Indexing Audio

Audio Technology
Human Ear: Acoustic Sound Waves

- Frequency: ~ Pitch (Tonlage)
- Amplitude: ~ Volume
- Subsonic noise 0 Hz up to 20 Hz
- Audible sound 20 Hz up to 20 KHz
- Intensive domain: 600 Hz - 6.000 Hz
- Ultrasound: 20 KHz up to 1 GHz

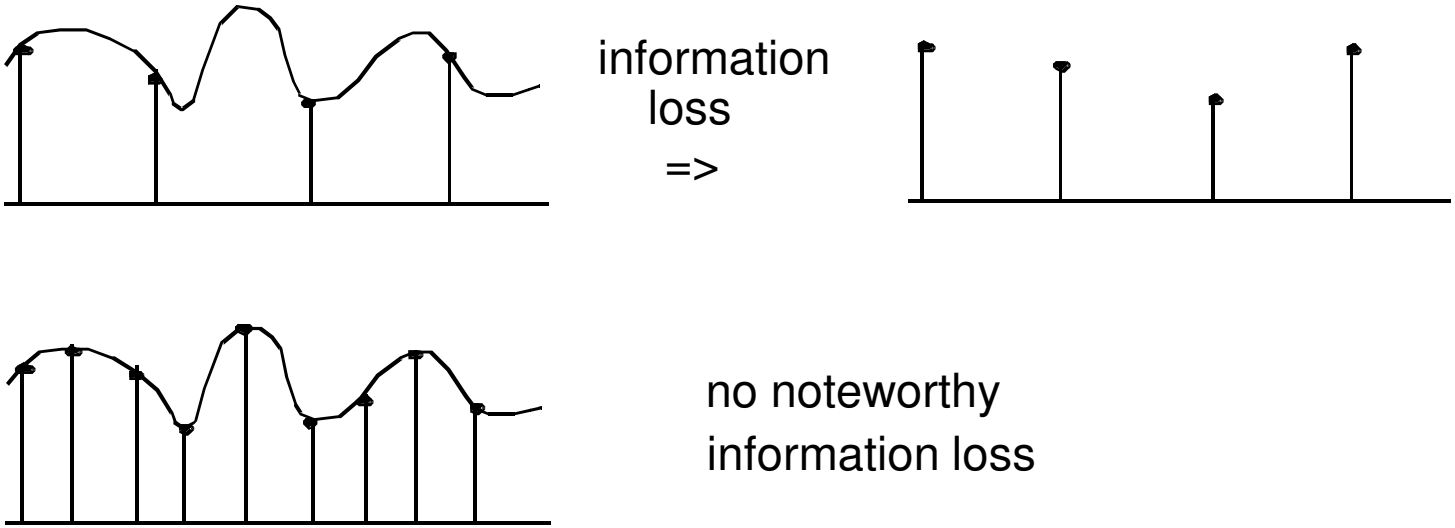
- In general no pure sinus waves: overlaps (harmonic, “tone”)

On Storage: Three Possibilities

1. Analog storage: compare to vinyl records, audio tapes.
2. Digital storage: e.g. audio CDs
3. Symbolic storage: MIDI-technology, e.g. synthesizer

Digital Audio Storage

- Sampling Theorem (Nyquist-Shannon):
  - Convert continuous signal into discrete signal
  - The sampling frequency has to be greater than twice the highest frequency occurring in the signal.



Digital Audio Storage

- Example:
  - Audible domain: 20 Hz - 20.000 Hz
  - Telephone: 4.000 Hz → > 8.000 sample points / sec.
  - Hifi: 22.000 Hz → 44.100 sample points / sec. (per channel)

Symbolic Storage: MIDI-Technology

- For music only (not suitable for language)
- Basic concept:
  - Instead of digitalization of the sound, the score itself is stored (compare to synthesizer)
  - Tuple:

    (Instrument, begin of a tone, end of a tone, fundamental frequency, volume)
  - MIDI enables:
    - 10 octaves
    - 128 instruments (incl. noises) altogether (piano, strings, brass, etc.)
    - Simultaneously: 16 channels = 16 instruments
    - Simultaneous tones per channel: 3-16 (quality feature of synthesizers) (needed e.g. for organs, not for flutes)

Symbolic Storage: MIDI-Technology

- Example: key of a piano, begin of a tone, end of a tone, release of the key, key = tone, velocity (Anschlagstärke).
  - 10 min. music ~ 200 KByte => 0,33 $\frac{\text{KByte}}{\text{sec}}$ data rate
- Result:
  - Compact description of music files, which enables lifelike playback.

Symbolic Storage: MIDI

- I/O:
  - Input via keyboard (piano keyboard) or edit tones at the screen
  - Previewer: PC
- Advantages of MIDI:
  - Compact presentation
  - Easily editable (sequencer: editor, converts the sheet of music into an internal MIDI format)
  - Extendable by intra document anchors (and so there exist links)
- Disadvantages of MIDI:
  - Not suitable for language
  - Artistic interpretation for one play? (only limited)
  - No documentary footage

Content-Based Search in Music Databases

- Goals :
  - Melody → Piece of music
  - Similar pieces of music, plagiarism, etc.
  - Assignment, classification, genre identification

MusicDB

- Lyrics
  - Easy => performing a text query
- Metadata Tags
  - iTunes
    - caches information from the audio format's tag (ID3)
- Music Genome Project
  - Discover new music
  - Collect details on every song
    - melody, harmony, instrumentation, rhythm, vocals, lyrics

    http://en.wikipedia.org/wiki/List_of_Music_Genome_Project_attributes
  - Find songs with interesting musical similarities
    - cf. Pandora – a subproject of the Music Genome Project

- Music Recommendation, Next Song Prediction
  - DBIS reseach papers
    - https://dbis-informatik.uibk.ac.at/context-aware-music-recommendation

Introduction

- Query by Humming-System (QbH)
  - Allows user to find a song by humming the tune
  - Challenges:
    - No perfect queries
    - Capturing pitches and tones from user is difficult
    - Melody extraction of a pre-recorded music file is difficult

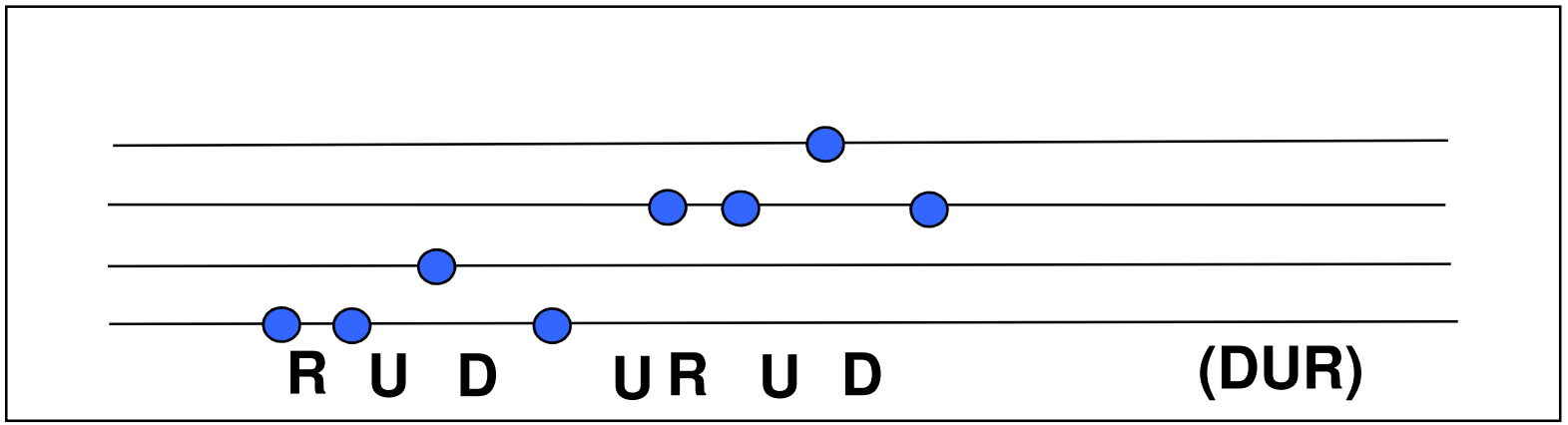
Query by Humming Concepts

- Based on MIDI files
  - No complicated pitch extraction necessary

- Indexing possibilities
  1. String-matching (-)
  2. DUR (UDS)
  3. CubyHum (Extension to UDS)
  4. QPD

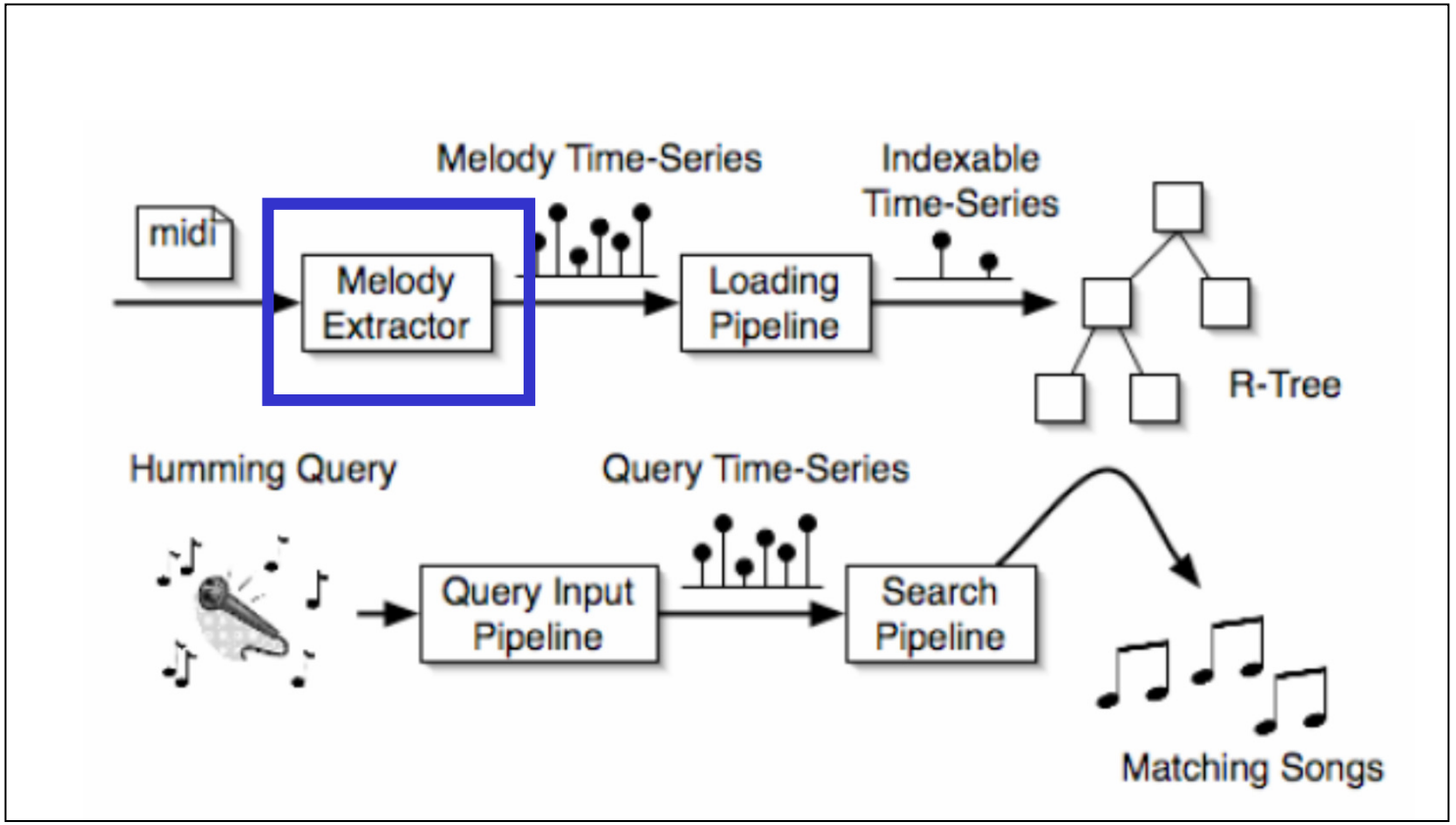
DUR

- MIDI -> Extraction of melody tracks
- Similarity match according DUR (simple approach)



Query by Humming: MusicDB

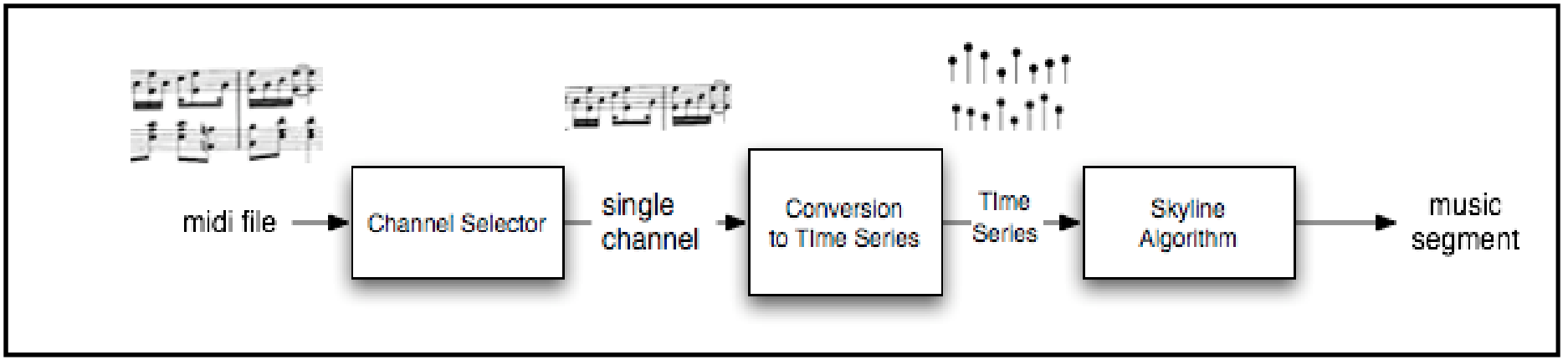
- Architecture of MusicDB



MusicDB: A Query by Humming System,
Edmond Lau, Annie Ding, Calvin On

MusicDB Concepts

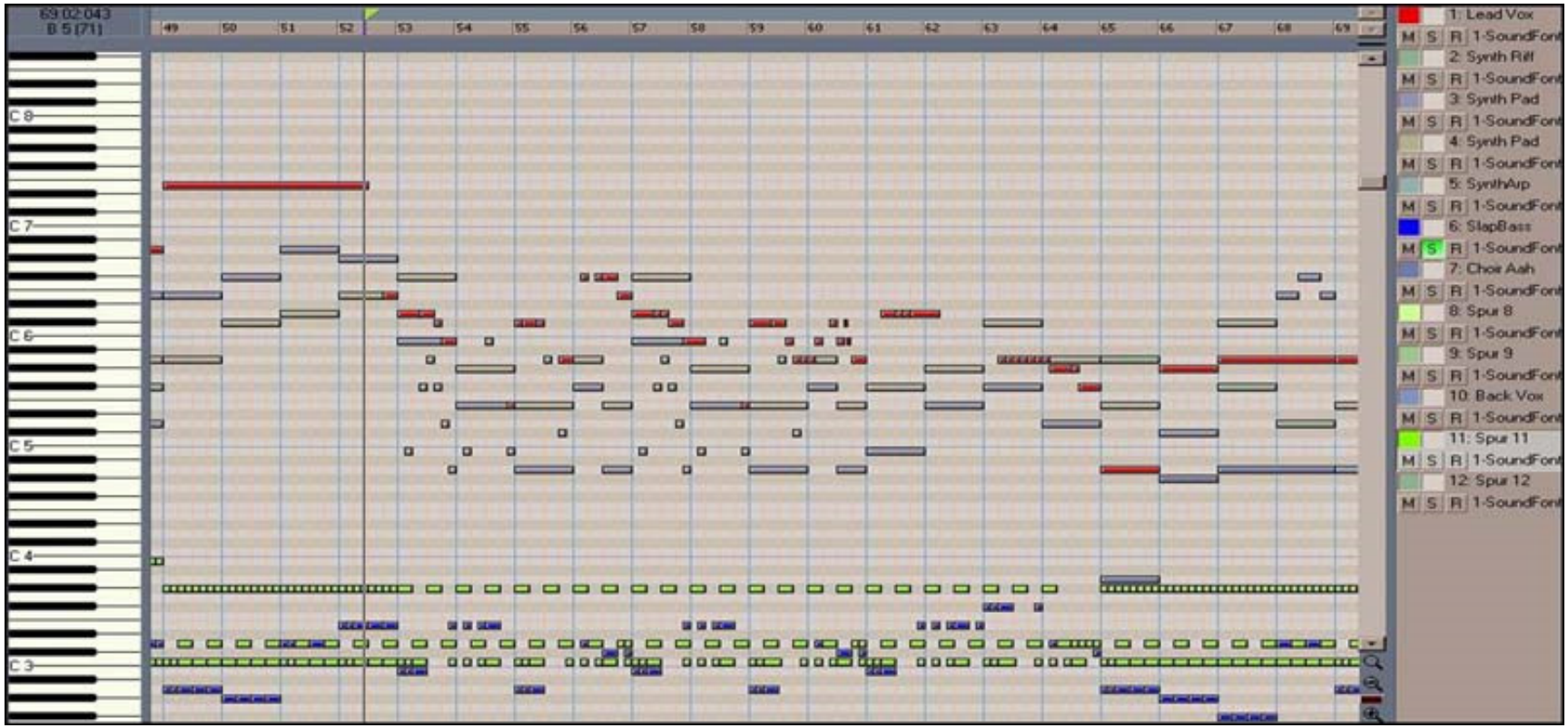
- Melody Extractor
  - Conversion of MIDI files into time series



MusicDB: A Query by Humming System,
Edmond Lau, Annie Ding, Calvin On

MusicDB Concepts

- Isolation of the channel containing the main melody
  - Melody is a time series characterized by the highest pitched (connected) tones in a song
  - MIDI data of a song:



Screenshots taken from Sonar 4

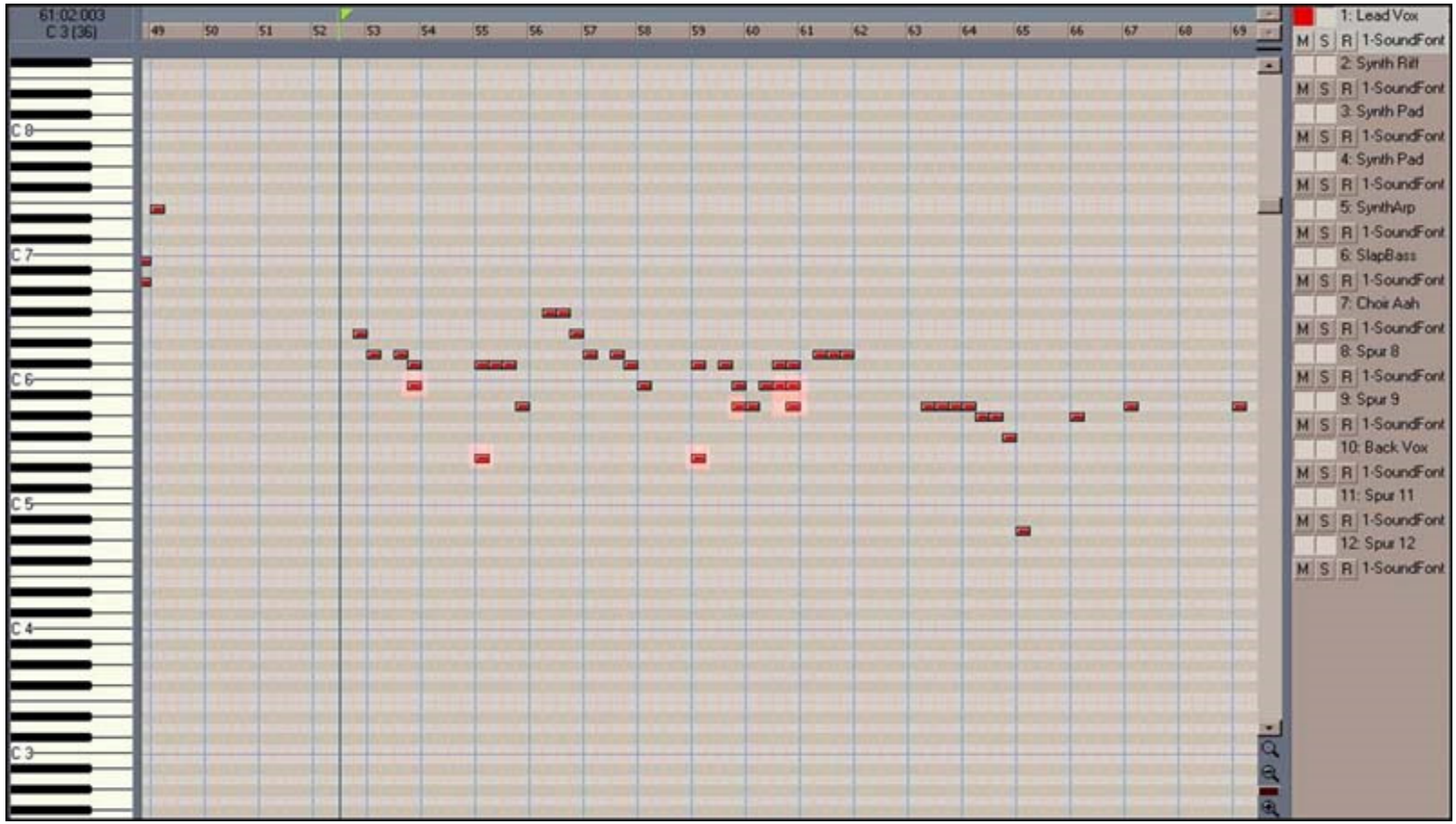
MusicDB Concepts

- Remove certain tracks and isolate the channels with the top 3 average pitch values & choose channel with most single tones

Screenshots taken from Sonar 4

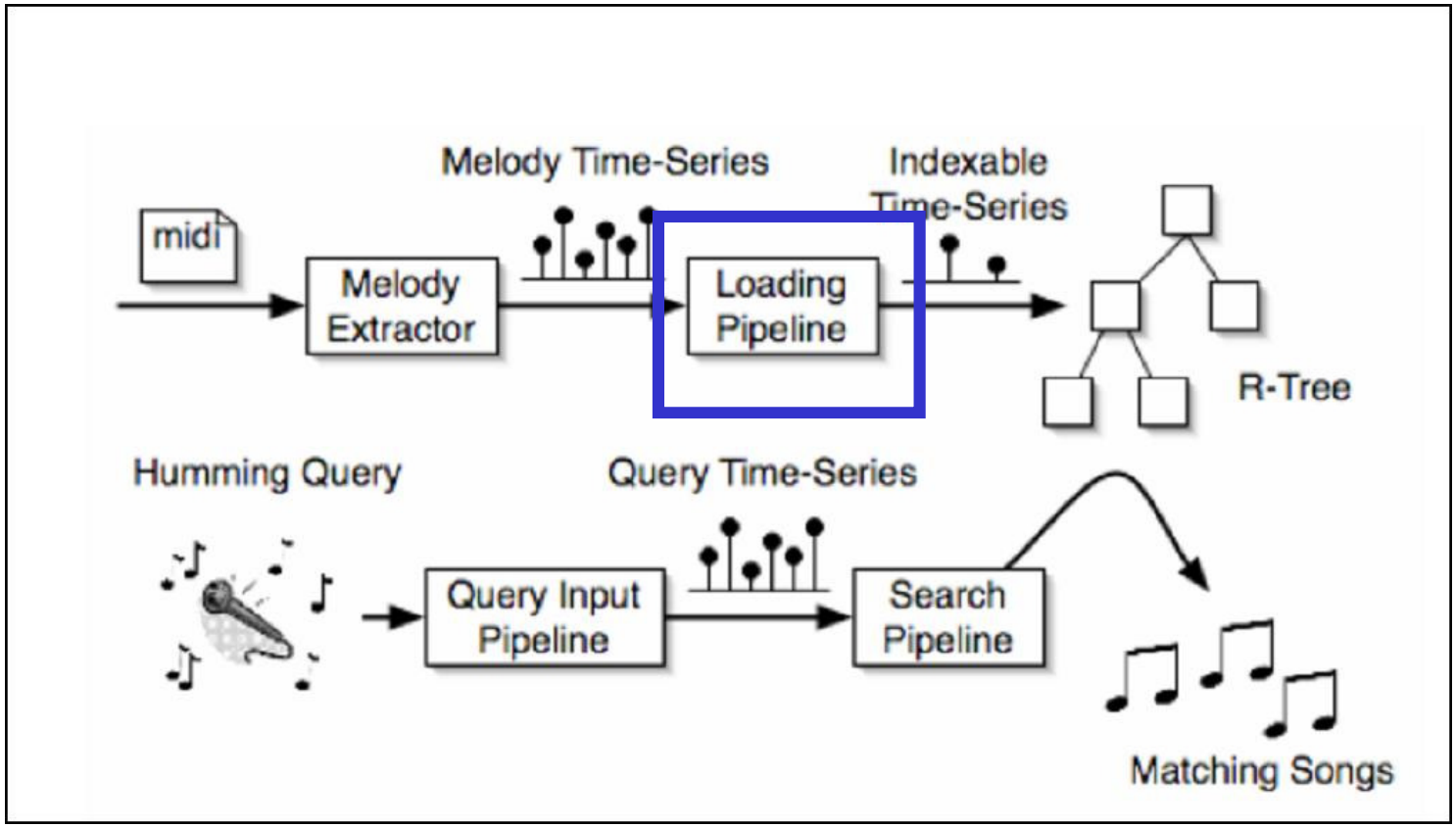
MusicDB Concepts

- Tone duration and start-time get quantized and overlapping tones will be deleted (Skyline-Alg)



Screenshots taken from Sonar 4

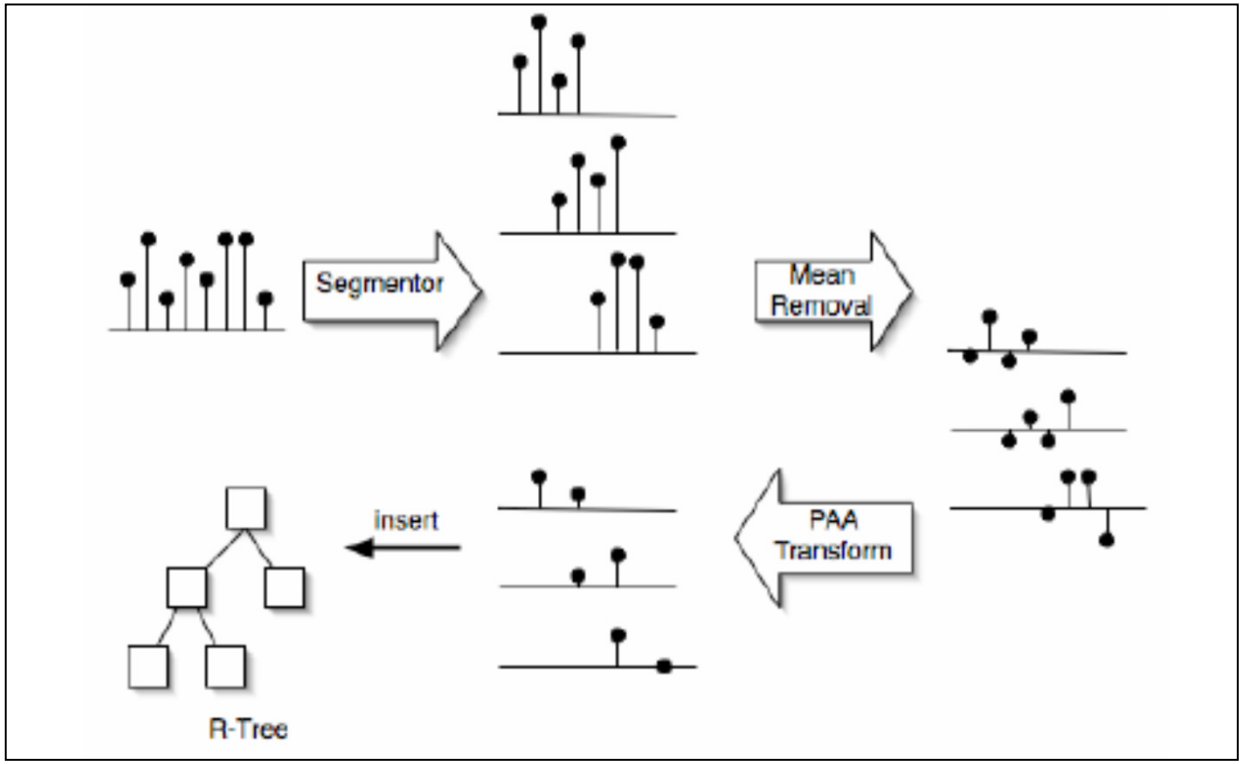
MusicDB Concepts



MusicDB: A Query by Humming System,
Edmond Lau, Annie Ding, Calvin On

MusicDB Concepts

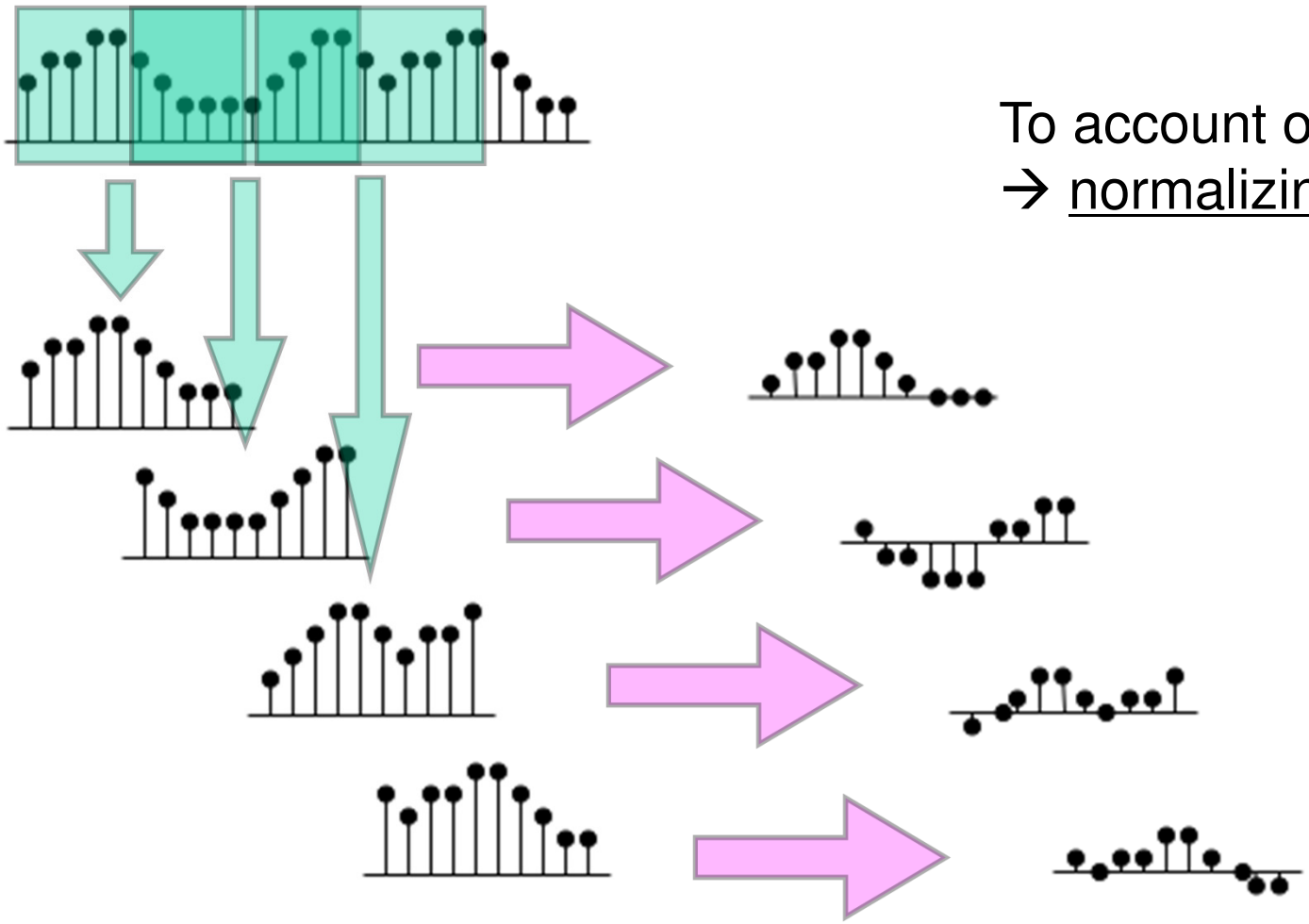
- Loading Pipeline



MusicDB: A Query by Humming System,
Edmond Lau, Annie Ding, Calvin On

MusicDB Concepts

Melody Time Series



To account off-key humming
→ normalizing

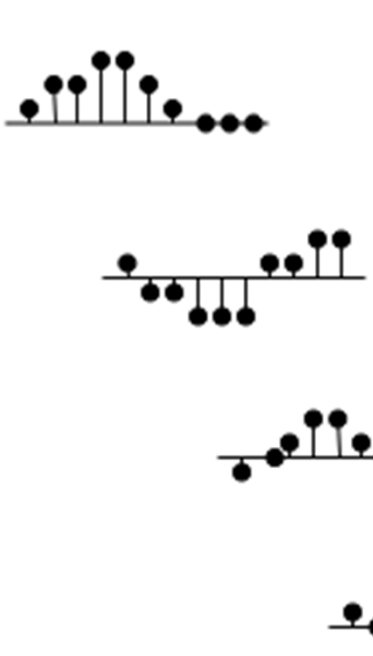
Time-series Slices

Normalized Time-Series

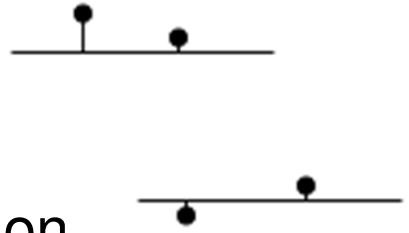
MusicDB Concepts

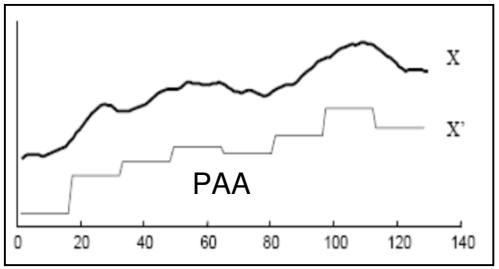
Time-Series Database

Transformed Database



PAA –
dimensionality reduction

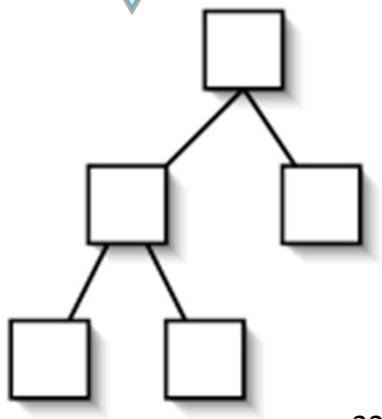




Insert

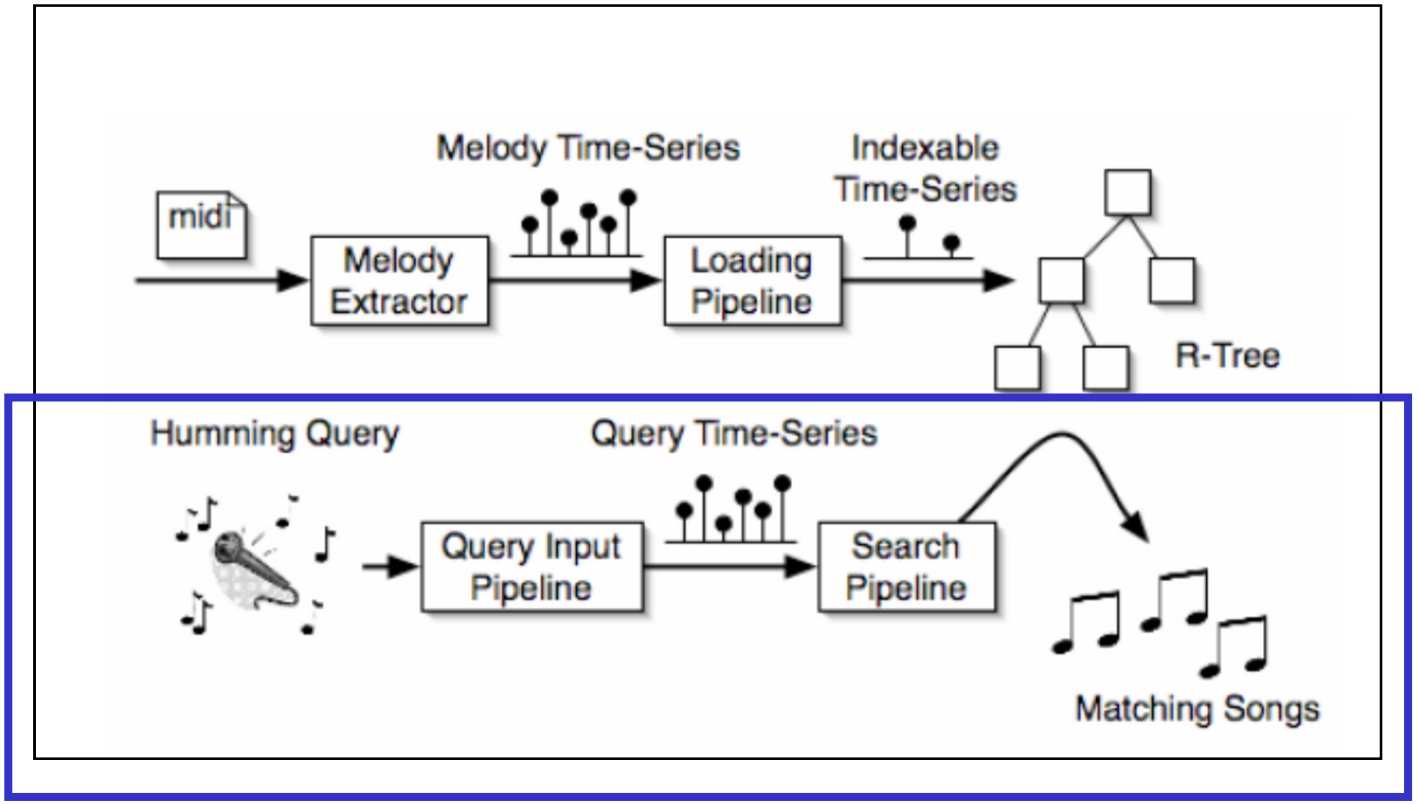
Insert Into R-tree with N dimensions

R-Tree



PAA: Piecewise Aggregate Approximation (take average of window of size n)
(Resulting transform is similar to a **Haar wavelet transform**)

MusicDB Concepts



MusicDB: A Query by Humming System,
Edmond Lau, Annie Ding, Calvin On

MusicDB Concepts

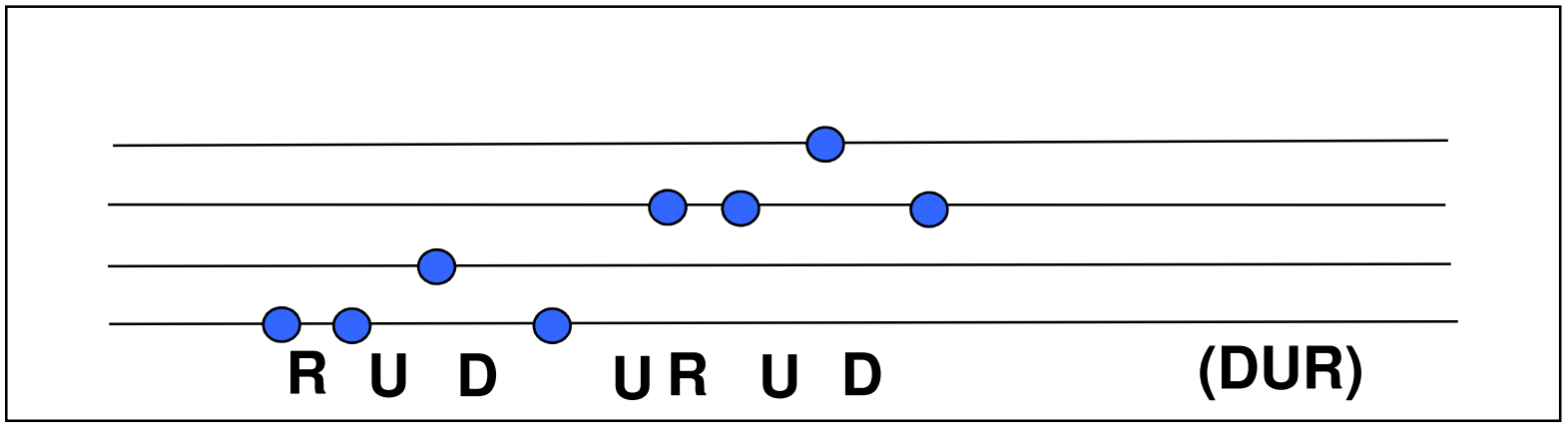
- Query
  - Same procedure for user-humming as for the MIDI files in the database
  - Comparison of both time series (user & R-tree) according to a distance metric
  - Results get ordered

Content-Based Search: Suitable Indexing

- QPD Approach in Detail:
  - QPD (Query by Pitch Dynamic)

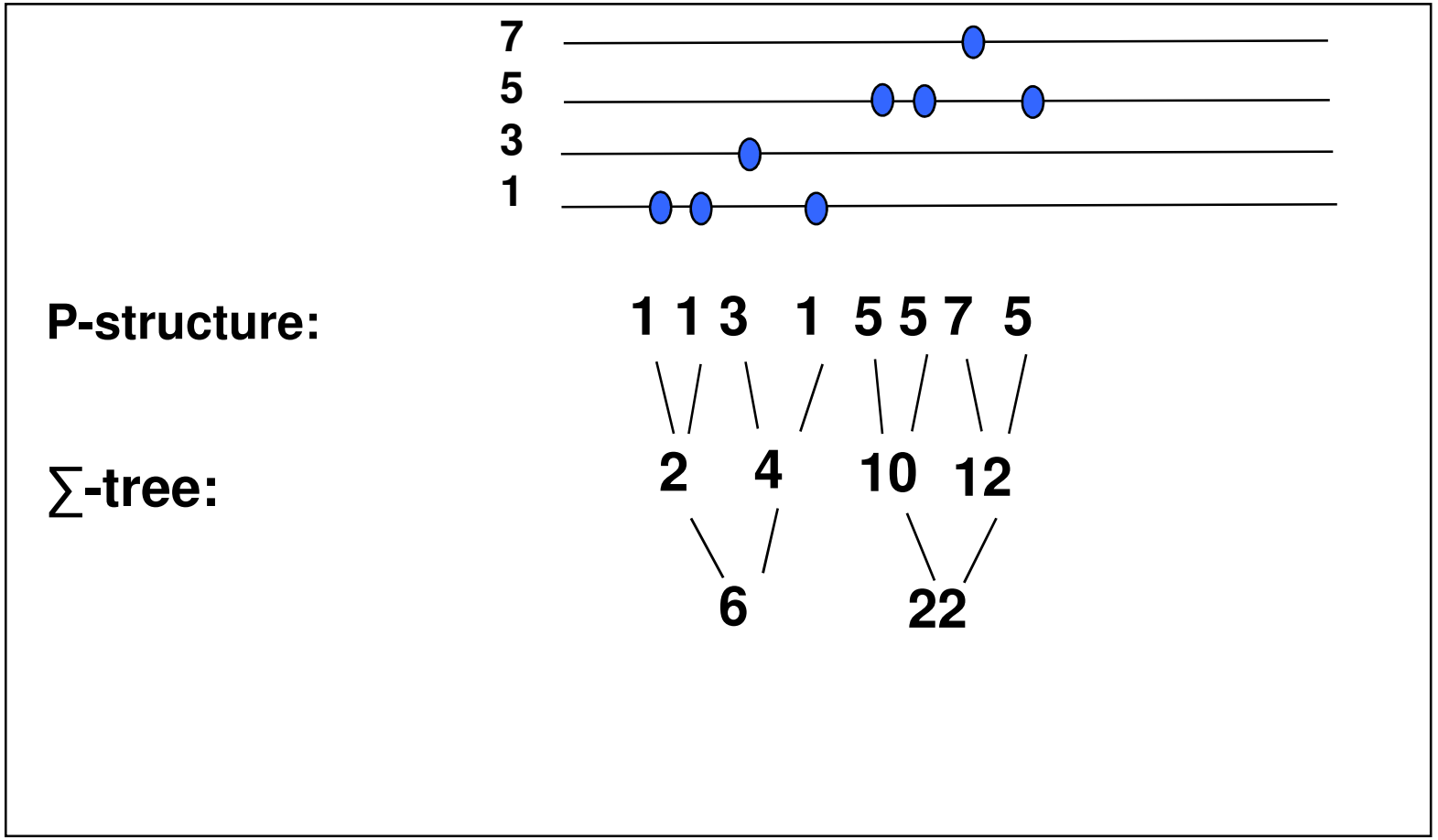
Query by Humming: DUR

- Recall: Simplest Approach DUR



Query by Pitch Dynamics (QPD)

- QPD Example

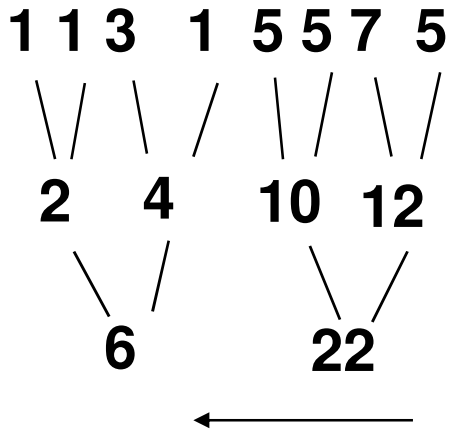


QPD

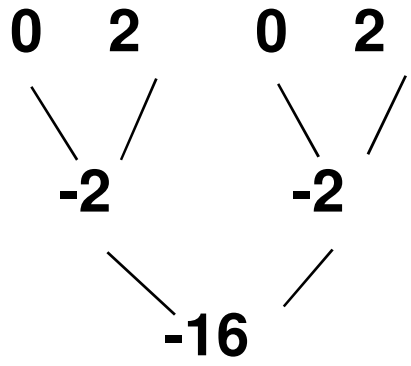
- Example cont'd

P-structure:

∑-tree:



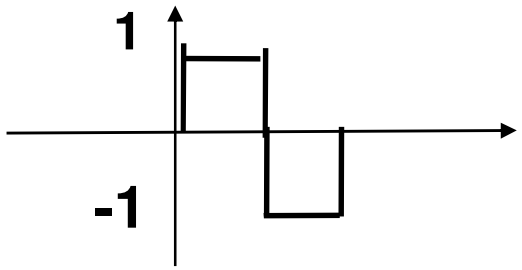
hierarchical differences:



QPD

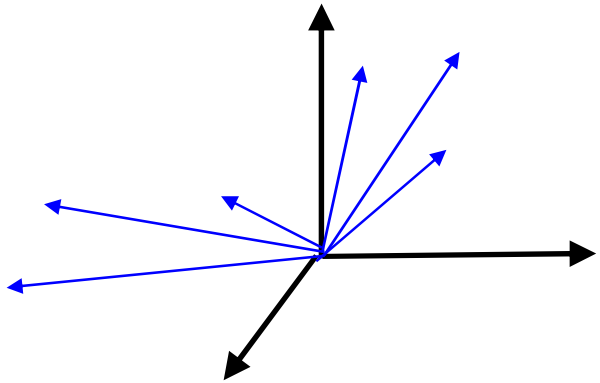
- Example cont’d

Haar-Wavelet Transformation (-16, -2, -2, 2, 0, 2, 0)



QPD

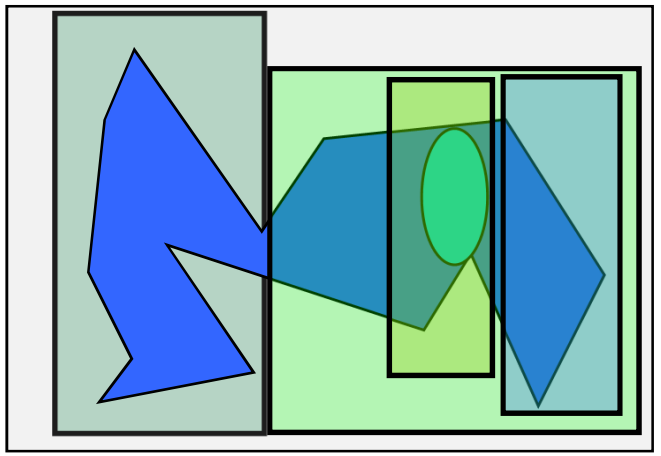
Vectors respectively points in the 2^{k-1}-dim. space



Distance metric:

$$d_E = \sqrt{(\vec{v}_1 - \vec{v}_2) \circ (\vec{v}_1 - \vec{v}_2)}$$

Storage e.g. in an R-tree:



Melody search: Point search
Similarity search: Bound-box search
Classifications: Cluster search