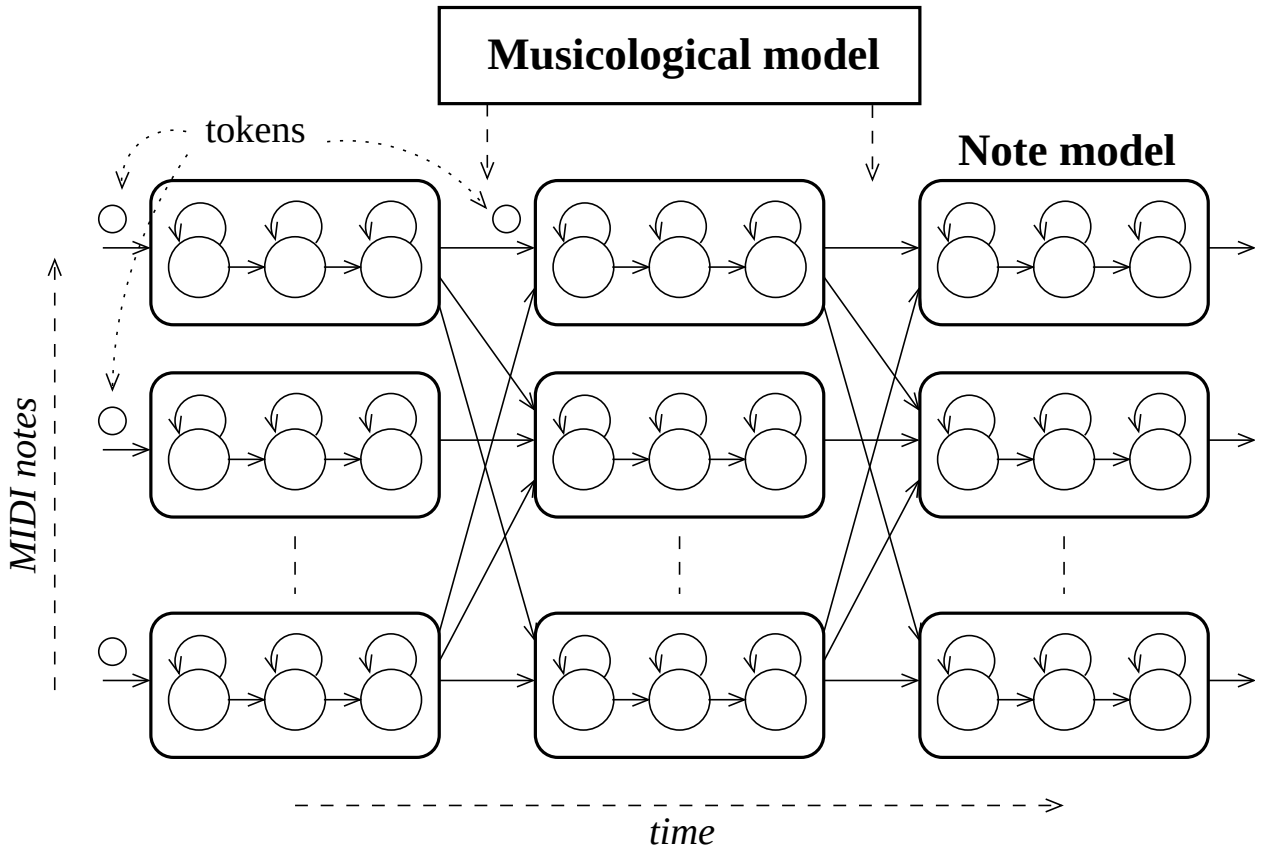
- Each note on the semitone scale is represented by a left-to-right *Hidden Markov Model* (HMM)
- Transition probabilities are constrained to avoid skips
- Features (pitch, voicing, onset detection function, meter function) are modelled by a Gaussian Mixture Model (GMM) conditionally to each state
- Silence is represented by a one-state HMM whose observation probability is equal to one minus the largest note observation probability
- Experimentally:
  - 3 to 4 states are enough to represent attack, sustain and transition
  - the optimal feature set contains pitch, voicing and meter

YIN: Post-Processing with Probabilistic Note Models



YIN: Post-Processing with Probabilistic Phrase Models

- Musical context can be represented using *n-gram* probabilities $P(c_t|c_{t-1}, \ldots, c_{t-n+1}, c_{\text{tonic}})$, where $c_t, c_{\text{tonic}} \in \{0, \ldots, 11\}$ are the chroma of the t-th note and tonic respectively
- This is a simple *language model*



PYIN: Extending YIN for Better Pitch Tracking

- YIN only estimates a single pitch candidate for each frame
- This pitch estimate depends on the value of the threshold
- PYIN is an extension to YIN (Mauch and Dixon, ICASSP 2014) which allows multiple candidates and tracks them over time with an HMM
- Given a prior probability distribution over the threshold, a set of pitch candidates and their probabilities are computed using YIN
- These values are then used as observations for a hidden Markov model with voiced and unvoiced states for each pitch on a 0.1 semitone step scale
- PYIN is more robust than YIN, giving better overall results, even on low quality (distorted) audio, and is less sensitive to threshold settings

Spectral Modelling with Non-negative Matrix Factorisation

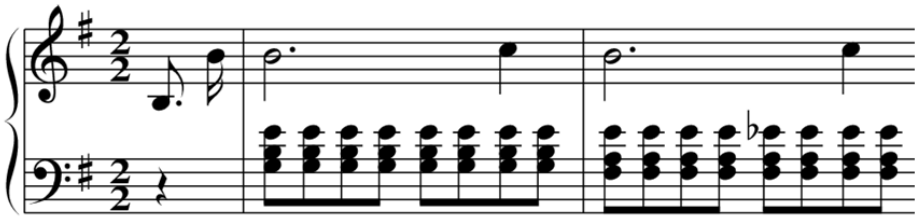
Nonnegative Matrix Factorisation

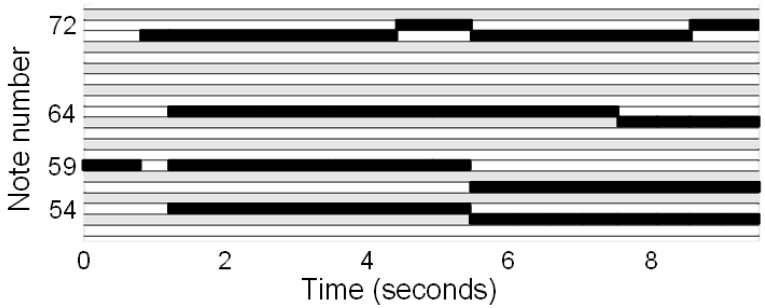
- Nonnegative Matrix Factorisation (NMF) models the observed short-term power spectrum $X_{f,n}$ as a sum of components with a fixed *basis spectrum* $W_{f,c}$ and a time-varying gain $H_{c,n}$ plus a residual or error term $E_{f,n}$ (Smaragdis 2003)

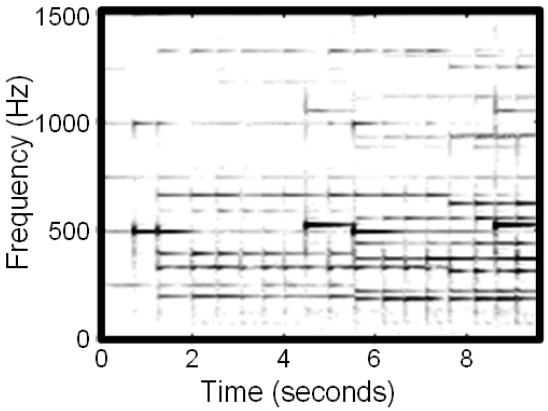
$$X_{f,n} = \sum_{c=1}^{C} H_{c,n} W_{f,c} + E_{f,n},$$

  or in matrix notation $X = WH + E$
- The only constraints on the basis spectra and gains are non-negativity and statistical independence of the basis vectors
- The residual is assumed to follow a Gaussian distribution

NMF Example

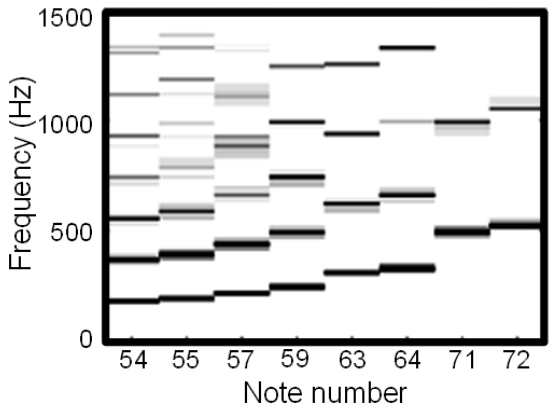






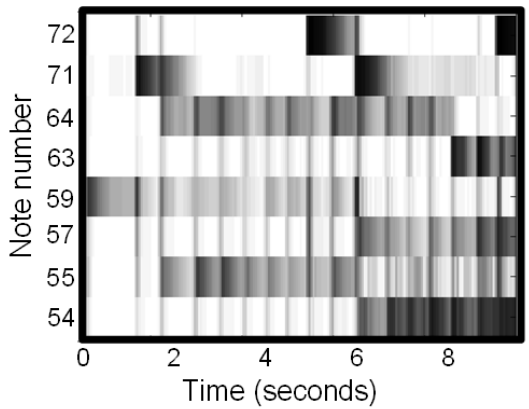
X

≈



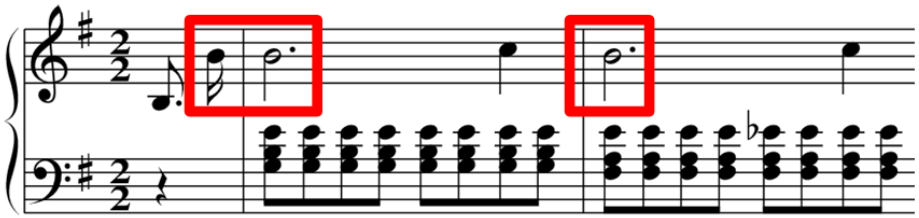
W

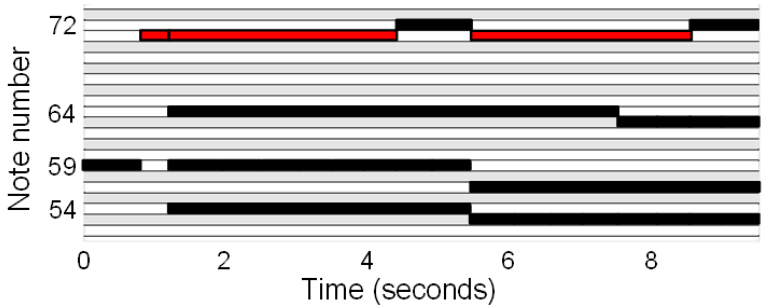
•

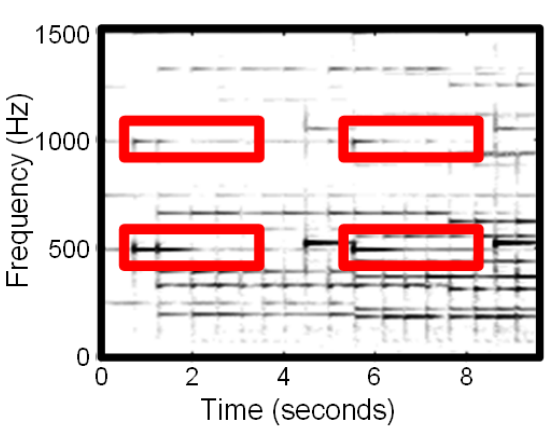


H

NMF Example

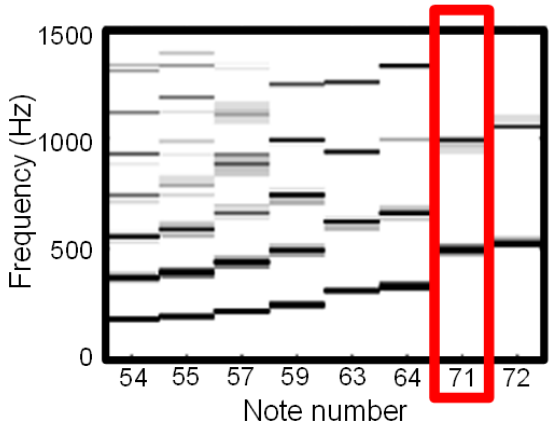






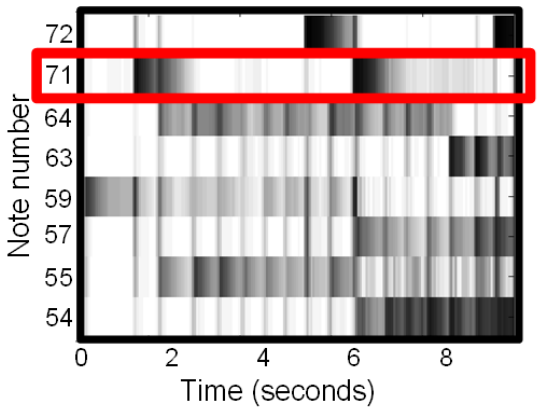
X

≈



W

•



H

NMF

- The aim is to find the basis spectra and the associated gains according to the *Maximum A Posteriori* (MAP) criterion:

$$(\widehat{W}, \widehat{H}) = \arg\max_{W,H} P(W, H|X)$$

- H and W are initialised with random values and iteratively updated using multiplicative update rules:

$$H_{c,n} := H_{c,n} \frac{(W^T X)_{c,n}}{(W^T W H)_{c,n}}$$

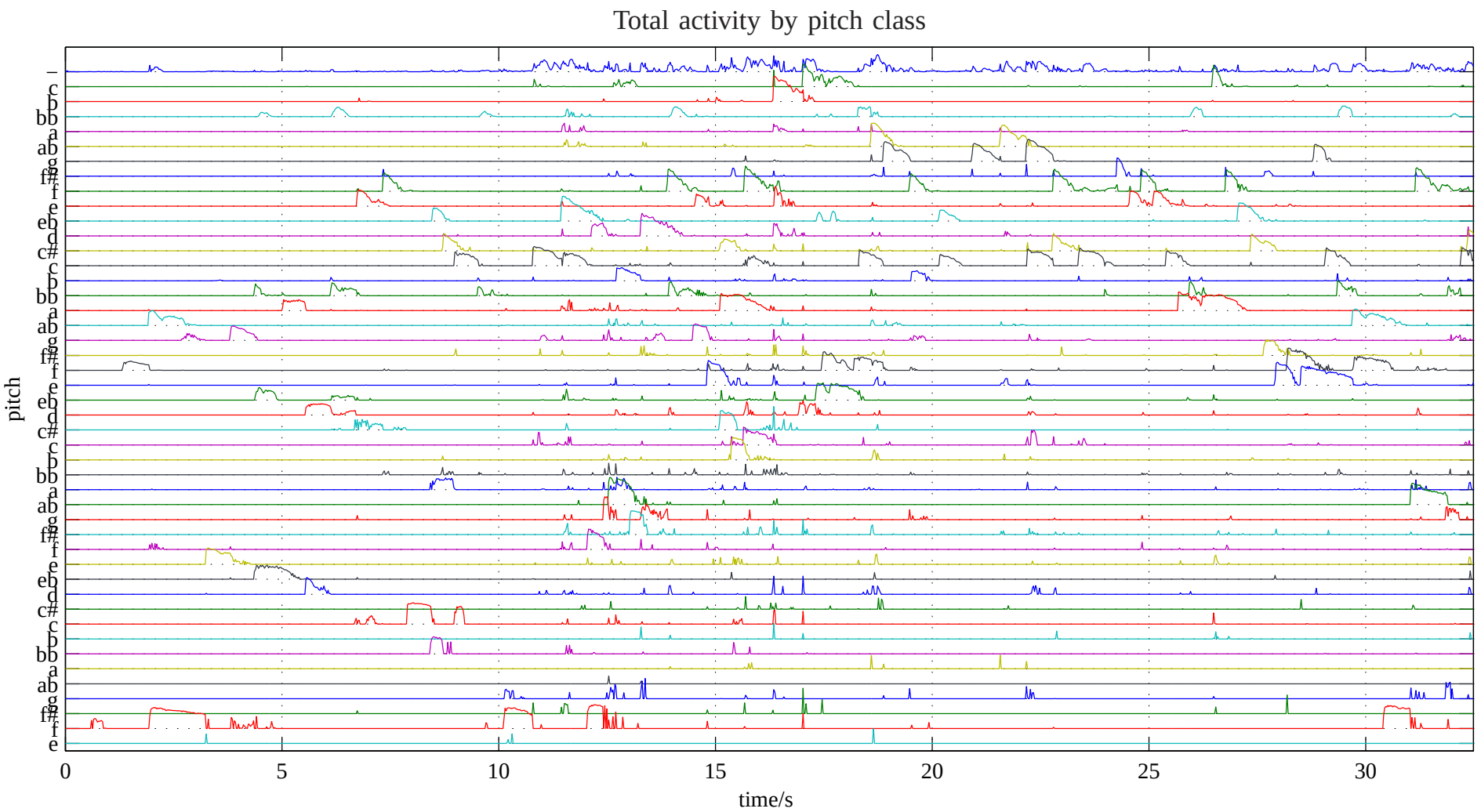
$$W_{f,c} := W_{f,c} \frac{(X H^T)_{f,c}}{(W H H^T)_{f,c}}$$

- Update rules ensure convergence to a local, not necessarily global, minimum of E

NMF

- This approach is valid for any instruments, provided the note frequencies are fixed
- The independence assumption tends to group parts of the input spectrum showing similar amplitude variations
- Basis spectra are not constrained to be harmonic, nor to have a particular spectral envelope
- They are not even constrained to represent notes: some components may represent chords or background noise
- Basis spectra must be processed to infer their pitch — one pitch might be represented by a linear combination of several basis spectra
- Variants of NMF add more prior information about the gains (e.g. requiring sparsity or temporal continuity), or the basis spectra (e.g. initialising with harmonic spectra or instrument templates)

NMF Example Output: Gain Matrix H



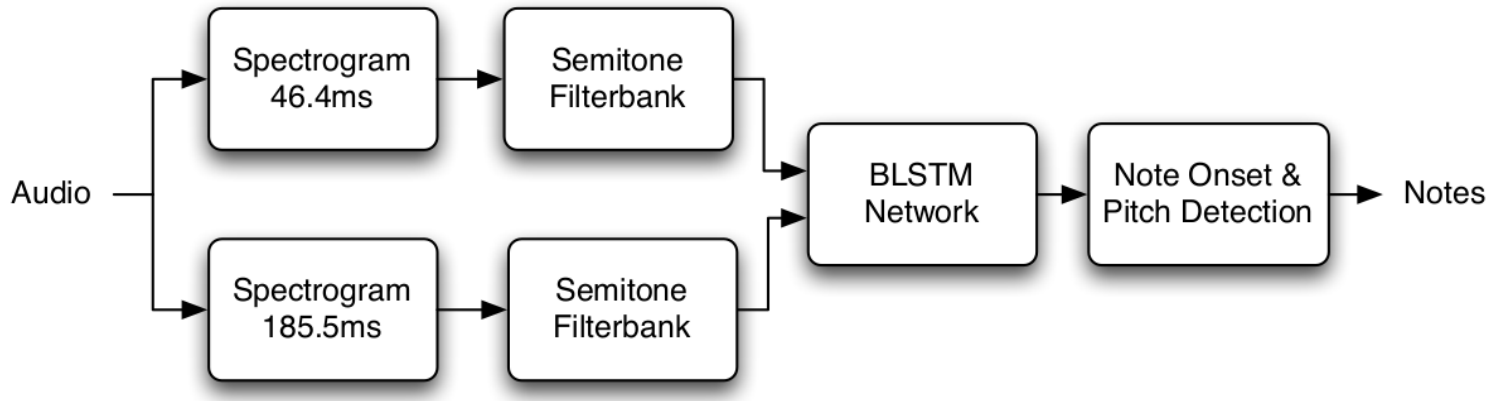
Transcription with Neural Networks

Neural Networks for Transcription

- Neural networks compute the composition of several non-linear functions, one for each layer of the network
- An optimisation problem is set up by defining a loss function which quantifies the difference between the system's current output and the target output
- The parameters of the network are learnt from data by minimising the loss using stochastic gradient descent
- Neural networks have recently achieved state of the art performance for pattern matching and signal processing problems such as image recognition, speech & language processing and source separation
- Recent progress can be attributed to the availability of larger data sets and more computing resources for training networks, as well as better techniques for training deep networks

Neural Network Transcription Systems

- Sonic (Marolt, 2001) used time-delay (TD) networks, (similar to convolutional networks in the time direction) to analyse the output of adaptive oscillators, to track and group partials outputted by a gammatone filterbank
- Böck & Schedl (2012) used 2 input spectrograms to enable the network to exploit both a high time accuracy (for onset detection) and high frequency resolution (for distinguishing near pitches); input is processed with Long Short-Term Memory (LSTM) layers to model temporal dependencies



Piano Transcription with Deep Networks

- Sigtia & Dixon (2016) proposed an end-to-end system consisting of an acoustic model and a language model
  - the acoustic model is a convolutional network that learns a frame-based probability distribution for pitch
  - the language model is a recurrent network trained on MIDI data to predict pitch combinations given past combinations
  - Kelz et al. (2016) tuned the acoustic model to reach similar performance with no language model
- A current baseline neural network transcription system would have a CRNN architecture
  - Input: time-frequency representation (e.g. CQT, VQT)
  - Convolutional layers for feature extraction
  - Recurrent layers for temporal smoothing
  - Output: frame-based piano-roll-like output

Piano Transcription: Onsets and Frames

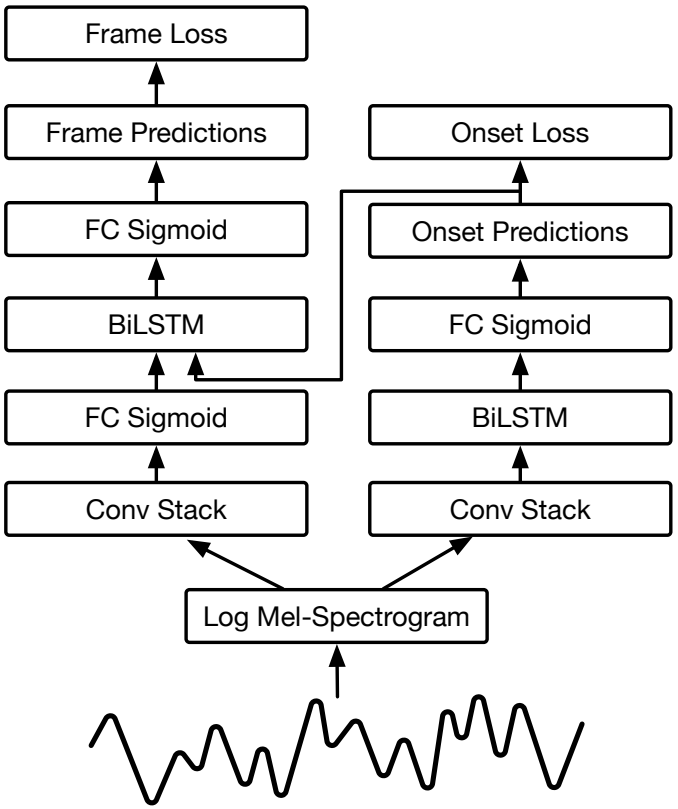
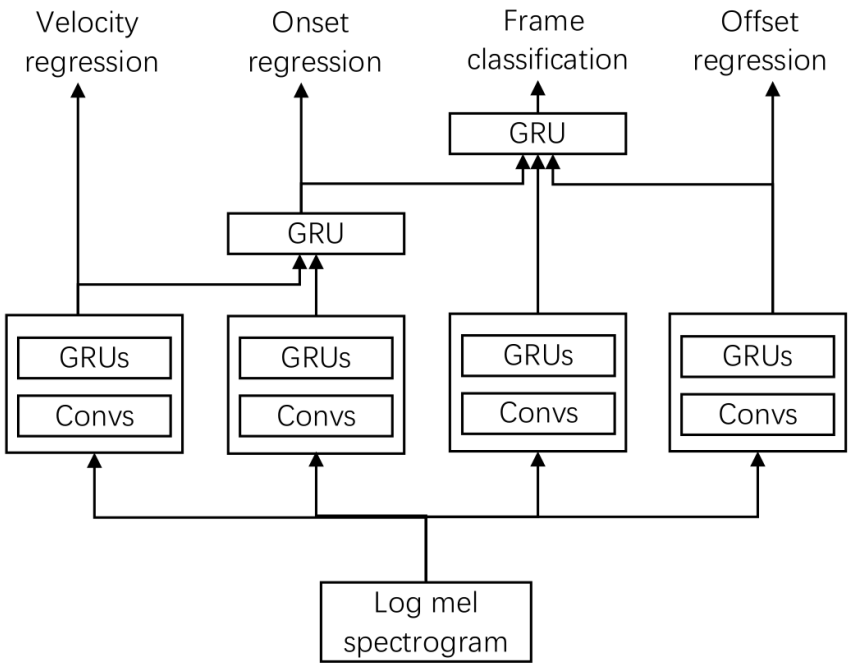


Figure 1. Diagram of Network Architecture

- Hawthorne et al. (2018) at Google Brain proposed a deep convolutional and recurrent neural network to predict onsets and frame-wise pitch jointly (*multi-task* learning)
- Both sides of the network combine convolutional layers with a bi-directional LSTM layer and fully-connected output layer
- Onset predictions are fed into the framewise detector, which allows new pitches only after an onset
- Captures differing temporal dynamics of onsets and pitch activities

High Resolution Piano Transcription

- Kong et al. (2021) at Bytedance focussed on sub-frame level timing of onsets, offsets, velocities and pedalling using regression
- Learning target: distance from centre of frame to nearest onset
- 4 acoustic models: conv. stack, FC, 2×biGRU and FC layers
- Onset predictions are conditioned on velocity; frame predictions are conditioned on onset and offset predictions
- Results: 96.7% for onsets on Maestro (Onsets & Frames: 94.8%)



MT3: Multi-instrument Transcription

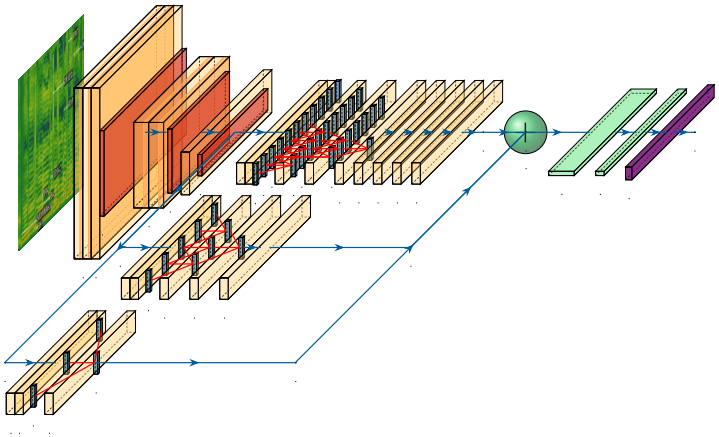
- Gardner et al. (2021) used a general purpose Transformer model (T5) to jointly transcribe multiple instruments
- Low-resource sequence learning task, similar to some NLP tasks
- Single text-to-text encoder-decoder model trained across multiple diverse datasets in 2-second chunks
  - Synthetic (Slakh2100, Cerberus4) and acoustic (MAESTROv3, MusicNet, GuitarSet, URMP) datasets
  - Spectrogram input, MIDI-like text output with absolute positional embeddings
  - Sampling strategy to partially balance dataset sizes
  - T5 “small” model: 60M parameters; larger $\rightarrow$ overfitting
  - Greedy autoregressive decoding
- MT3 outperforms previous state of the art systems
- Version trained on all datasets better than on specific datasets

Automatic Lyrics Transcription

- Task: determine words from audio containing singing and instruments
- Similar to speech recognition, but:
  - Interference from instrumental accompaniment
  - Different pronunciation (esp. phoneme lengths)
  - Off-the-shelf speech recognition systems fail (badly)
- MSTRE-Net (Demirel et al., 2021)
  - Hybrid DNN-HMM architecture using the Kaldi framework
  - Acoustic model: neural network with convolutional (CNN), multistreaming time-delay (MTDNN) and fully connected layers
  - Language model: trained on lyrics
  - Pronunciation model: CMU dictionary with extended vowels and automatically generated out-of-vocabulary pronunciations
  - Trained on accompanied and unaccompanied singing
  - Two “no-vocal” tags in training: accompaniment, silence

MSTRE-Net Acoustic Model and Results

- Input processing: 2-D CNN
- MTDNN variant: number of hidden layers varies between streams
- Phoneme activation output: 2 fully connected and a softmax layer



- Word error rates on unaccompanied (DAMP) & accompanied data:

	DAMP-test	Jamendo	DALI-test
Dabike (2019)	16.86	67.12	76.37
Gupta (2020)	56.90	50.64	N/A
Demirel (2020)	17.16	66.96	76.72
Demirel (2021)	**15.38**	**34.94**	**42.11**

Conclusion: Transcription Issues

- Perceptual intentions of music oppose its transcription
  - Consonant pitch intervals and synchronised onsets make notes "blend" together better
- Other difficulties include inharmonicity, presence of drum sounds, and unknown number of notes
- Pitch estimation could benefit from integration of multiple cues:
  - musicological cues
  - timbre cues
  - prior information about the signal to be transcribed
- Source separation systems look promising as a preprocessing step
  - But in principle it is better to perform pitch estimation and streaming jointly rather than successively
- Deep learning approaches have caught up
  - End to end systems look set to outperform traditional approaches

Further Resources

- E. Benetos, S. Dixon, Z. Duan, and S. Ewert (2019), *Automatic Music Transcription: An Overview*, IEEE Signal Processing Magazine, 36 (1), 20–30
- Q. Kong, B. Li, X. Song, Y. Wan, Y. Wang (2020), *High-resolution Piano Transcription with Pedals by Regressing Onsets and Offsets Times*, arXiv:2010.01815
- C. Hawthorne, E. Elsen, J. Song, A. Roberts, I. Simon, C. Raffel, J. Engel, S. Oore, and D. Eck (2018), *Onsets and frames: Dual-objective piano transcription*, Proceedings of the International Society for Music Information Retrieval Conference, pp 50–57
- E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, A. Klapuri (2013), *Automatic Music Transcription: Challenges and Future Directions*, Journal of Intelligent Information Systems, 41 (3), 407–434
- A. Klapuri and M. Davy, Eds. (2006), *Signal Processing Methods for Music Transcription*, New York: Springer-Verlag