Introduction to Audio Content Analysis

Module 7.5: Musical Key Recognition

alexander lerch

Georgia Tech Center for Music Technology
College of Design

Georgia Tech Center for Music Technology
College of Design

corresponding textbook section

Section 7.5

- **lecture content**
  - definition of musical key
  - pitch chroma feature
  - standard approach for key recognition

- **learning objectives**
  - explain the defining properties of a musical key
  - implement a simple pitch chroma feature extractor
  - describe and discuss a simple automatic key recognition system

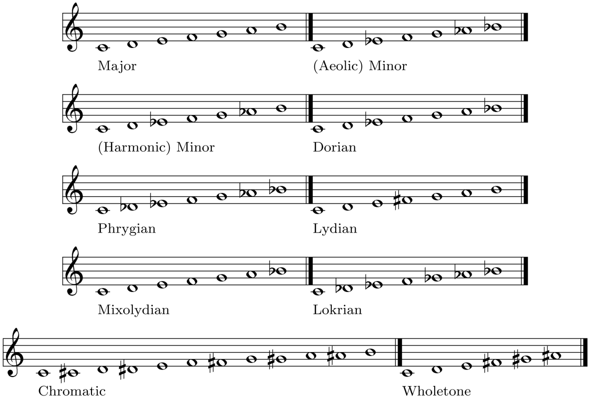
corresponding textbook section

Section 7.5

- **lecture content**
  - definition of musical key
  - pitch chroma feature
  - standard approach for key recognition
- **learning objectives**
  - explain the defining properties of a musical key
  - implement a simple pitch chroma feature extractor
  - describe and discuss a simple automatic key recognition system

key
tonic & mode

- **tonic**: first scale degree
  - most “important” pitch class
- **mode**: set of diatonic pitch relationships
  - Major: 2, 2, 1, 2, 2, 2, 1
  - Minor: 2, 1, 2, 2, 1, 2, 2



- **key**:
  defined by *tonic* (root note) and *mode*
  - defines a set of pitch classes constructing both pitch and harmonic content

- **modulation** (local key changes):
  common in various styles, uncommon in others

- **key signature**:
  indicates current key with accidentals (score notation)

- **key**:
  defined by *tonic* (root note) and *mode*
  - defines a set of pitch classes constructing both pitch and harmonic content

- **modulation** (local key changes):
  common in various styles, uncommon in others

- **key signature**:
  indicates current key with accidentals (score notation)

key
key & key signature 1/2

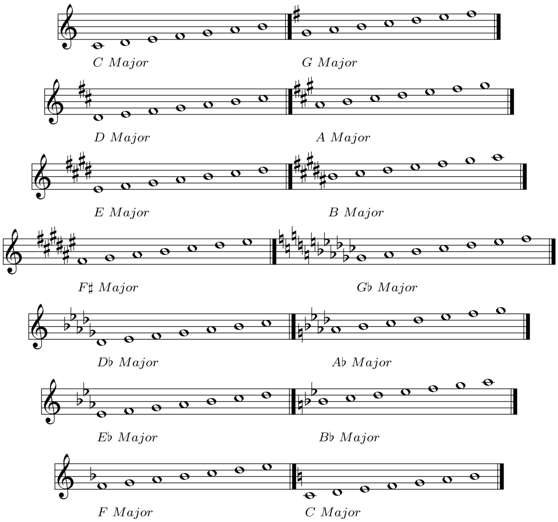
- **key**:
  defined by *tonic* (root note) and *mode*
  - defines a set of pitch classes constructing both pitch and harmonic content

- **modulation** (local key changes):
  common in various styles, uncommon in others

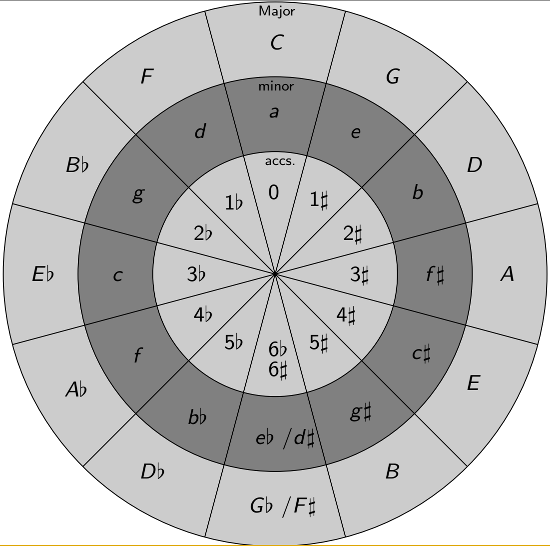
- **key signature**:
  indicates current key with accidentals (score notation)

key

key & key signature 2/2

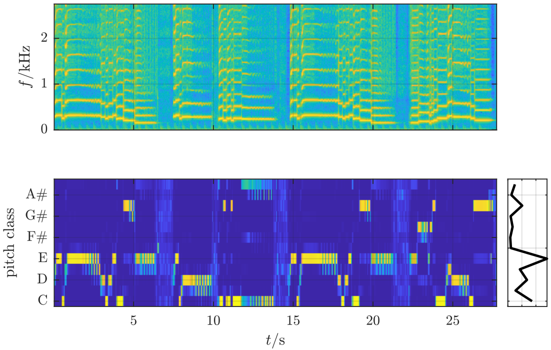


musical pitch
key: circle of fifths



pitch chroma
introduction

- pitch class distribution
- 12-dimensional vector
- **no** octave information
  - robust representation
  - no differentiation between unison and octave



matlab source: plotPitchChroma.m

pitch chroma
computation 1/2

1. divide spectral representation into **semi-tone bands**
2. compute **mean per band**

$$\mu(j,n) = \frac{1}{k_\mathrm{u}(j) - k_\mathrm{l}(j) + 1} \sum_{k=k_\mathrm{l}(j)}^{k_\mathrm{u}(j)} |X(k,n)|^2$$

3. sum/mean every 12th band

$$\begin{aligned} \nu(j\%12, n) &= \sum_{o=o_l}^{o_u} \mu(j,n), \\ \nu(n) &= [\nu(0,n), \nu(1,n), \nu(2,n), \ldots, \nu(10,n), \nu(11,n)]^\mathrm{T} \end{aligned}$$

pitch chroma
computation 1/2

1. divide spectral representation into **semi-tone bands**
2. compute **mean per band**

$$\mu(j,n) = \frac{1}{k_{\mathrm{u}}(j) - k_{\mathrm{l}}(j) + 1} \sum_{k=k_{\mathrm{l}}(j)}^{k_{\mathrm{u}}(j)} |X(k,n)|^2$$

3. sum/mean every 12th band

$$\begin{aligned} \nu(j\%12, n) &= \sum_{o=o_l}^{o_u} \mu(j,n), \\ \nu(n) &= [\nu(0,n), \nu(1,n), \nu(2,n), \ldots, \nu(10,n), \nu(11,n)]^{\mathrm{T}} \end{aligned}$$

pitch chroma
computation 1/2

1. divide spectral representation into **semi-tone bands**
2. compute **mean per band**

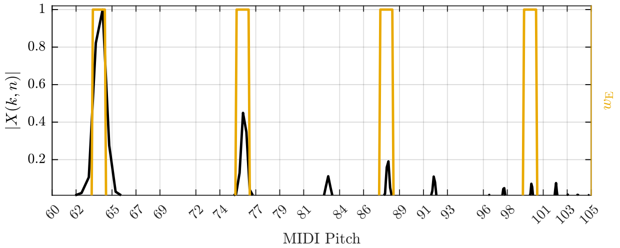
$$\mu(j,n) = \frac{1}{k_\mathrm{u}(j) - k_\mathrm{l}(j) + 1} \sum_{k=k_\mathrm{l}(j)}^{k_\mathrm{u}(j)} |X(k,n)|^2$$

3. sum/mean every 12th band

$$\begin{aligned}
\nu(j\%12, n) &= \sum_{o=o_l}^{o_u} \mu(j,n), \\
\boldsymbol{\nu}(n) &= [\nu(0,n), \nu(1,n), \nu(2,n), \ldots, \nu(10,n), \nu(11,n)]^\mathrm{T}
\end{aligned}$$

pitch chroma

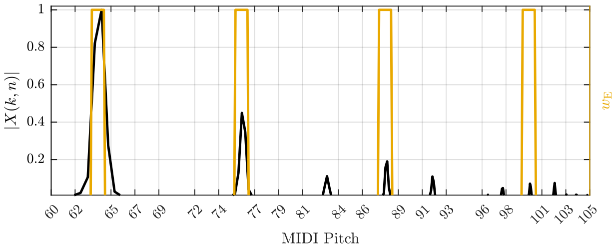
computation 2/2



matlab source: plotPitchChromaGrouping.m

pitch chroma
computation: simple variants

- **STFT**:
  - *weighted* mean of bins (window function)
  - *tonalness preprocessing* (local maxima etc)
- sum of **filterbank** output energies
- **CQT**:
  - sum of bins/peaks
- beat-synchronous chroma

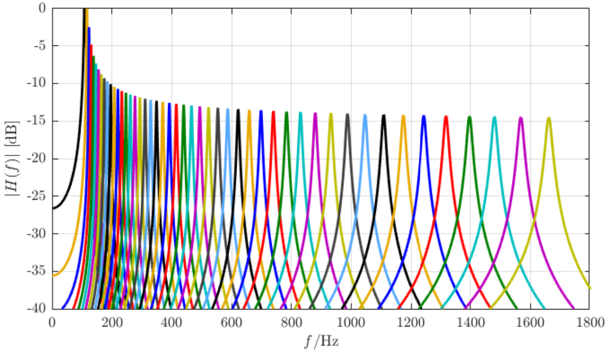


matlab source: plotPitchChromaGrouping.m

pitch chroma

computation: simple variants

- **STFT**:
  - *weighted* mean of bins (window function)
  - *tonalness preprocessing* (local maxima etc)
- sum of **filterbank** output energies
- CQT:
  - sum of bins/peaks
- beat-synchronous chroma



matlab source: plotResonanceFilterBank.m

pitch chroma

computation: simple variants

- **STFT**:
  - *weighted* mean of bins (window function)
  - *tonalness preprocessing* (local maxima etc)
- sum of **filterbank** output energies
- **CQT**:
  - sum of bins/peaks
- beat-synchronous chroma

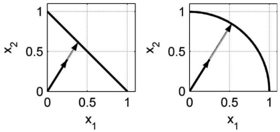
pitch chroma
computation: simple variants

- **STFT**:
  - *weighted* mean of bins (window function)
  - *tonalness preprocessing* (local maxima etc)
- sum of **filterbank** output energies
- **CQT**:
  - sum of bins/peaks
- beat-synchronous chroma

pitch chroma
normalization

- pitch chroma as *distribution*:

$$\sum_{k=0}^{11} \nu(k, n) = 1$$



- pitch chroma as *vector*:

$$\sqrt{\sum_{k=0}^{11} \nu(k, n)^2} = 1$$

- other options:
  - e.g., short-term energy normalization (CENS)

pitch chroma
normalization

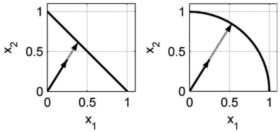
- pitch chroma as *distribution*:

$$\sum_{k=0}^{11} \nu(k,n) = 1$$

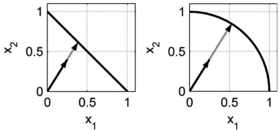
- pitch chroma as *vector*:

$$\sqrt{\sum_{k=0}^{11} \nu(k,n)^2} = 1$$

- other options:
  - e.g., short-term energy normalization (CENS)



pitch chroma
normalization



- pitch chroma as *distribution*:

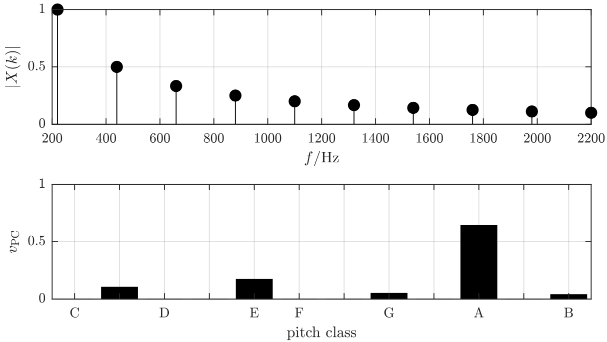
$$\sum_{k=0}^{11} \nu(k,n) = 1$$

- pitch chroma as *vector*:

$$\sqrt{\sum_{k=0}^{11} \nu(k,n)^2} = 1$$

- other options:
  - e.g., short-term energy normalization (CENS)

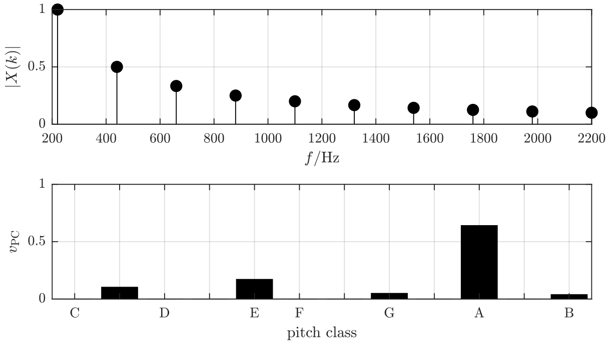
pitch chroma
problem 1: amplitude distortion



- every pitch contains not only fundamental but higher harmonics
  - ⇒ de-emphasize higher frequencies
  - ⇒ build amplitude model
  - ⇒ use multi-pitch detection system

matlab source: plotPitchChromaLeakage.m

pitch chroma
problem 1: amplitude distortion



- every pitch contains not only fundamental but higher harmonics
  - ⇒ de-emphasize higher frequencies
  - ⇒ build amplitude model
  - ⇒ use multi-pitch detection system

matlab source: plotPitchChromaLeakage.m

pitch chroma

problem 2: frequency distortion

- higher harmonics are not "in-tune"

Harmonic	$\lvert\Delta C(f, f_T)\rvert$
$f = f_0$	0
$f = 2 \cdot f_0$	0
$f = 3 \cdot f_0$	1.955
$f = 4 \cdot f_0$	0
$f = 5 \cdot f_0$	13.6863
$f = 6 \cdot f_0$	1.955
$f = 7 \cdot f_0$	31.1741
$\mu_{\lvert\Delta C\rvert}$	6.9672

key detection
introduction

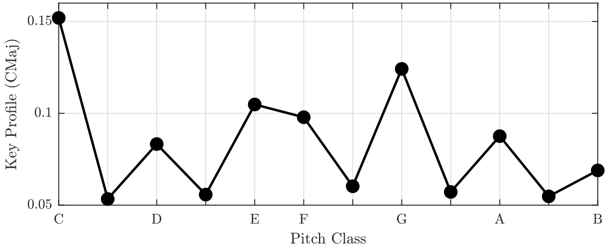
assumption:

- *pitch class distribution* is prototypical for key
  - tonic/root note is tonal center
  - tonal and harmonic relations define importance and occurrence of individual pitch classes
  - different root notes result in simple shift of distribution

key detection

processing steps of simple key detection

1. define reference distribution for specific keys
2. extract average pitch chroma from audio
3. compute distance between template and extracted chroma

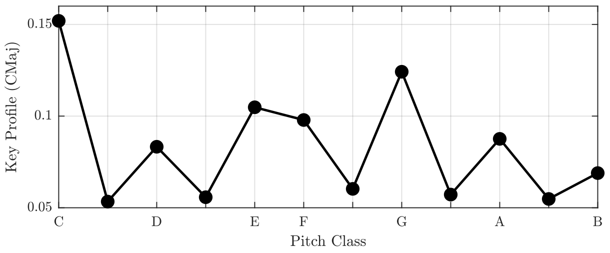


matlab source: plotKeyProfileKrumhansl.m

key detection

processing steps of simple key detection

1. define reference distribution for specific keys
2. extract average pitch chroma from audio
3. compute distance between template and extracted chroma

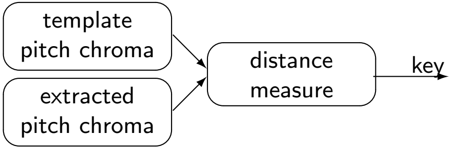


matlab source: plotKeyProfileKrumhansl.m

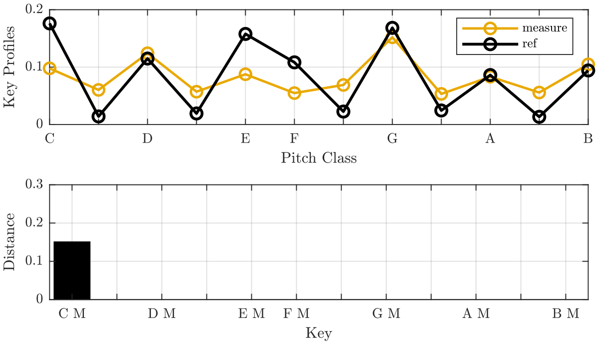
key detection

processing steps of simple key detection

1. define reference distribution for specific keys
2. extract average pitch chroma from audio
3. compute distance between template and extracted chroma



key detection
key template distance animation

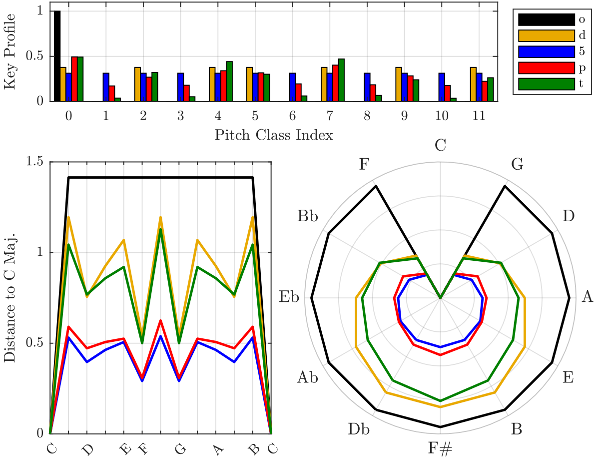


matlab source: matlab/animateKeyDetection.m

key detection
key templates

Georgia Tech | Center for Music Technology
College of Design

- *Orthogonal* ν_{o}: root note is most salient component, other components negligible
    - same distance to all keys
    - no major/minor distinction
- *Diatonic* ν_{d}: all key-inherent pitches weighted equally
    - linear increasing key dist
- *Probe tone Ratings* ν_{p}: derived from perceptual tonal similarity
- *Extracted Key Profiles* ν_{t}: derived from real-world data

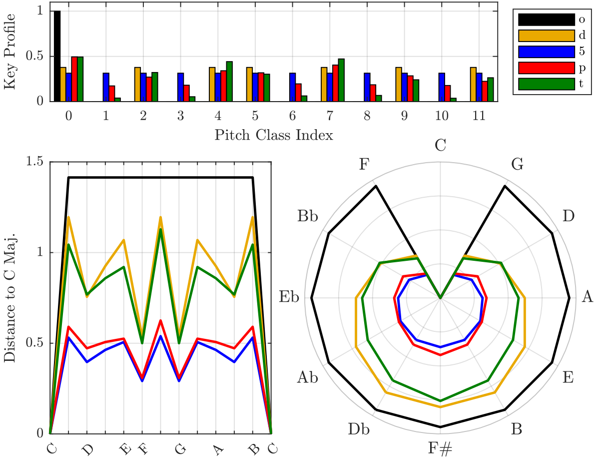


matlab source: plotKeyProfiles.m

key detection

key templates

- *Orthogonal* ν_o: root note is most salient component, other components negligible
  - same distance to all keys
  - no major/minor distinction
- *Diatonic* ν_d: all key-inherent pitches weighted equally
  - linear increasing key dist
- *Probe tone Ratings* ν_p: derived from perceptual tonal similarity
- *Extracted Key Profiles* ν_t: derived from real-world data

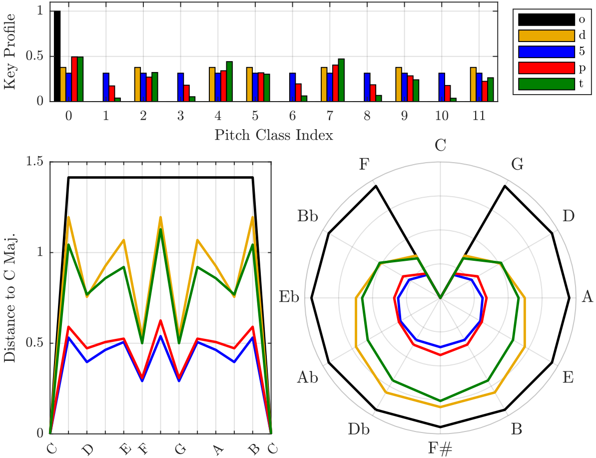


matlab source: plotKeyProfiles.m

key detection

key templates

- *Orthogonal* ν_o: root note is most salient component, other components negligible
  - same distance to all keys
  - no major/minor distinction
- *Diatonic* ν_d: all key-inherent pitches weighted equally
  - linear increasing key dist
- *Probe tone Ratings* ν_p: derived from perceptual tonal similarity
- *Extracted Key Profiles* ν_t: derived from real-world data

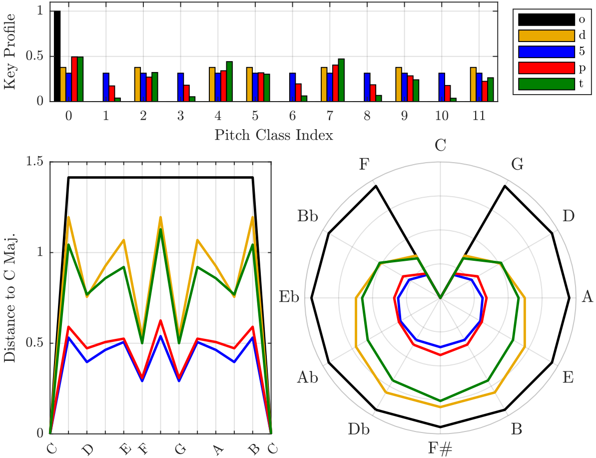


matlab source: plotKeyProfiles.m

key detection

key templates

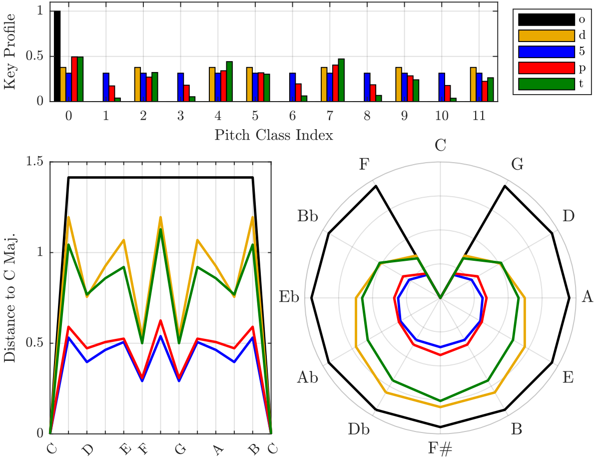
- *Orthogonal* $\boldsymbol{\nu}_{\mathrm{o}}$: root note is most salient component, other components negligible
  - same distance to all keys
  - no major/minor distinction
- *Diatonic* $\boldsymbol{\nu}_{\mathrm{d}}$: all key-inherent pitches weighted equally
  - linear increasing key dist
- *Probe tone Ratings* $\boldsymbol{\nu}_{\mathrm{p}}$: derived from perceptual tonal similarity
- *Extracted Key Profiles* $\boldsymbol{\nu}_{\mathrm{t}}$: derived from real-world data



matlab source: plotKeyProfiles.m

key detection
key templates

- *Orthogonal* $\boldsymbol{\nu}_\mathrm{o}$: root note is most salient component, other components negligible
  - same distance to all keys
  - no major/minor distinction
- *Diatonic* $\boldsymbol{\nu}_\mathrm{d}$: all key-inherent pitches weighted equally
  - linear increasing key dist
- *Probe tone Ratings* $\boldsymbol{\nu}_\mathrm{p}$: derived from perceptual tonal similarity
- *Extracted Key Profiles* $\boldsymbol{\nu}_\mathrm{t}$: derived from real-world data



matlab source: plotKeyProfiles.m

key detection
variants

- **tonalness weight**:
  estimate the tonality/noisiness and weight instantaneous pitch chroma
- **multiple estimations**:
  split piece into regions and estimate key through majority
- **real-time key detection**:
  estimate in sliding window

key detection
variants

- **tonalness weight**:
  estimate the tonality/noisiness and weight instantaneous pitch chroma
- **multiple estimations**:
  split piece into regions and estimate key through majority
- **real-time key detection**:
  estimate in sliding window

key detection
variants

- **tonalness weight**:
  estimate the tonality/noisiness and weight instantaneous pitch chroma
- **multiple estimations**:
  split piece into regions and estimate key through majority
- **real-time key detection**:
  estimate in sliding window

key detection

results & typical errors

- typical errors: related keys
  - Dominant
  - Subdominant
  - Relative
  - Major/Minor

graph from[1]

[1] A. Lerch, "Ein Ansatz zur automatischen Erkennung der Tonart in Musikdateien," in *Proceedings of the VDT International Audio Convention (23. Tonmeistertagung)*, Leipzig, Nov. 2004.

summary
lecture content

- **musical key**
  - set of pitch classes constructing pitched content
  - defined by *tonic* (important center) and *mode* (scale)
- **pitch chroma**
  - reduced 12-dimensional octave-independent pitch representation
  - relatively robust against timbre variation
- **automatic key recognition**
  - standard approach is template-based
  - extracted average pitch chroma is compared with predefined template
  - inverse distance measure indicates key likelihoods