Lecture 4: Chord Recognition

Li Su

Recap: short-time Fourier transform

- Given a discrete-time signal $x(t)$ sampled at a rate f_s. Let window size N samples, hop size H samples, then the short-time Fourier transform $X(n,k)$ is:
- $X(n,k) = \sum_{m=0}^{N-1} x(m+nH)h(m)e^{-\frac{j2\pi km}{n}}$

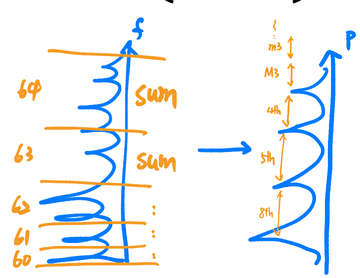
頻率刻度 $\Delta f = \frac{f_s}{N}$

- k: frequency index, $f(k) := \frac{kf_s}{N}$ (linear scale!)
- n: time index , $t(n) := \frac{nH}{f_s}$
- Spectrogram: $|X(n,k)|^2$

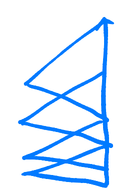
Li Su : hop size = 10% or 25% window size
最高不超過 50%，即 window 要有重疊
Window function : Hann , Hamming

MIDI number

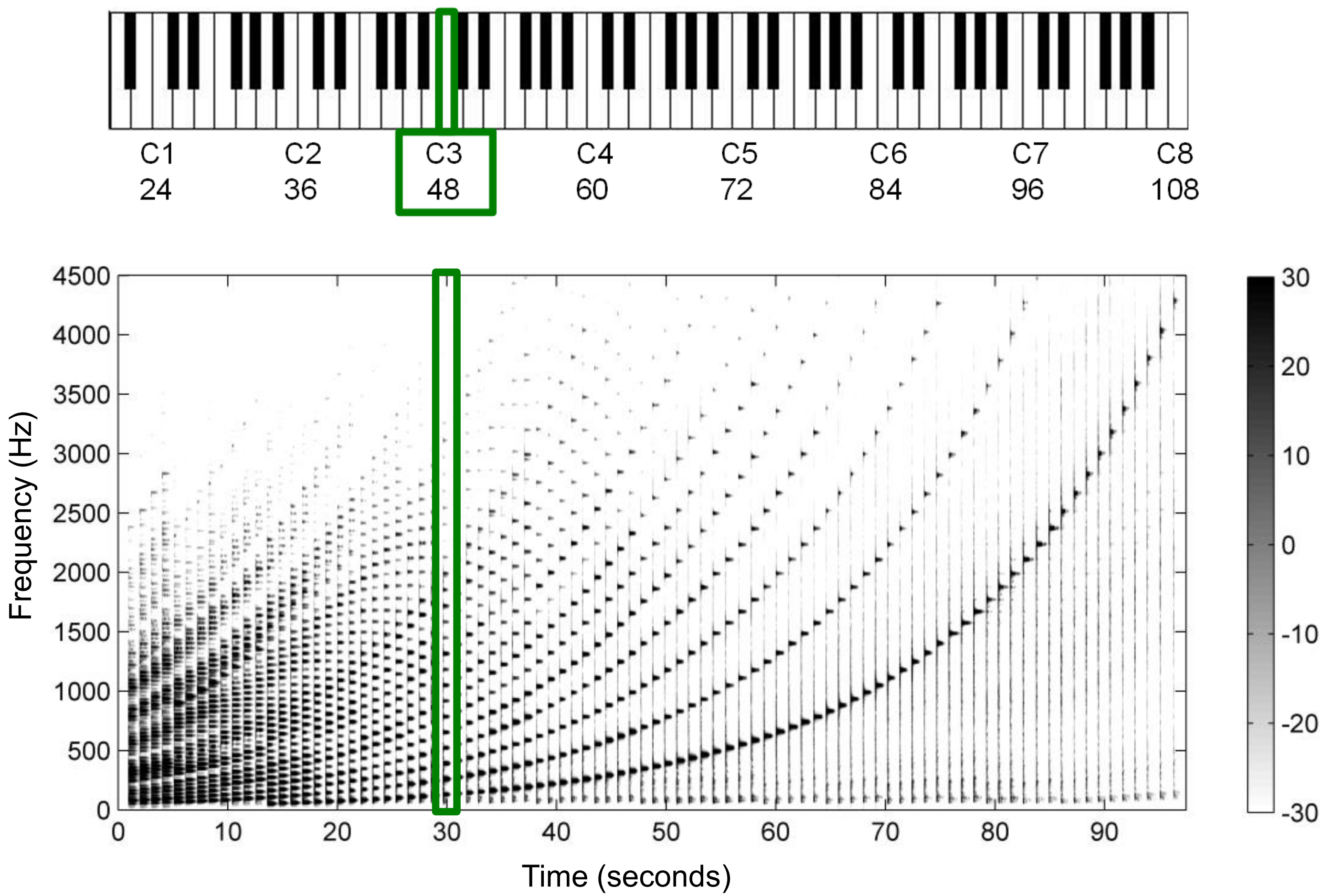
$f = 440 \times 2^{\frac{p-69}{12}}$ A0



即 linear frequency
轉 log frequency (pitch)
亦可用 filter bank 做 weighted sum ex:



Spectrogram 時頻圖



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 3, Springer 2015

Log-frequency spectrogram

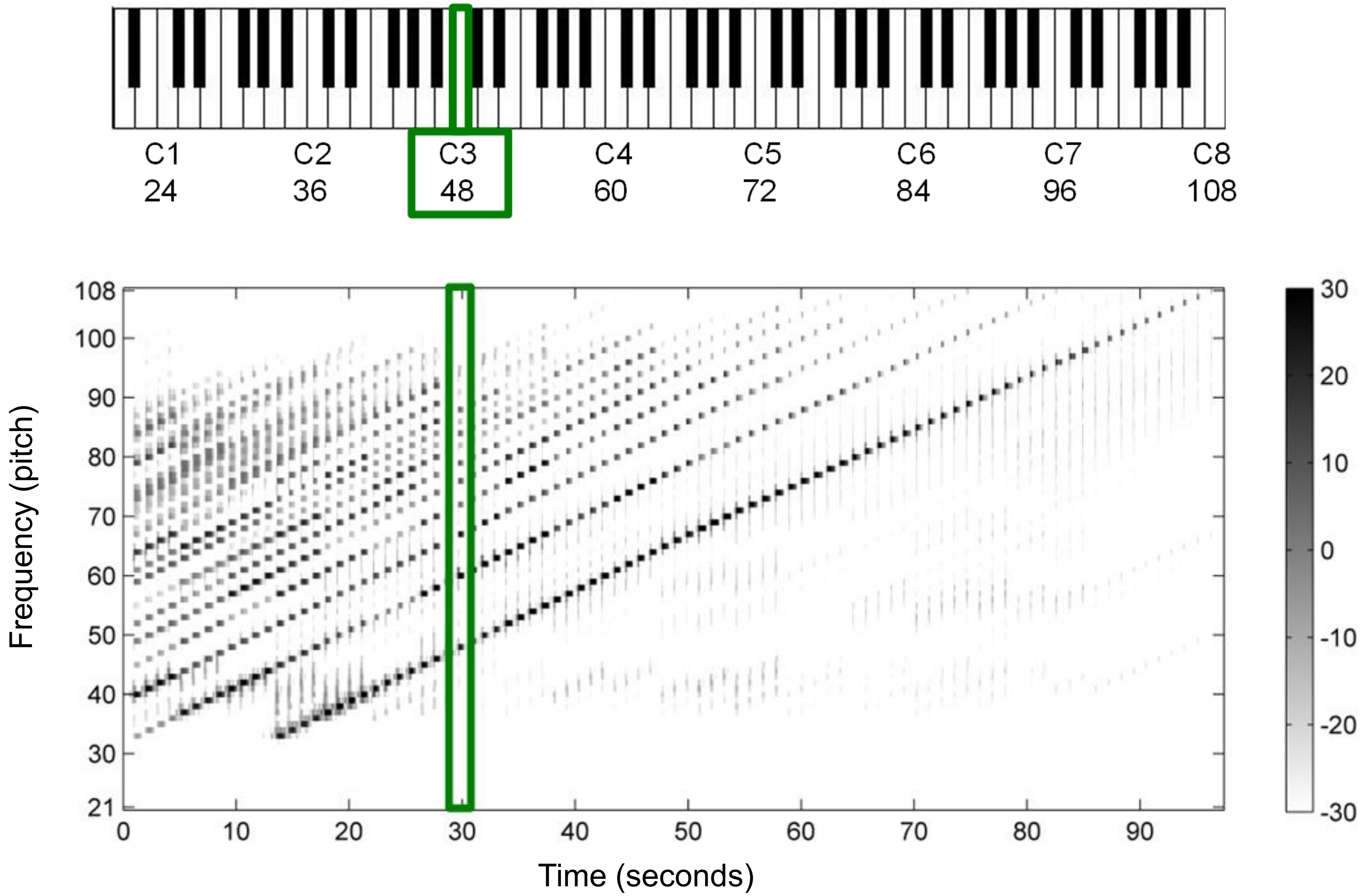
- Basic idea: merge the bins (usually larger than one) within ± 0.5 semitones from each pitch center frequency into one bin
- $F_{\text{pitch}}(p) = 440 \times 2^{\frac{p-69}{12}}$
- $P(p) = \{k: F_{\text{pitch}}(p - 0.5) < k < F_{\text{pitch}}(p + 0.5)\}$
- The log-frequency spectrogram:
- $Y(n, p) = \sum_{k \in P(p)} |X(n, k)|^2$

Example: pitch scale and frequency

Note	p	$F_{\text{pitch}}(p)$	$F_{\text{pitch}}(p-0.5)$	$F_{\text{pitch}}(p+0.5)$	$\text{BW}(p)$
C4	60	261.63	254.18	269.29	15.11
C$^\sharp$4	61	277.18	269.29	285.30	16.01
D4	62	293.66	285.30	302.27	16.97
D$^\sharp$4	63	311.13	302.27	320.24	17.97
E4	64	329.63	320.24	339.29	19.04
F4	65	349.23	339.29	359.46	20.18
F$^\sharp$4	66	369.99	359.46	380.84	21.37
G4	67	392.00	380.84	403.48	22.65
G$^\sharp$4	68	415.30	403.48	427.47	23.99
A4	69	440.00	427.47	452.89	25.41
A$^\sharp$4	70	466.16	452.89	479.82	26.93
B4	71	493.88	479.82	508.36	28.53
C5	72	523.25	508.36	538.58	30.23

From: M. Mueller, *Fundamentals of Music Processing*, Chapter 3, Springer 2015

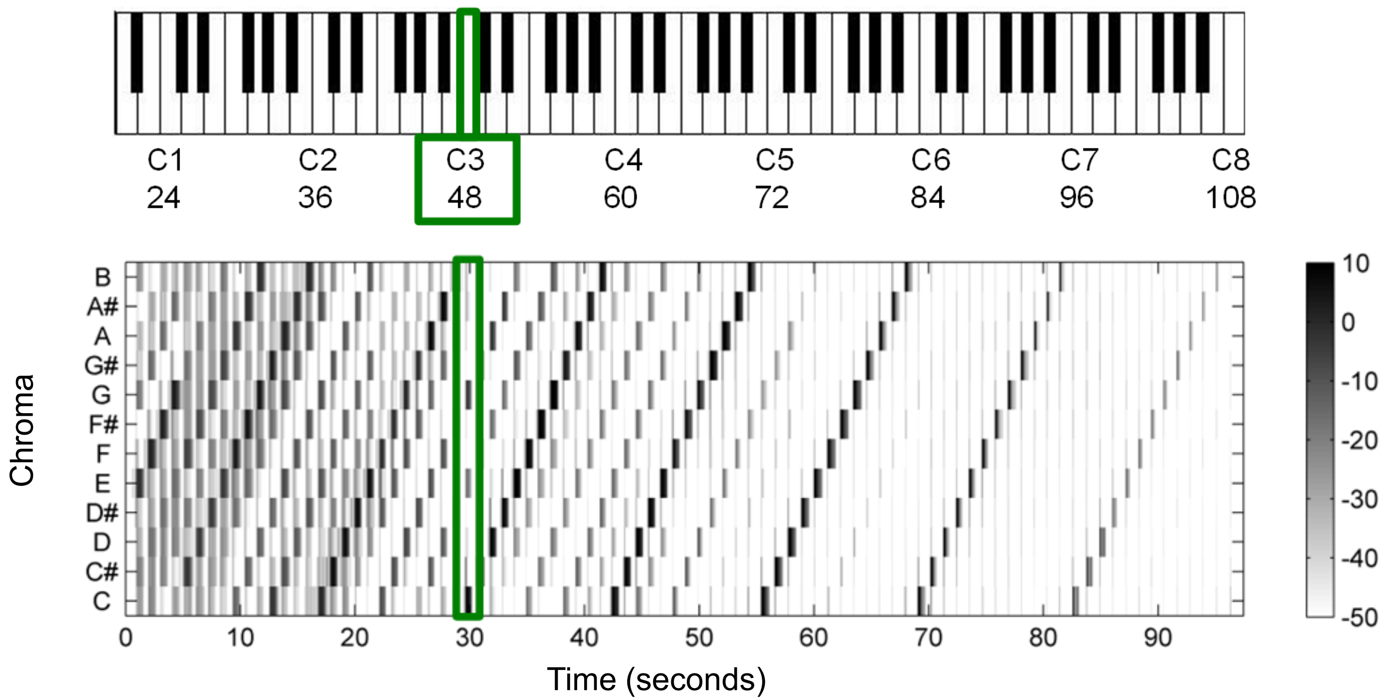
Conversion into pitch scale



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 3, Springer 2015

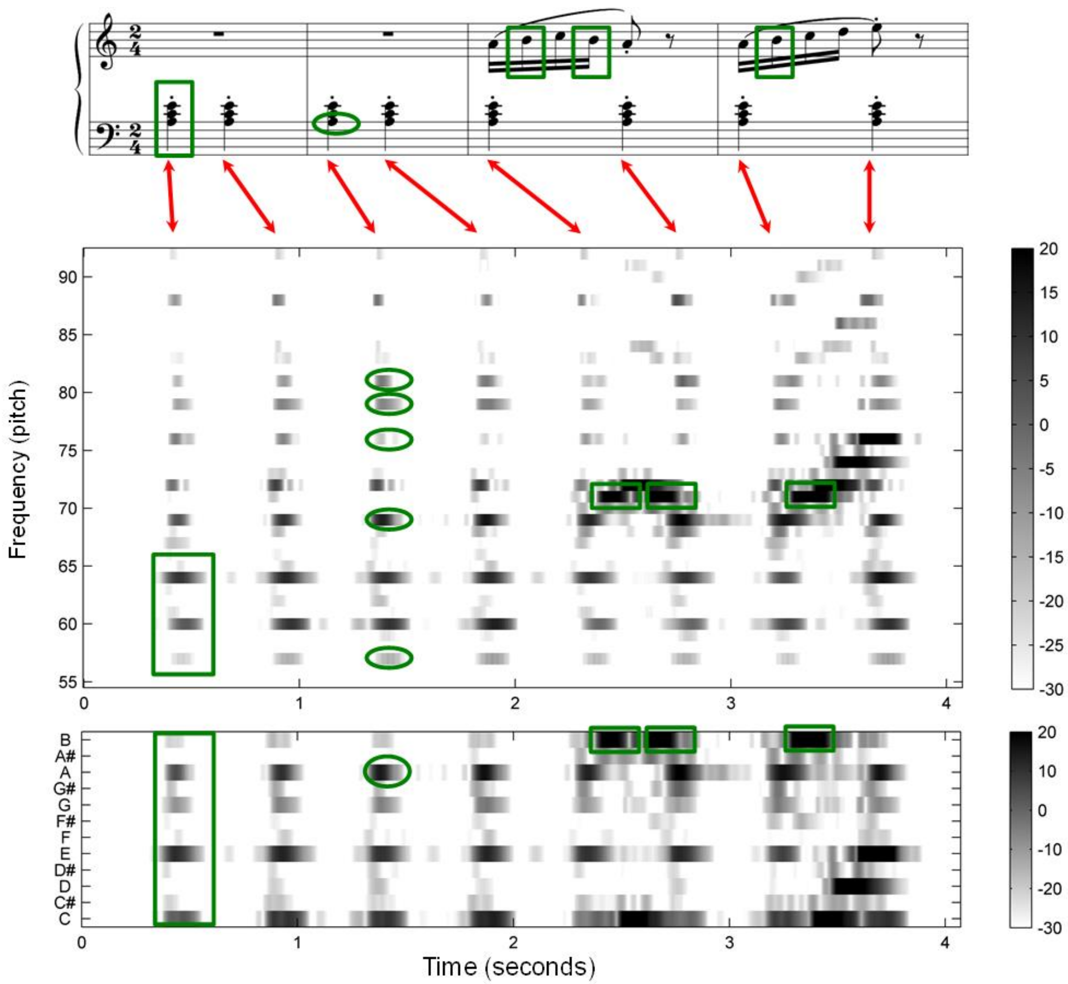
Chromagram

- Basic idea: merge all the bins with the same note name of all octaves together 不區分8度音，把每隔12格的數字加起來
- 12-dimensional vector for each time index



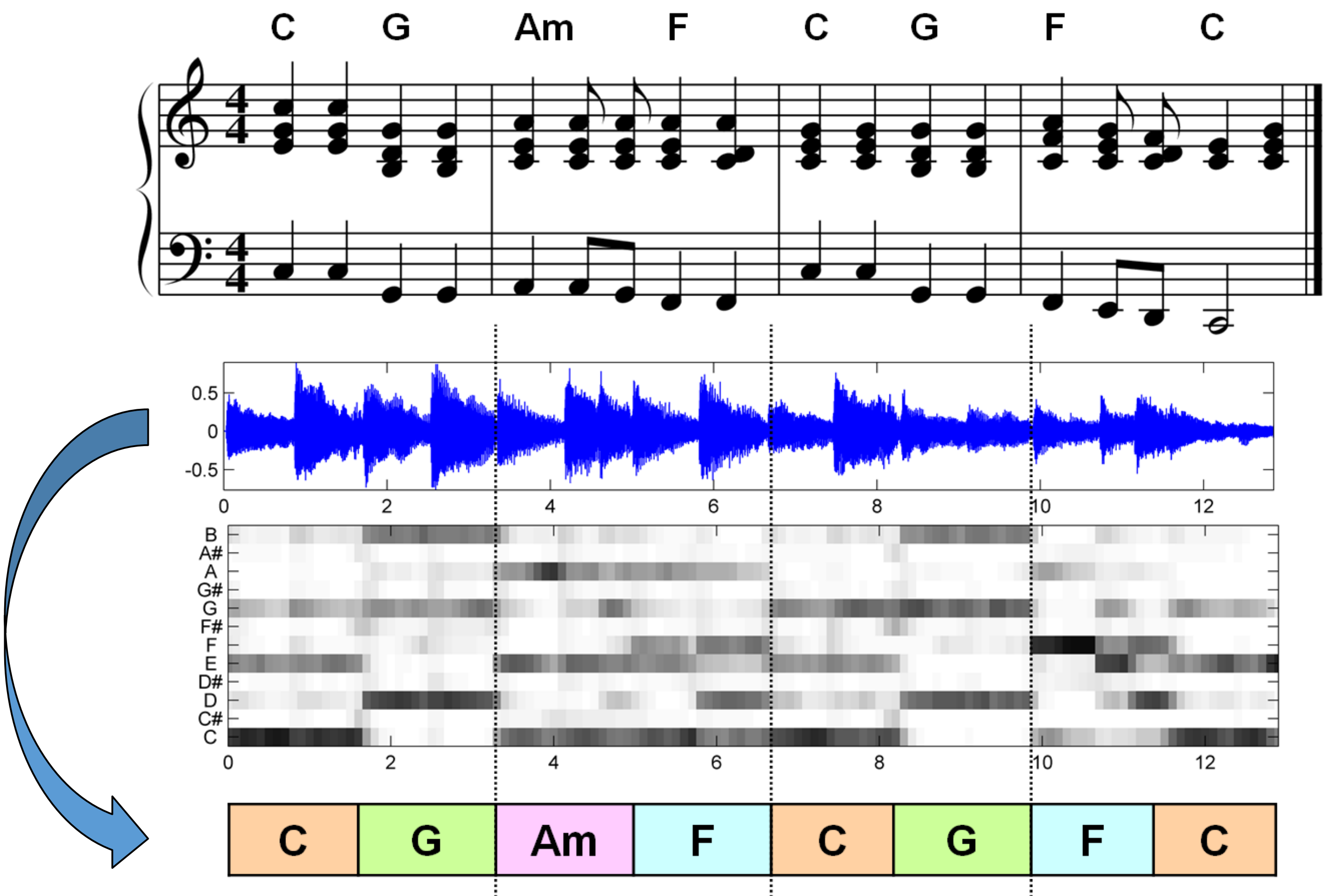
From: M. Mueller, *Fundamentals of Music Processing*, Chapter 3, Springer 2015

An example



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 3, Springer 2015

Our goal: chord recognition



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

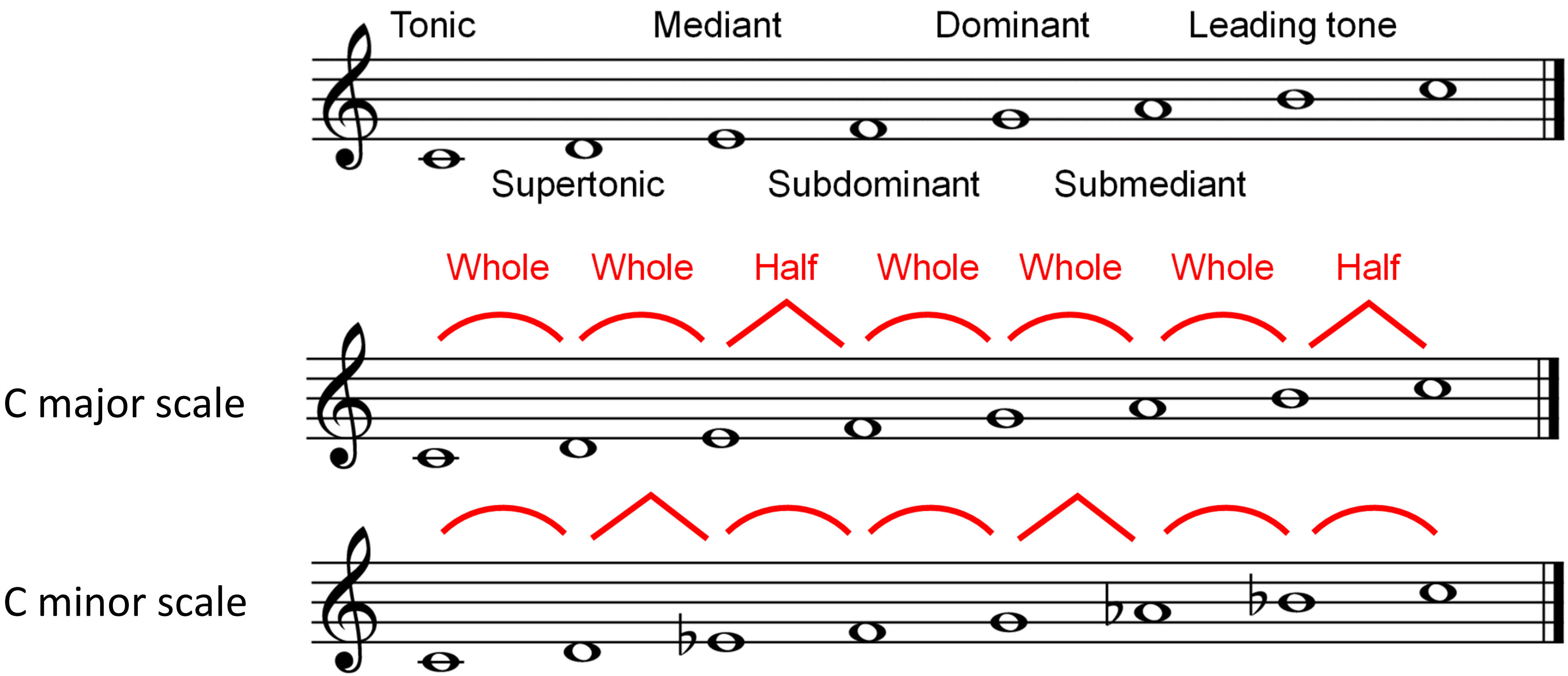
The chromatic scale

- One semitone: a half step; two semitones: a whole step
- One semitone = 100 cents = frequency ratio of $2^{\frac{1}{12}} \approx 1.059$
- Enharmonic equivalence: C# = Db (we only discuss equal temperament here)

From: M. Mueller, *Fundamentals of Music Processing*, Chapter 1, Springer 2015

Diatonic scales

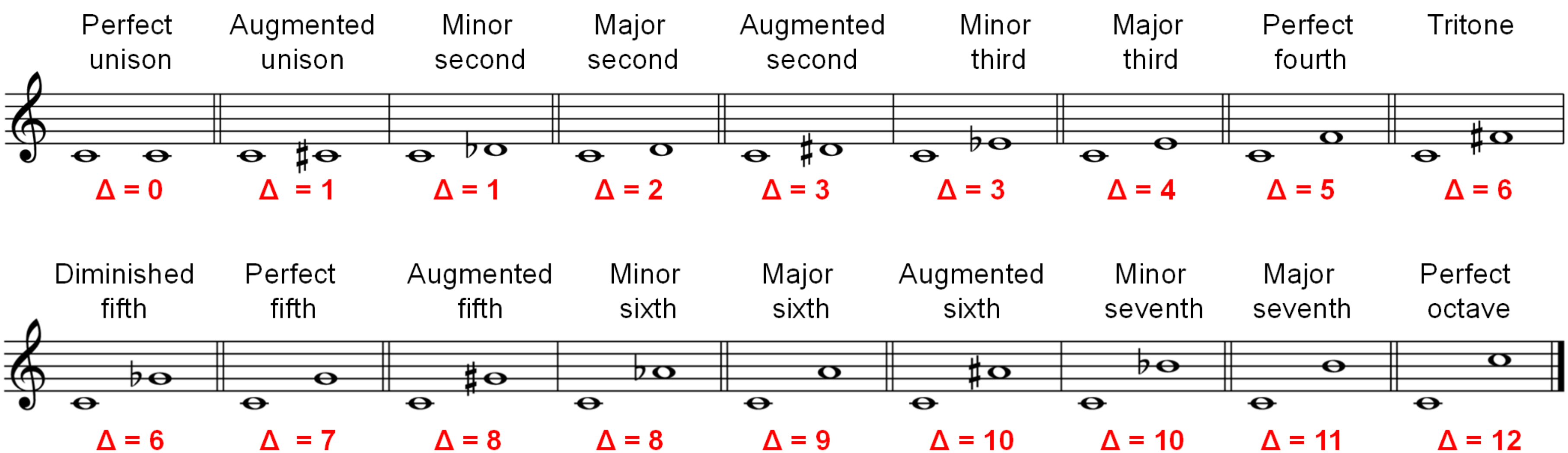
- “Tonal music”:
  - There is one (and only one) “home” pitch, called tonal center
  - Use the 7-note diatonic scales with the following relationship:



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Intervals

- Abundant information here
  - Relations of note (musical approach)
  - Relations of harmonic frequencies (physical approach)
  - Relation of geometry (mathematical approach)
- Twelve-tone equal-tempered scale



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 1, Springer 2015

Intervals and their ratios of frequency

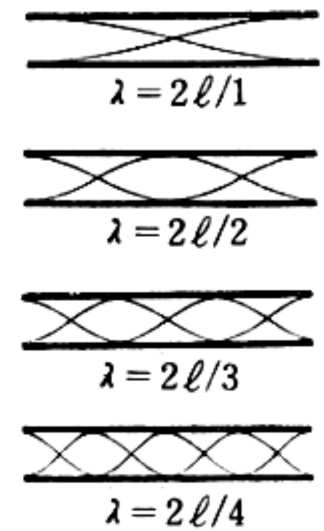
- Pythagorean tuning: only 2:3
- Just intonation: evolution from some basic ratios
- Equal-tempered: exponential scale

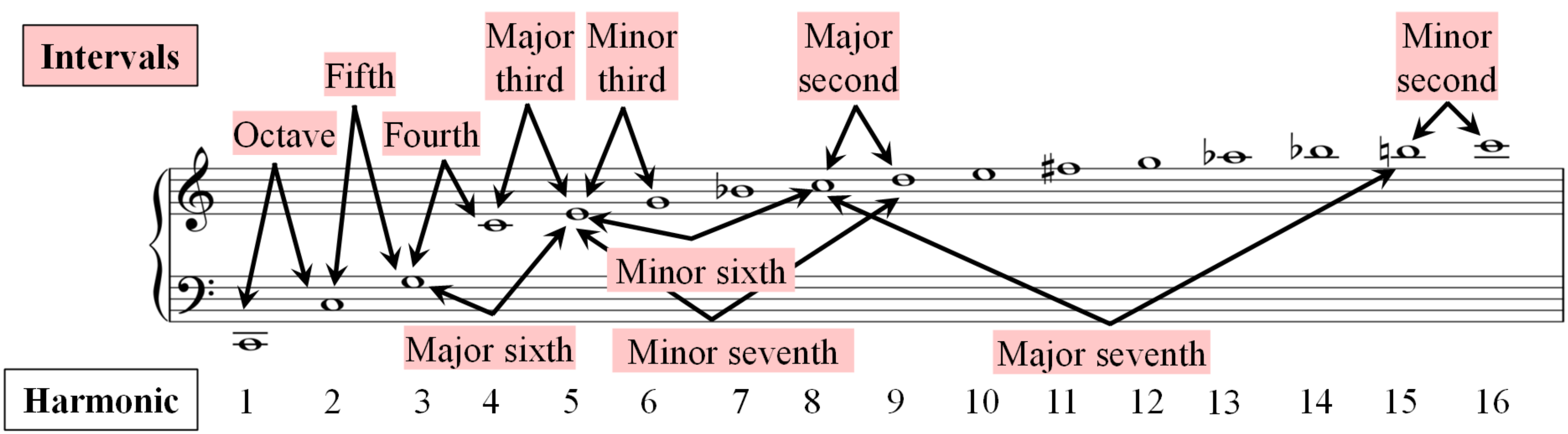
Δ	Interval name	Interval	JI ratio	Pyt. ratio	E-T
0	(Perfect) unison	C4 – C4	1:1	1:1	$1:1$
1	Minor second	C4 – D♭4	15:16	$3^5:2^8$	$1:2^{1/12}$
2	Major second	C4 – D4	8:9	$2^3:3^2$	$1:2^{2/12}$
3	Minor third	C4 – E♭4	5:6	$3^3:2^5$	$1:2^{3/12}$
4	Major third	C4 – E4	4:5	$2^6:3^4$	$1:2^{4/12}$
5	(Perfect) fourth	C4 – F4	3:4	$3:2^2$	$1:2^{5/12}$
6	Tritone	C4 – F♯4	32:45	$2^9:3^6$ or $3^6:2^{10}$	$1:2^{6/12}$
7	(Perfect) fifth	C4 – G4	2:3	2:3	$1:2^{7/12}$
8	Minor sixth	C4 – A♭4	5:8	$3^4:2^7$	$1:2^{8/12}$
9	Major sixth	C4 – A4	3:5	$2^4:3^3$	$1:2^{9/12}$
10	Minor seventh	C4 – B♭4	5:9	$3^2:2^4$	$1:2^{10/12}$
11	Major seventh	C4 – B4	8:15	$2^7:3^5$	$1:2^{11/12}$
12	(Perfect) octave	C4 – C5	1:2	1:2	$1:2$

Modified from: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Related to the harmonic series

- Natural harmonics corresponds to musical pitch
- Utilized by many brass/woodwind instruments





From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Basic chords: triad

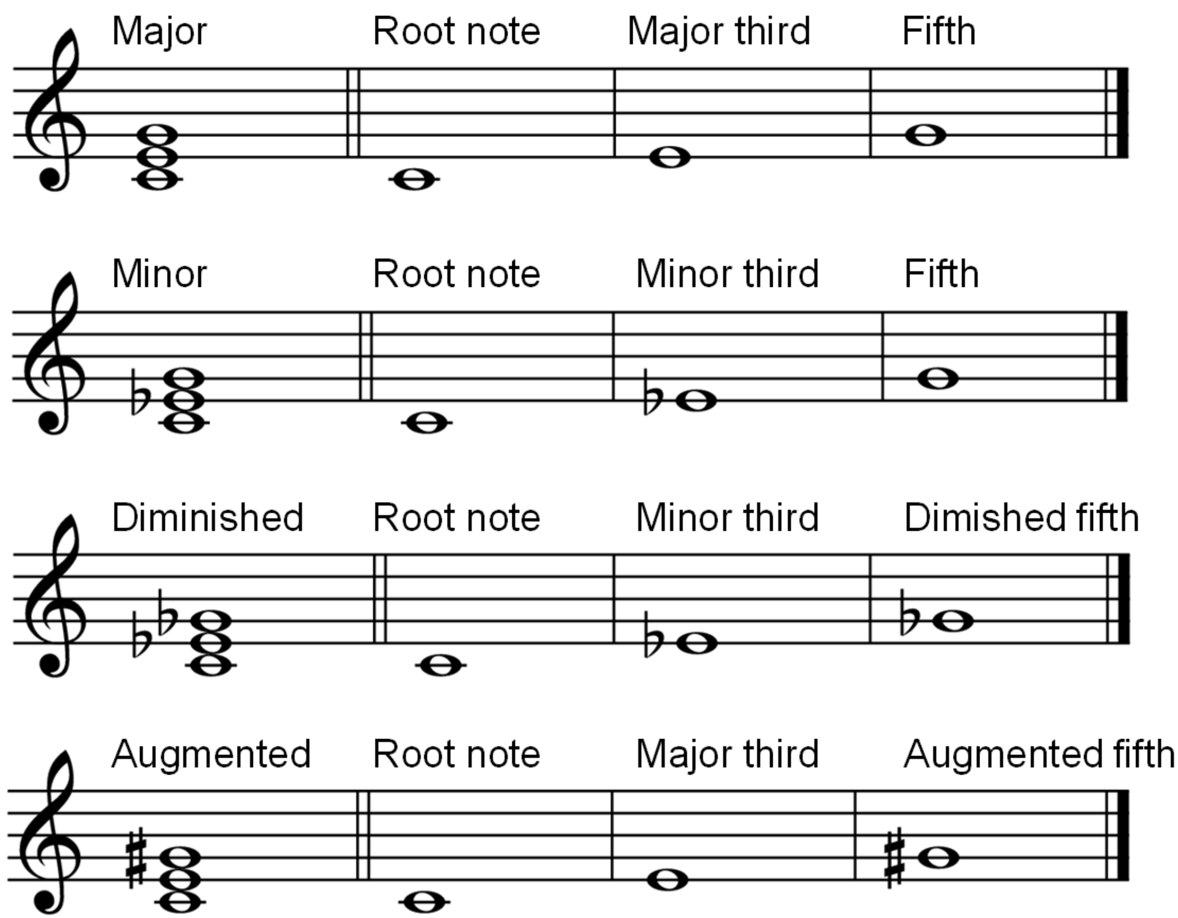
- 1+3+5

Major triad

Minor triad

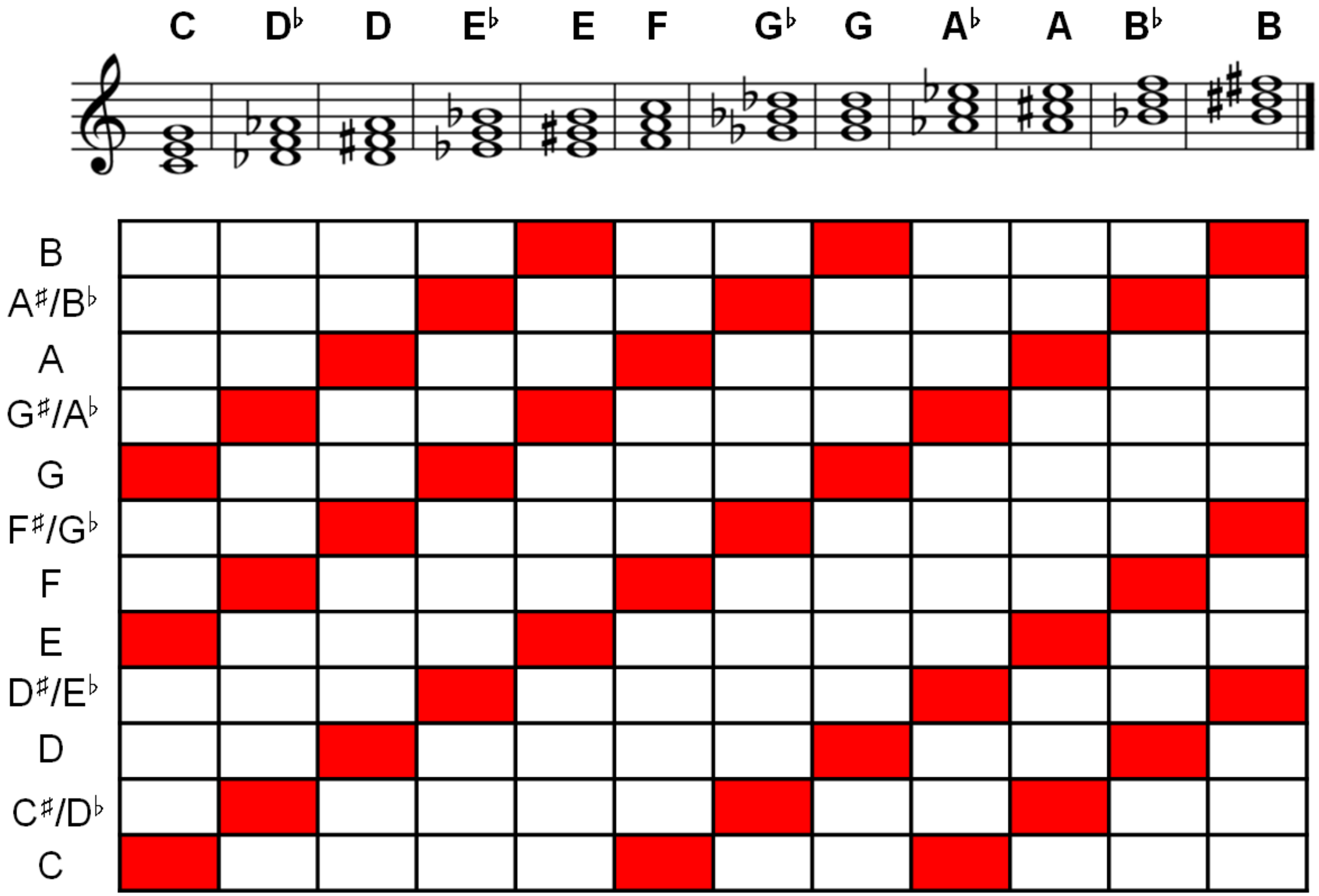
Diminished triad

Augmented triad



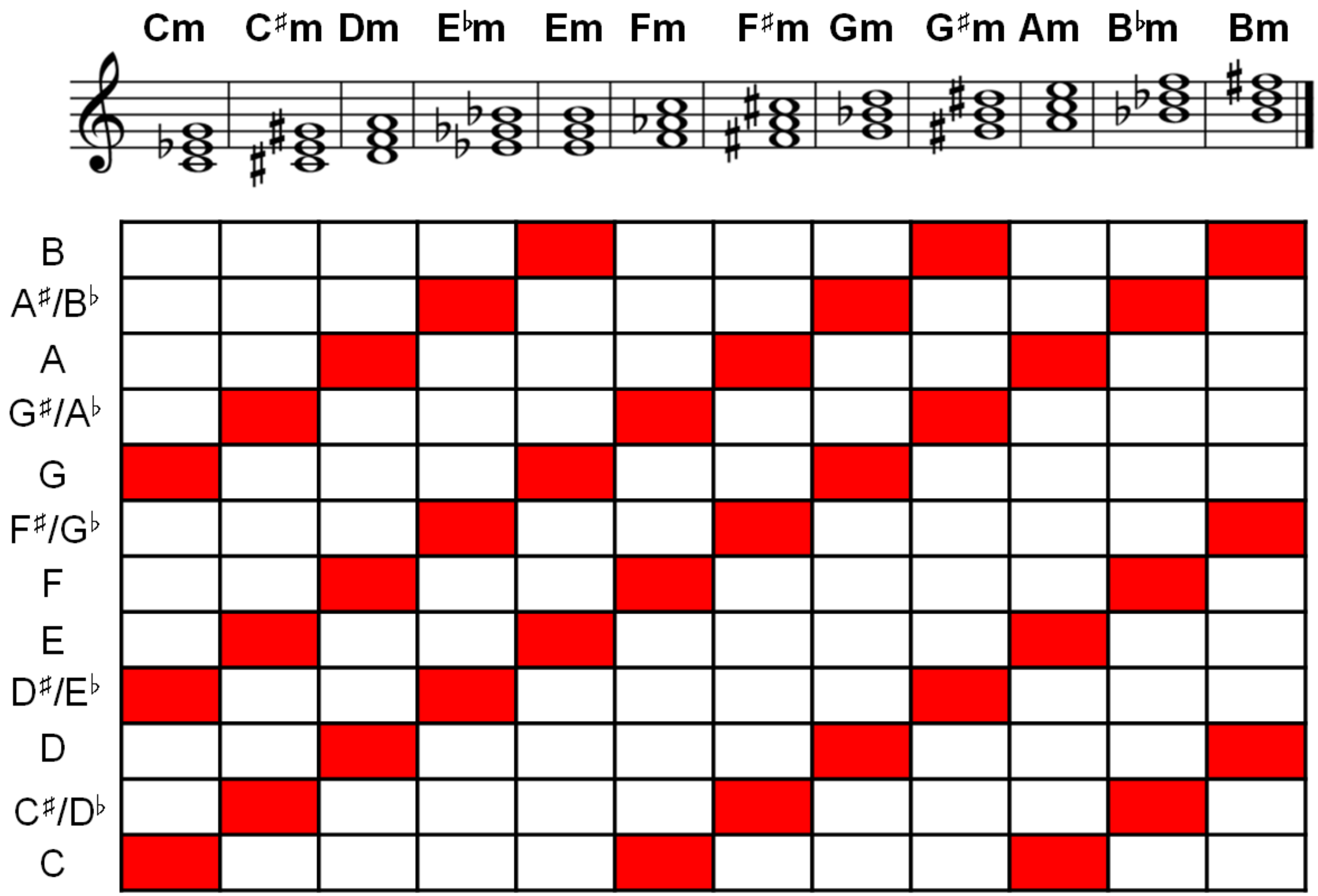
From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Ideal chromagrams of major triads



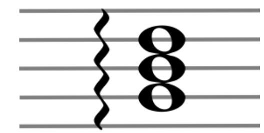
From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Ideal chromagrams of minor triads

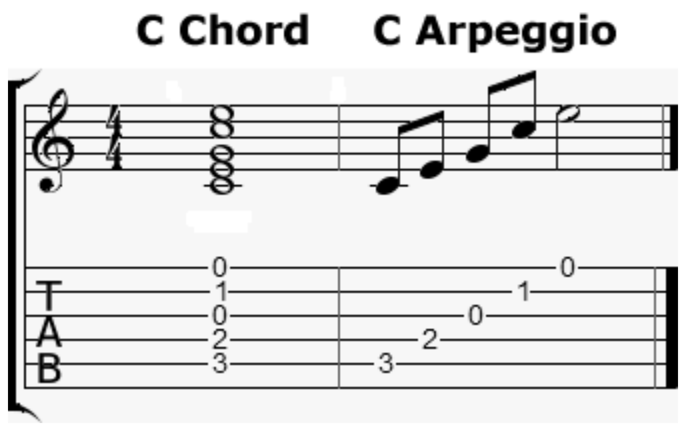


From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

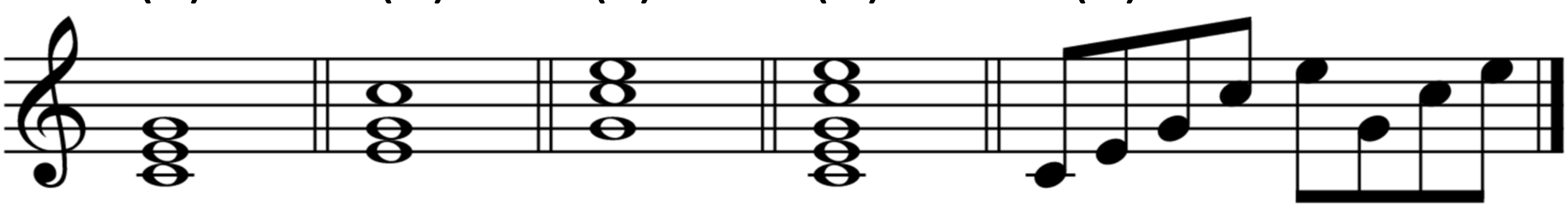
Inversion of chords



- (a) Root position
- (b) First inversion
- (c) Second inversion
- (d) Octave doubling
- (e) Broken chord



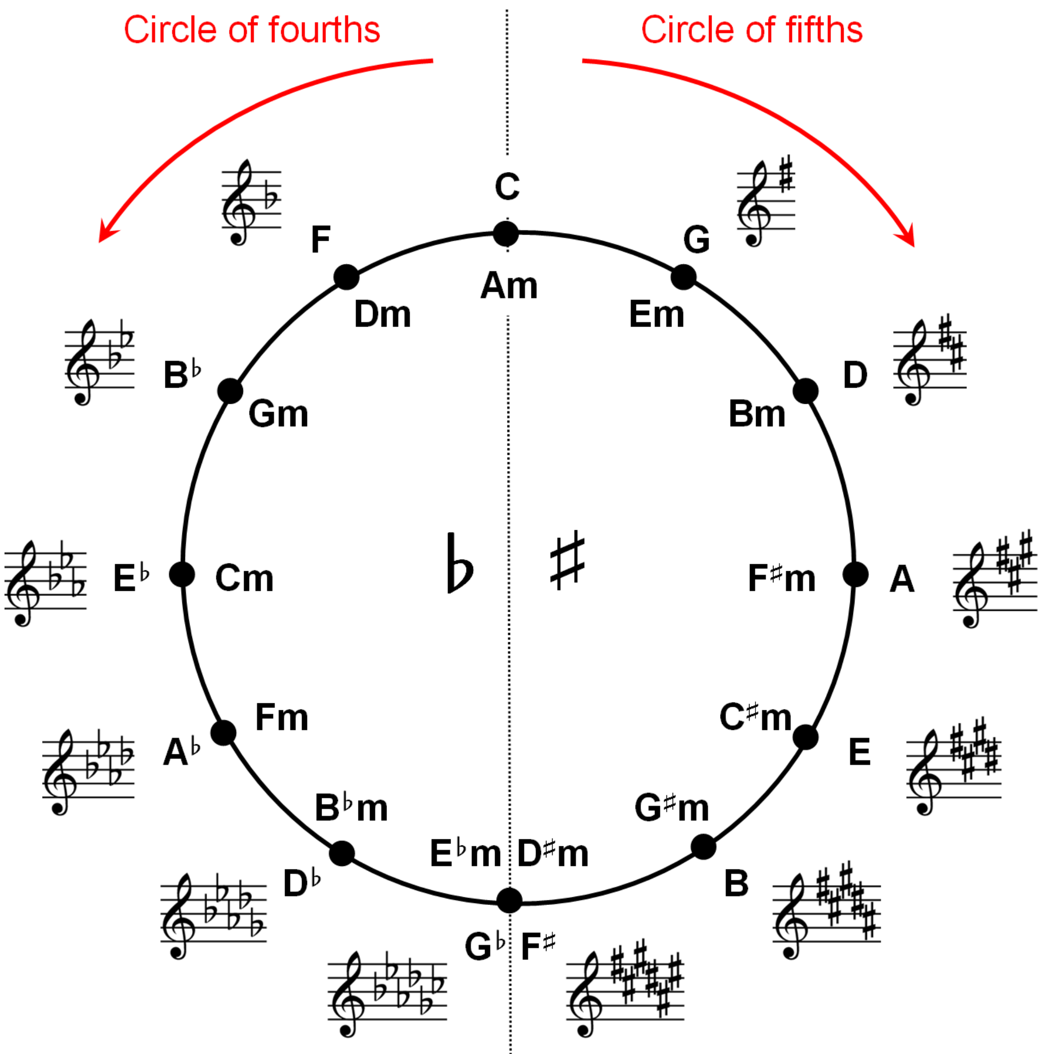
- (a) (b) (c) (d) (e)



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

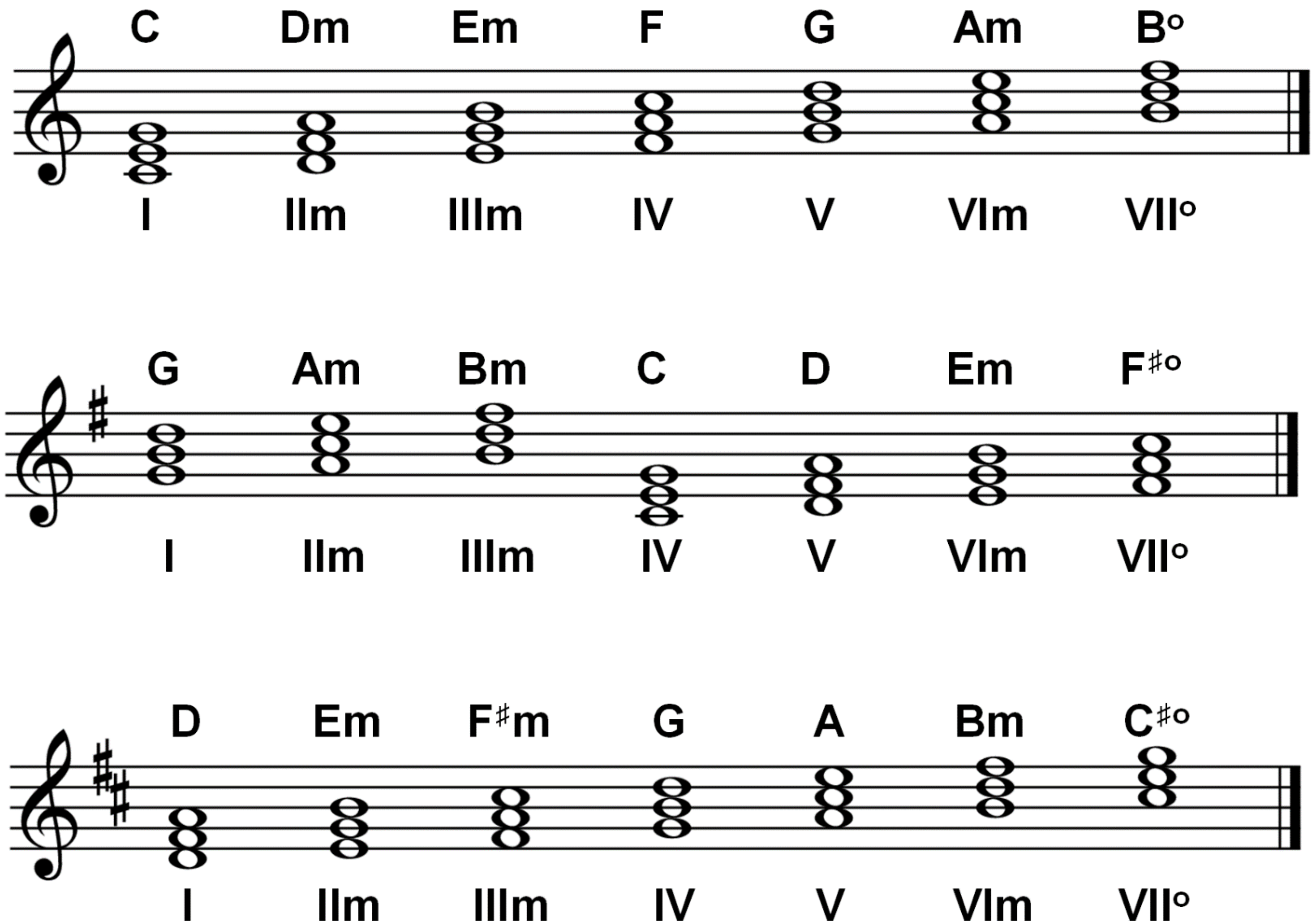
Circle of fifths

- The diatonic scale can be obtained from a chain of 6 successive perfect fifths
- Transposition
- The circle of fifths generates different keys



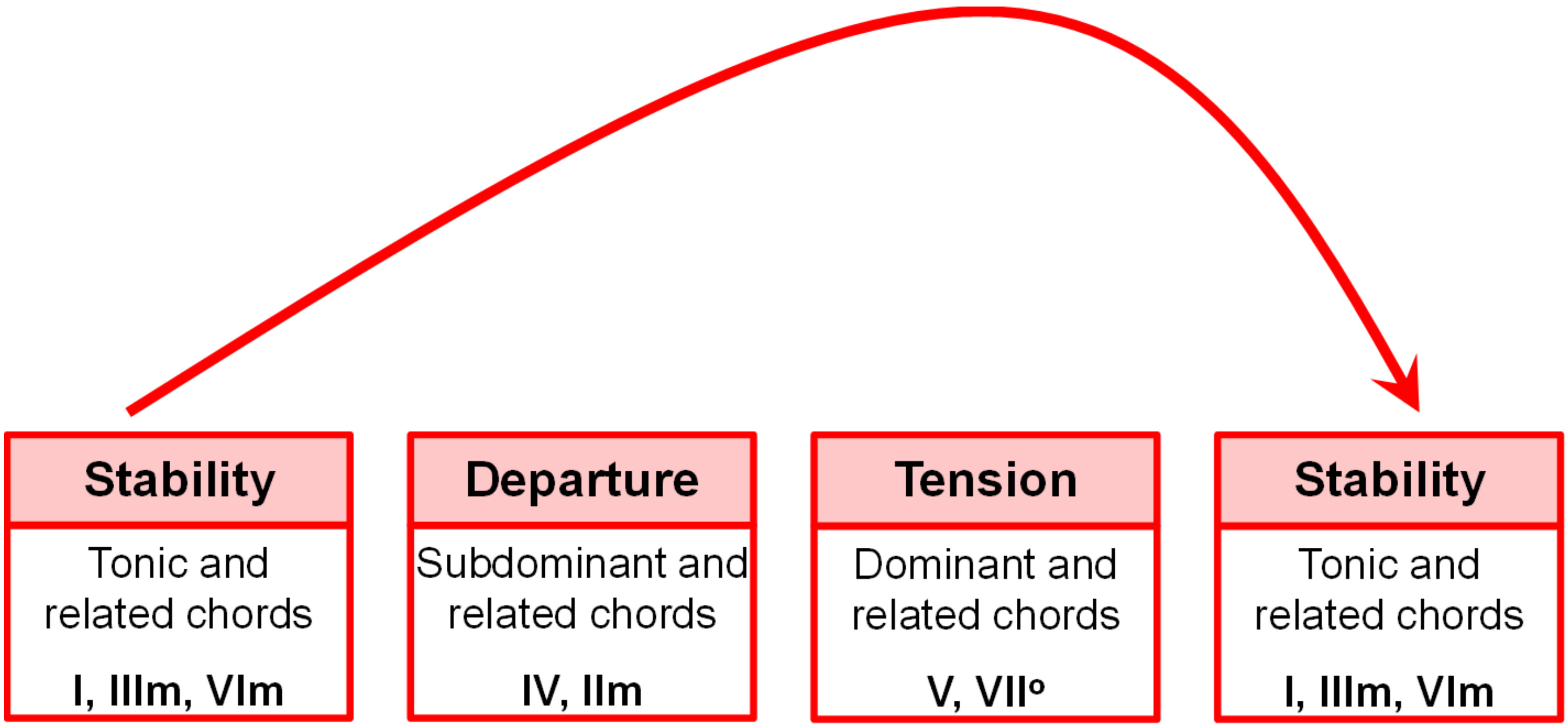
From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Functional harmony



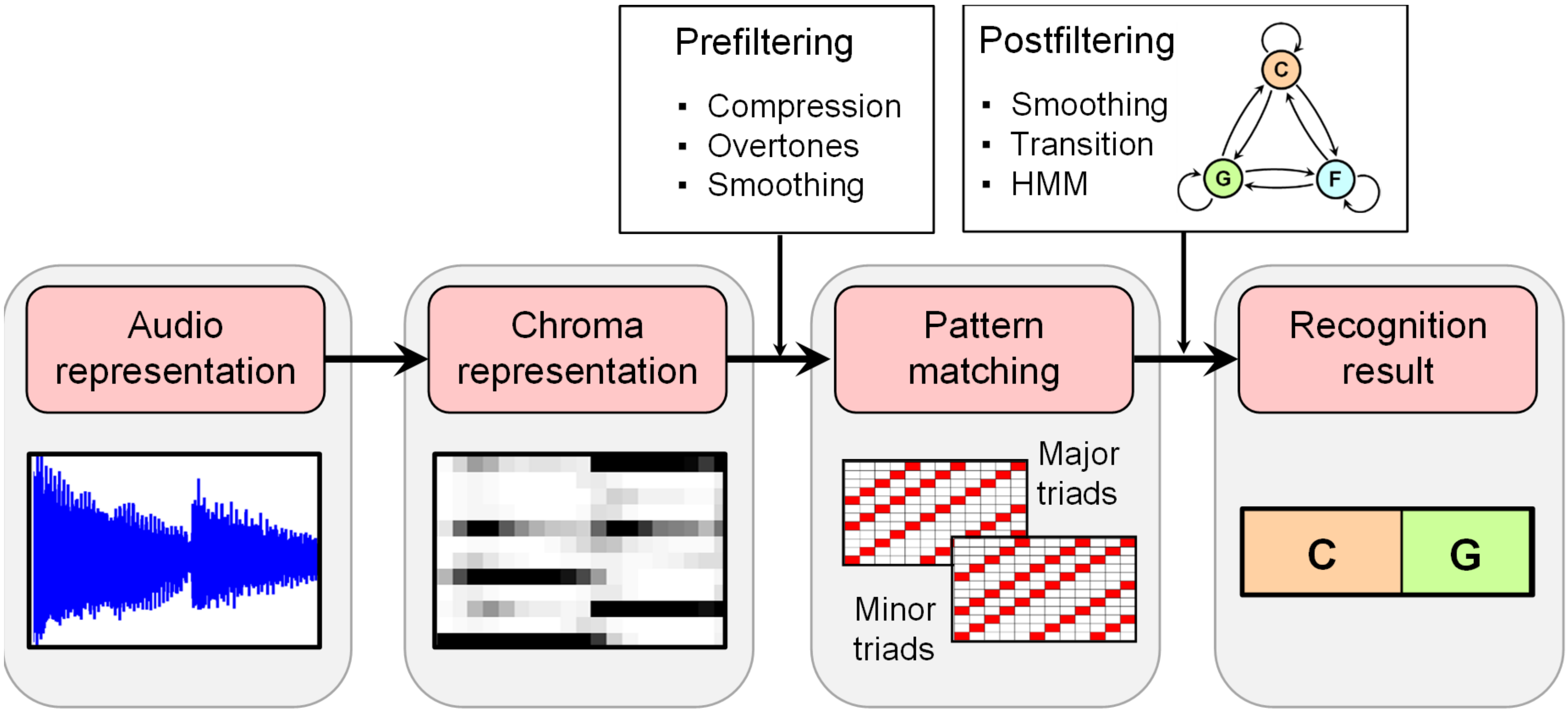
From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Chord progressions



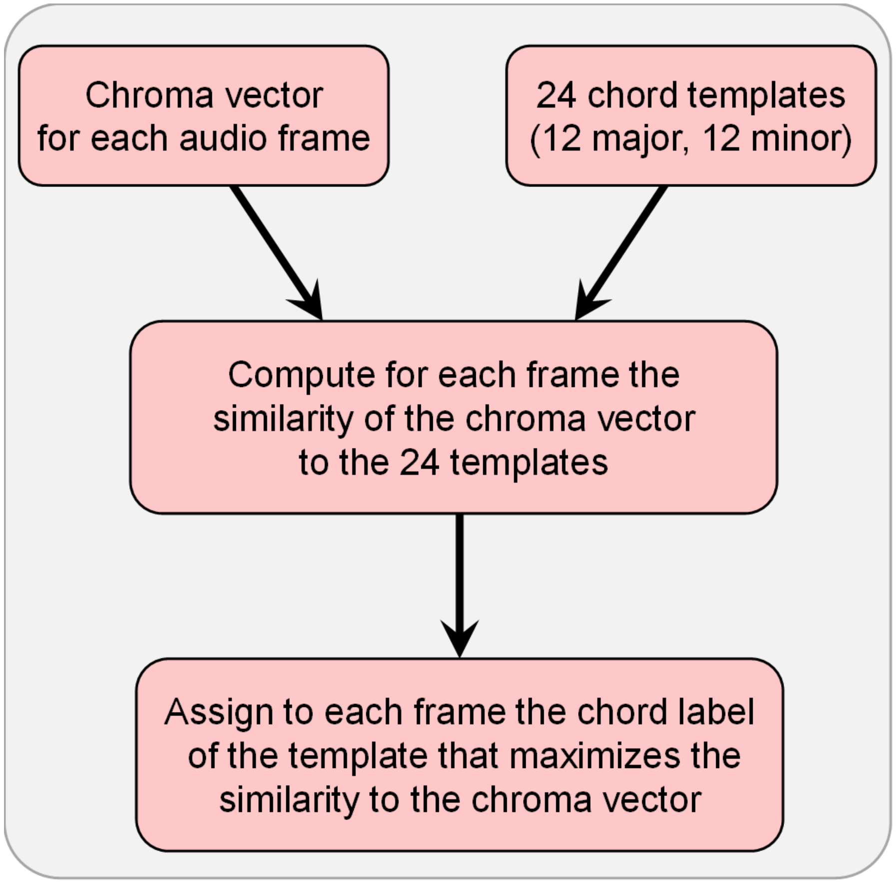
From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Chord recognition system



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Template-based chord recognition



Template

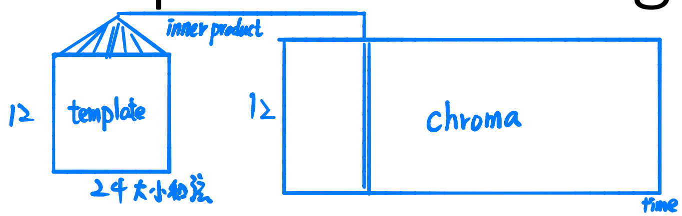
	C	C$^\sharp$	D	…	Cm	C$^\sharp$m	Dm	…
B	0	0	0	…	0	0	0	…
A$^\sharp$	0	0	0	…	0	0	0	…
A	0	0	1	…	0	0	1	…
G$^\sharp$	0	1	0	…	0	1	0	…
G	1	0	0	…	1	0	0	…
F$^\sharp$	0	0	1	…	0	0	0	…
F	0	1	0	…	0	0	1	…
E	1	0	0	…	0	1	0	…
D$^\sharp$	0	0	0	…	1	0	0	…
D	0	0	1	…	0	0	1	…
C$^\sharp$	0	1	0	…	0	1	0	…
C	1	0	0	…	1	0	0	…

From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

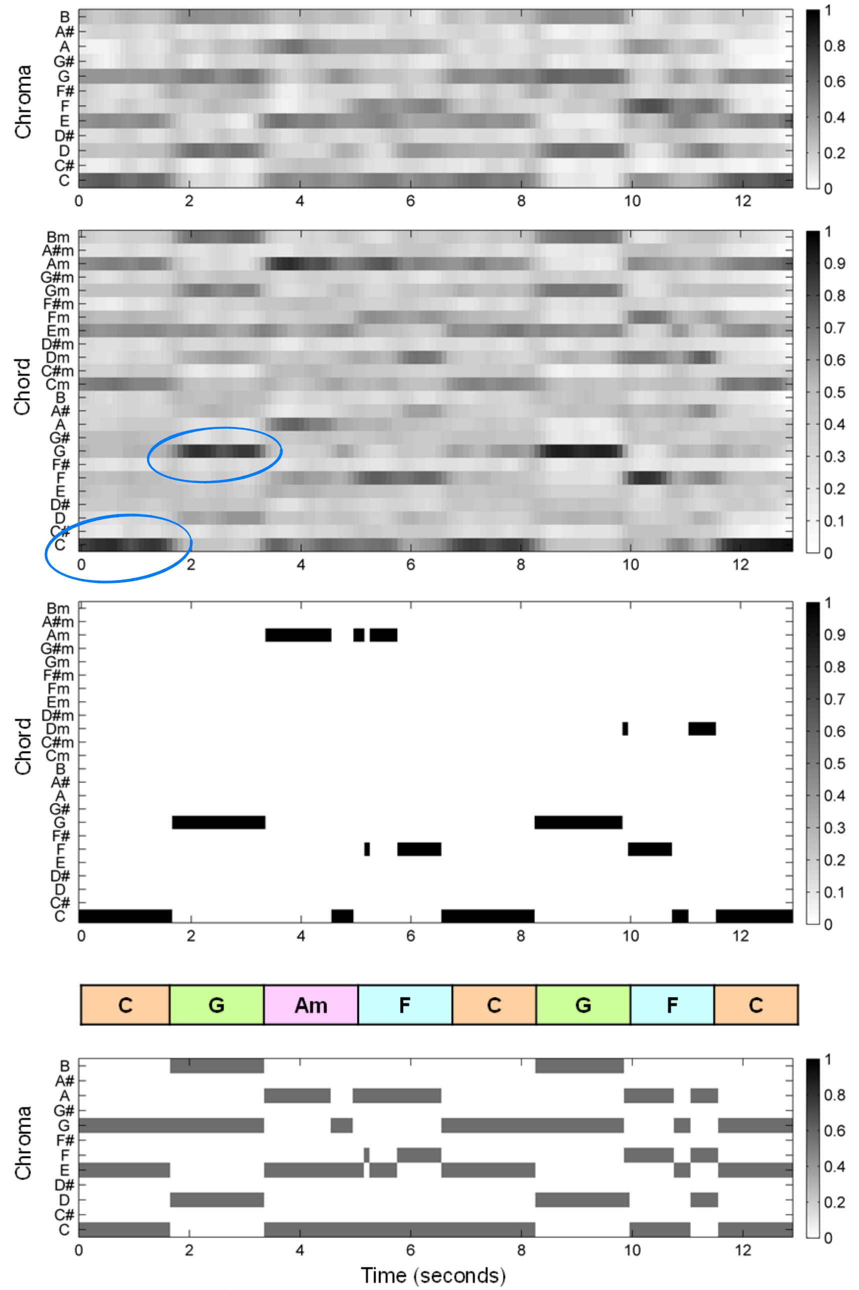
Template-based chord recognition

- Chord labels: 12 major triads and 12 minor triads
- $\Lambda \coloneqq \{\mathbf{C}, \mathbf{C\#}, \mathbf{D}, \dots, \mathbf{B}, \mathbf{Cm}, \mathbf{C\#m}, \mathbf{Dm}, \dots, \mathbf{Bm}\}$
- A chroma feature: a 12-dimensional vector $\boldsymbol{x}_n = [x(0), x(1), x(2), \dots, x(11)]$ at time index n
- Chroma templates:
  - $\boldsymbol{t}_{\mathbf{C}} = (1,0,0,0,1,0,0,1,0,0,0,0)$
  - $\boldsymbol{t}_{\mathbf{A_m}} = (1,0,0,0,1,0,0,0,0,1,0,0)$
  - $\boldsymbol{t}_{\mathbf{G}} = (0,0,1,0,0,0,0,1,0,0,0,1)$ ……
- Find the chord template that maximize the similarity measure and assign the chord label

Template matching

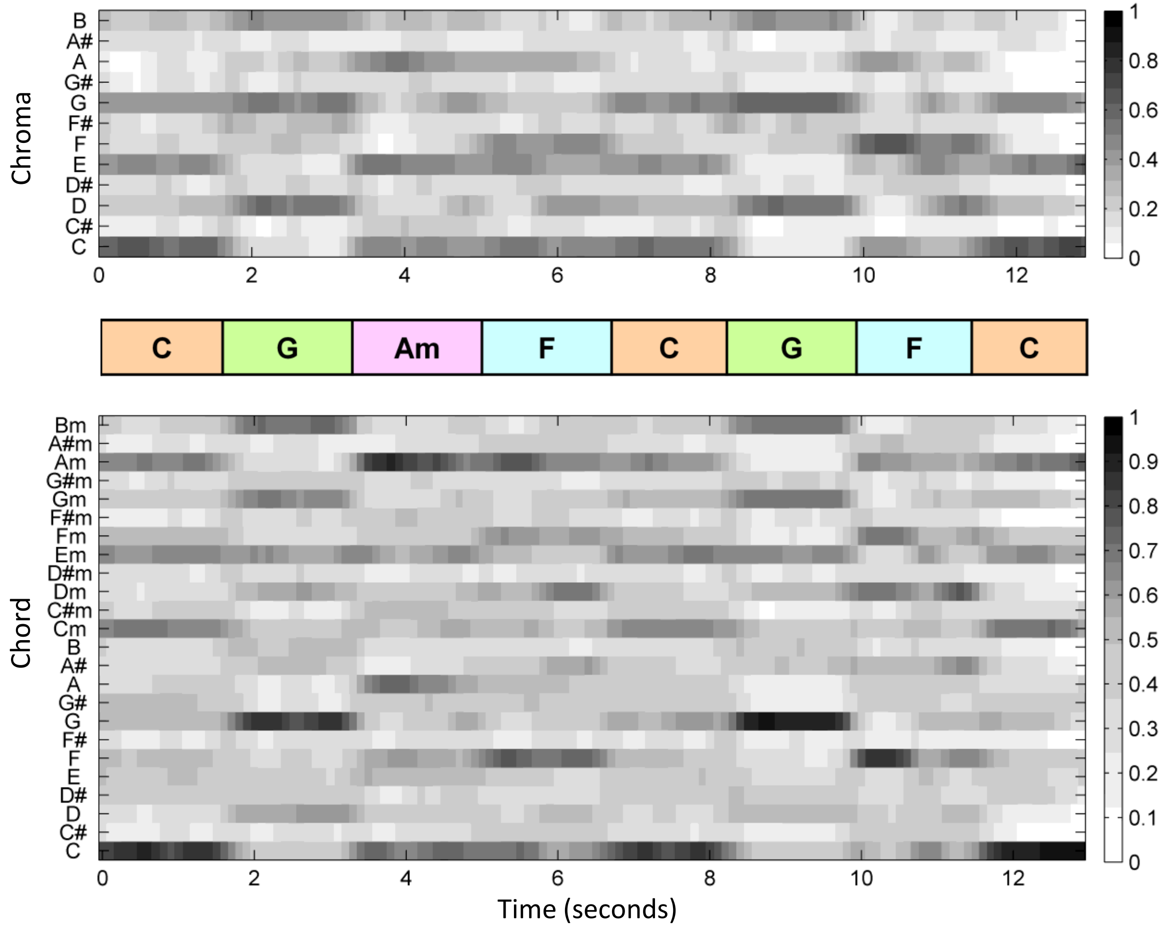


- Cosine similarity:
  - $s(\mathbf{x}, \mathbf{y}) = \frac{\langle \mathbf{x} | \mathbf{y} \rangle}{||\mathbf{x}||\,||\mathbf{y}||}$
- Find the chord template maximizing the similarity:
  - $\lambda_n := \mathrm{argmax}_{\lambda \in \Lambda} s(\mathbf{t}_\lambda, \mathbf{x}_n)$
- Time-chord representation



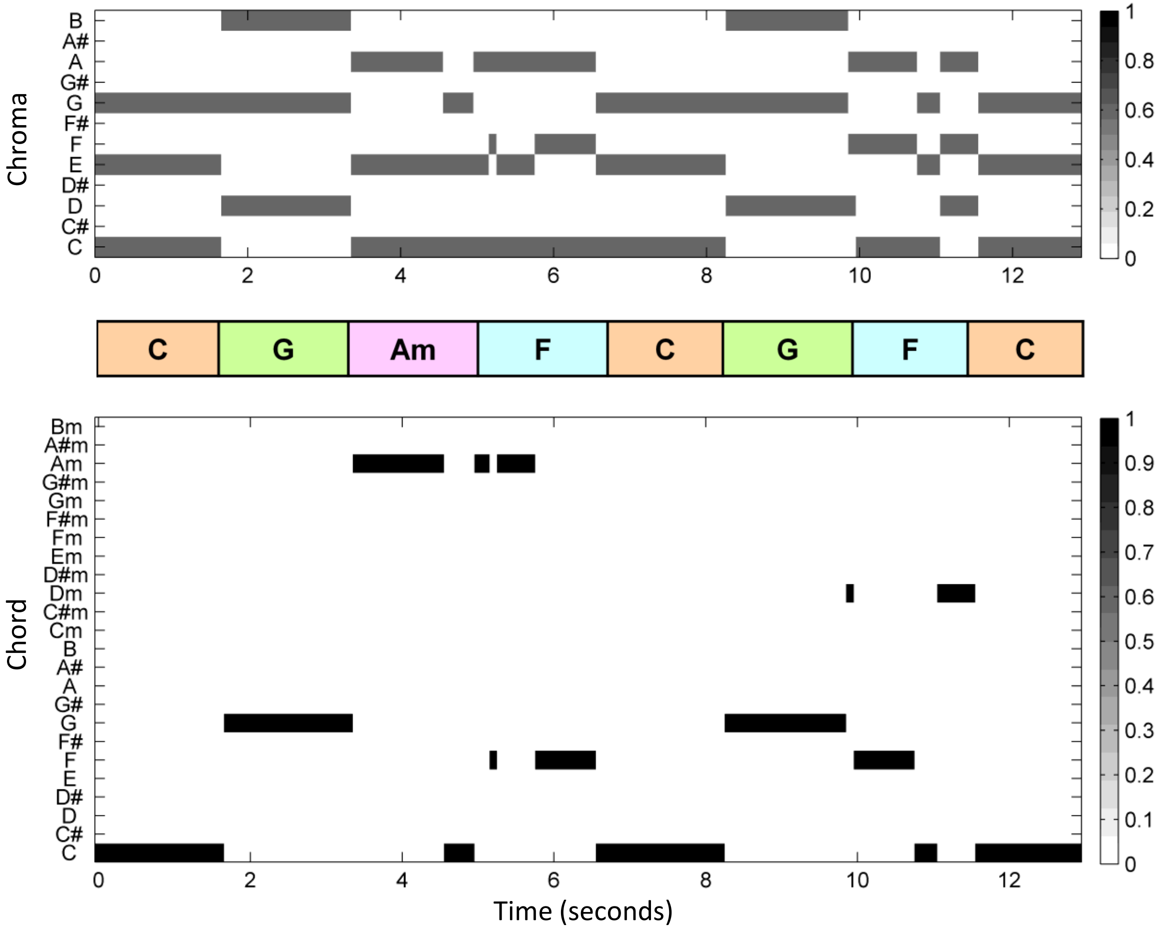
From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

A closer look (1)



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

A closer look (2)



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

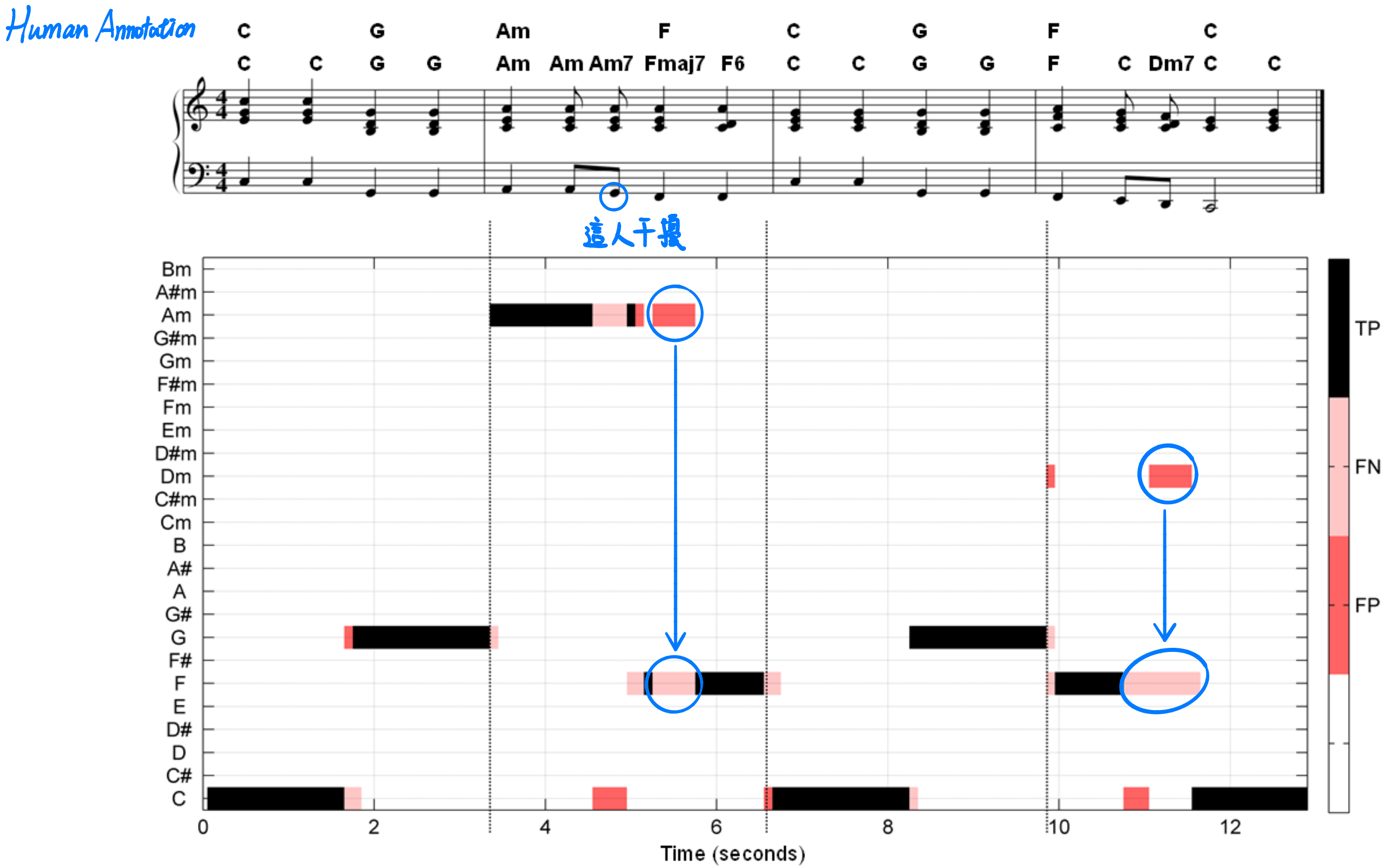
Evaluation

expert's

- Annotation
  - Chord annotation (by experts) is not 100% objective
  - Chord annotation could be complicated (e.g., seventh chords, ninth chords, …)
  - Potential issues: passing notes, suspended notes, …
- Figure-of-merit:
  - True positive (TP), false positive (FP), false negative (FN)
  - Precision (P), recall (R), and F-measure (F), and accuracy

$$\mathrm{P} = \frac{\#\mathrm{TP}}{\#\mathrm{TP} + \#\mathrm{FP}}, \qquad \mathrm{R} = \frac{\#\mathrm{TP}}{\#\mathrm{TP} + \#\mathrm{FN}}, \qquad \mathrm{F} = \frac{2\mathrm{PR}}{\mathrm{P} + \mathrm{R}}$$

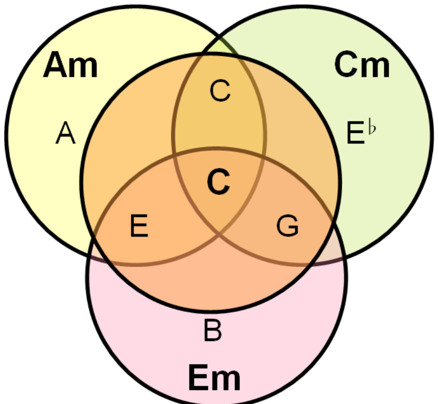
Evaluation



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Ambiguities in chord recognition (1): chord ambiguities

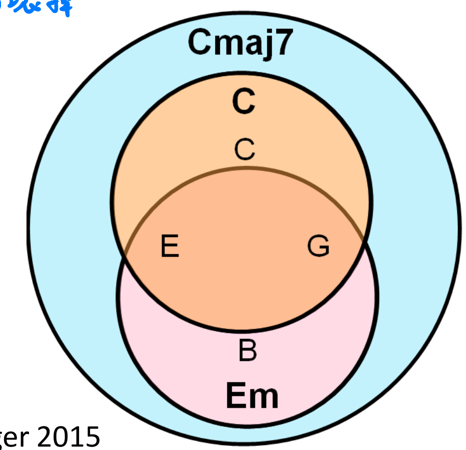
- Different chords could be closely related by sharing some of their notes
  - Am: A+C+E
  - Cm: C+Eb+G
  - Em: E+G+B



- If there is a “Major 7” chord …
  - CMaj7: C+E+G+B

7和弦在template matching上很容易壞掉

  - $s(\mathbf{t}_{\mathbf{CMaj7}}, \mathbf{t}_{\mathbf{C}}) = s(\mathbf{t}_{\mathbf{CMaj7}}, \mathbf{t}_{\mathbf{Em}})$



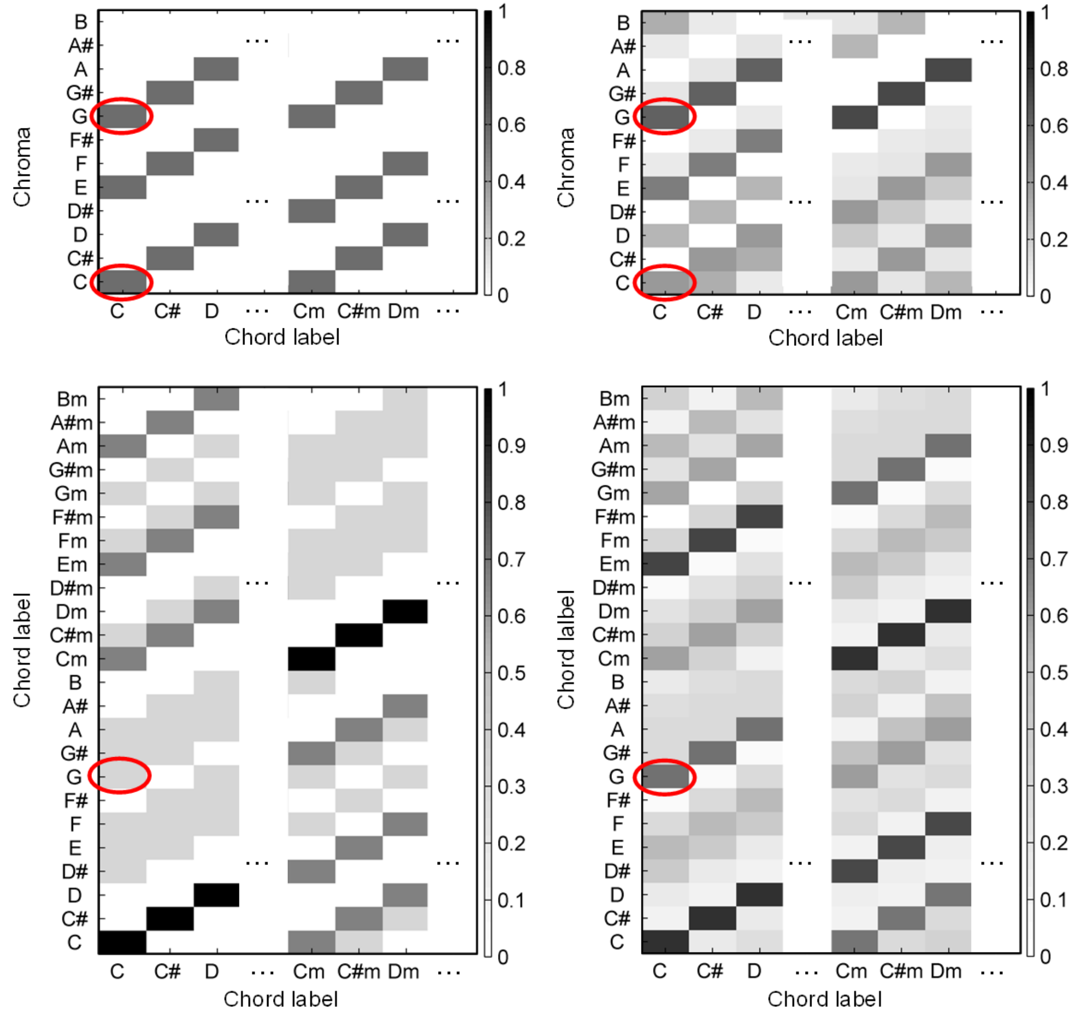
From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Ambiguities in chord recognition (2): acoustic ambiguities

- Example: the harmonics of C2

C2的 自然泛音們

  - C2, C3, G3, C4, E4, G4, Bb4, C5, …
  - C2 note already contains G, E, Bb
- Major-minor confusion
  - **Cm** chord (C+ Eb + G) contains **C** (C + E + G)



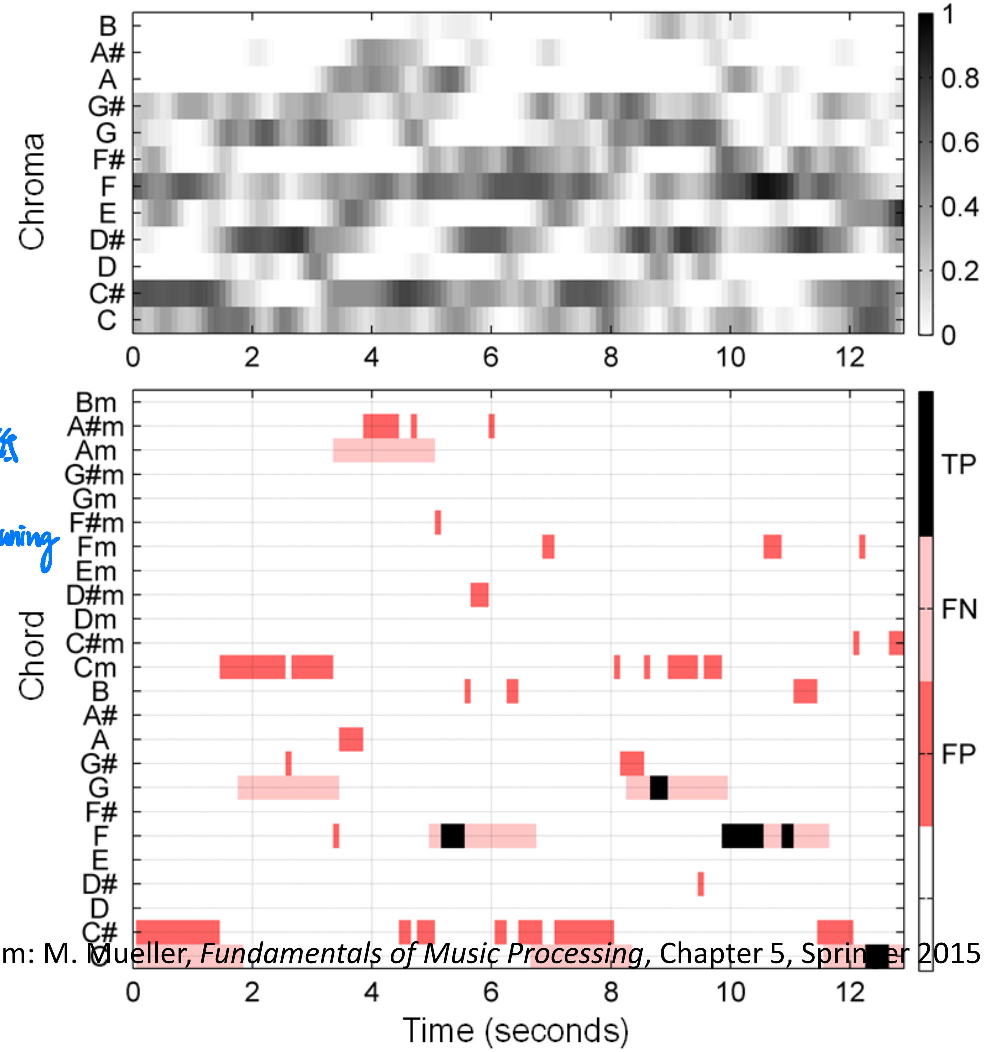
From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Ambiguities in chord recognition (3): Tuning

調音 A4=440,442Hz ...

- Recorded audio may not be in standard tuning
- Example: pitch shifted upward by 50 cents (half a semitone)

準度很低 (<10%) 應懷疑tuning

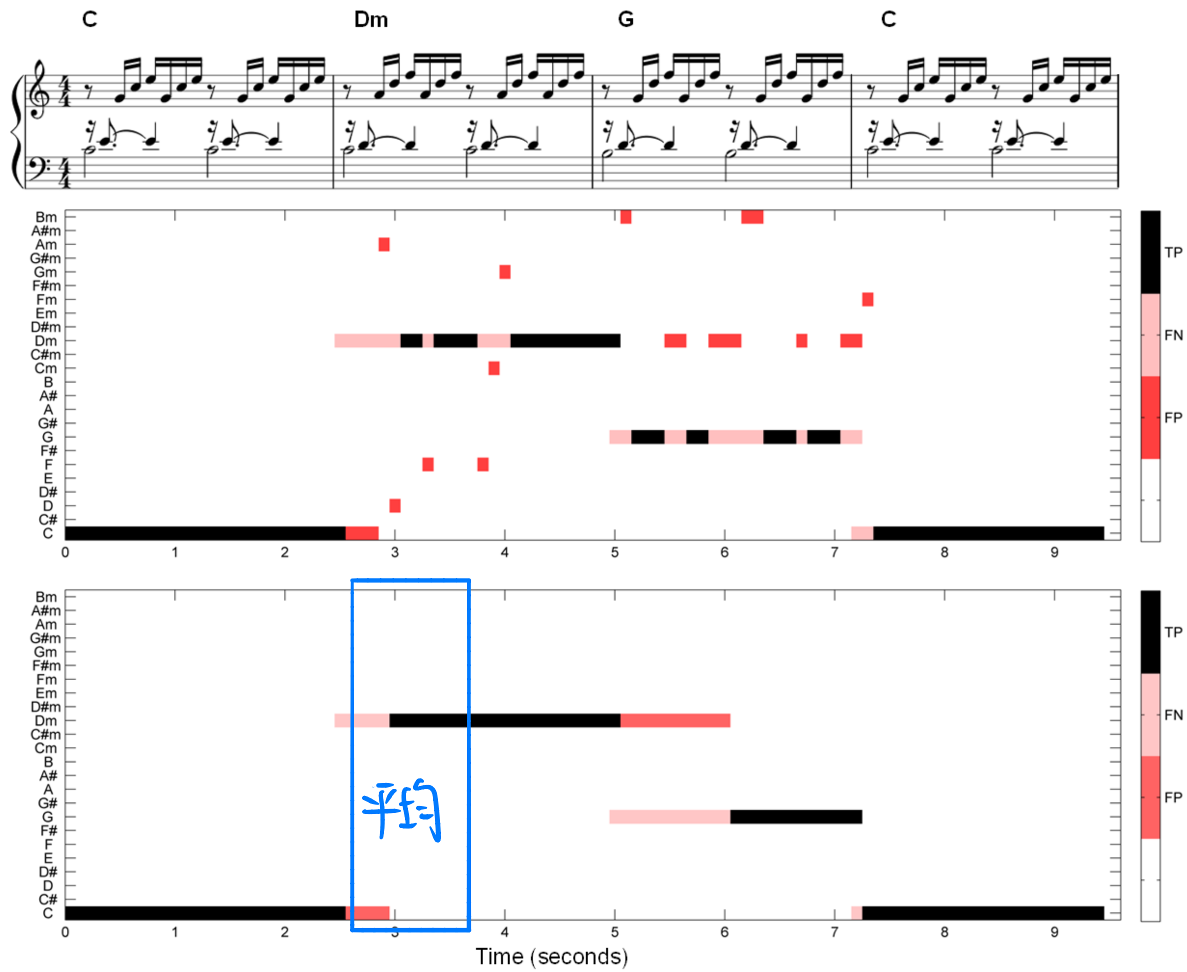


From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Ambiguities in chord recognition (4): Segmentation ambiguities

- Broken chord 分解和弦

一次看一區的chroma



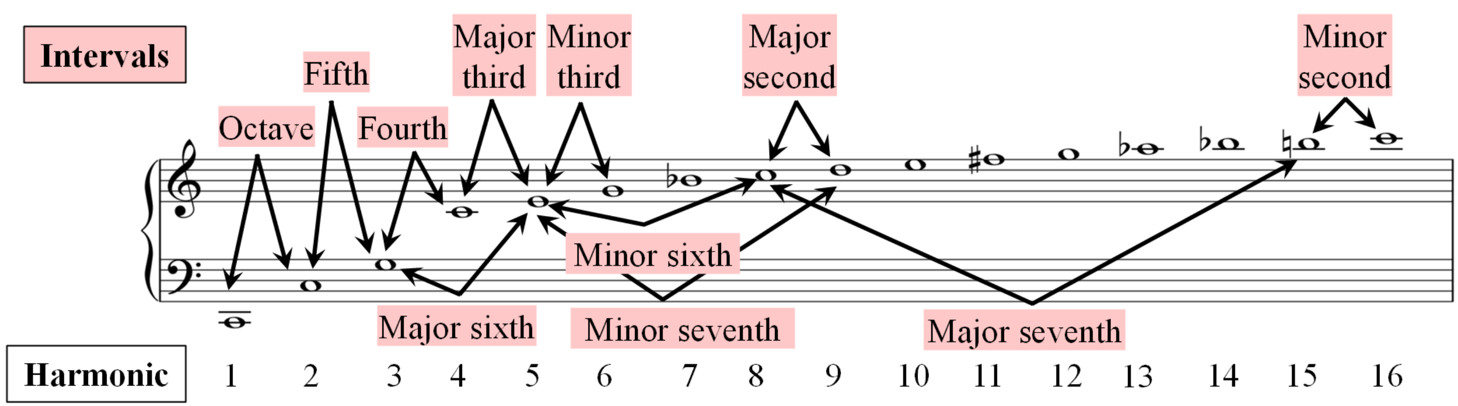
From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Enhancement strategies (1): templates with harmonics

- The original chord template:
  - $\boldsymbol{t}_{\mathrm{C}} = (1,0,0,0,1,0,0,1,0,0,0,0)$
- Assume the energy of the k-th partial is α^{k-1} for some $\alpha \in [0, 1]$

第k的倍频的强度

  - $\boldsymbol{u}_{\mathrm{C}}^{h} = (1 + \alpha + \alpha^3 + \alpha^7, 0,0,0, \alpha^4, 0,0, \alpha^2 + \alpha^5, 0,0, \alpha^6, 0)$
  - $\boldsymbol{t}_{\mathrm{C}} = \boldsymbol{u}_{\mathrm{C}}^{h} + \boldsymbol{u}_{\mathrm{E}}^{h} + \boldsymbol{u}_{\mathrm{G}}^{h}$



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Enhancement strategies (2): templates from examples or learning

真實世界的音

- Supervised learning: use labeled training data to determine chroma templates
- Exemplar-based: just samples chroma features from train data as templates (can be many templates for one chord label) Ex: 錄88鍵音，再合成 chord template
- Take average among the chord templates
- Clustering, dictionary learning, …

從很多 C 的 template 選出比較有代表性的

or

其它和弦

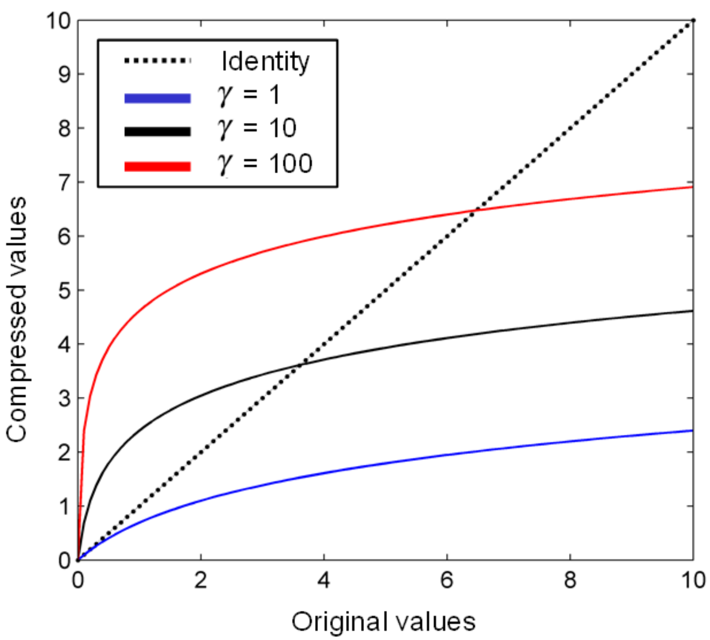
Enhancement strategies (3): nonlinear mapping

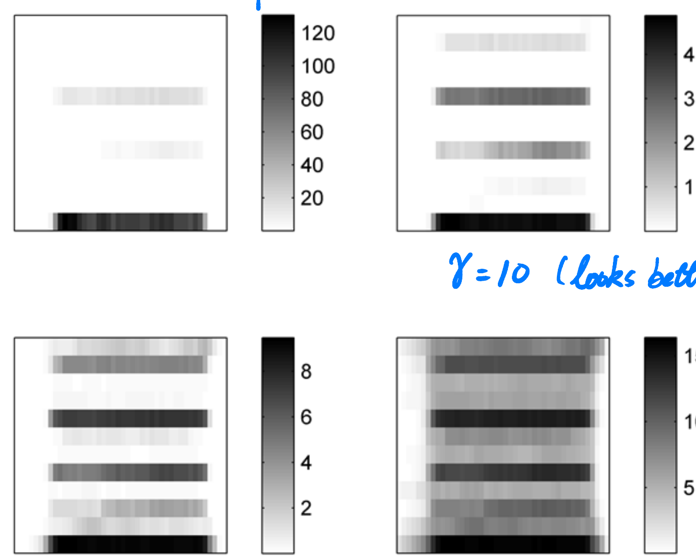
音量問題

大小聲適合的 template 可能不同

- Our perception of sound intensity is log-scale
  - But taking logarithm is unfeasible in some cases (taking logarithm maps 0 to $-\infty$)
- Solution: taking $\Gamma_\gamma(x) := \log(1 + \gamma|x|)$, or power scale $|x|^\gamma$

power transformation





$\gamma=10$ (looks better)

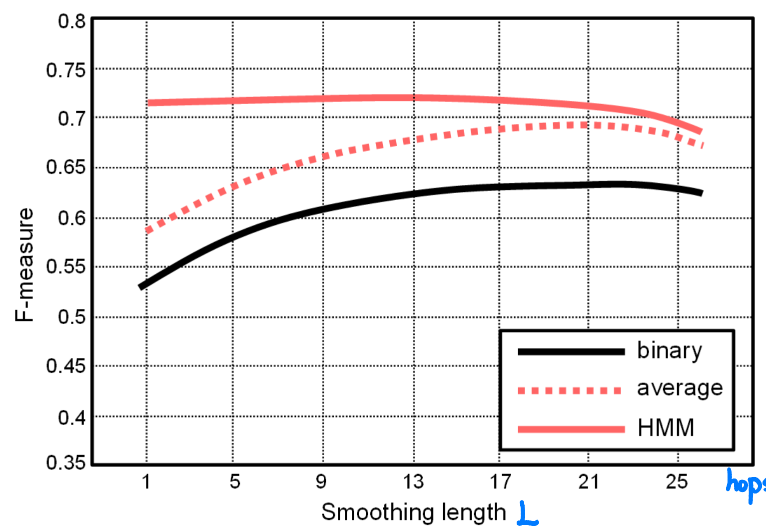
$\gamma=100$ $\gamma=1000$

From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Enhancement strategies (4): pre-filtering

before template matching

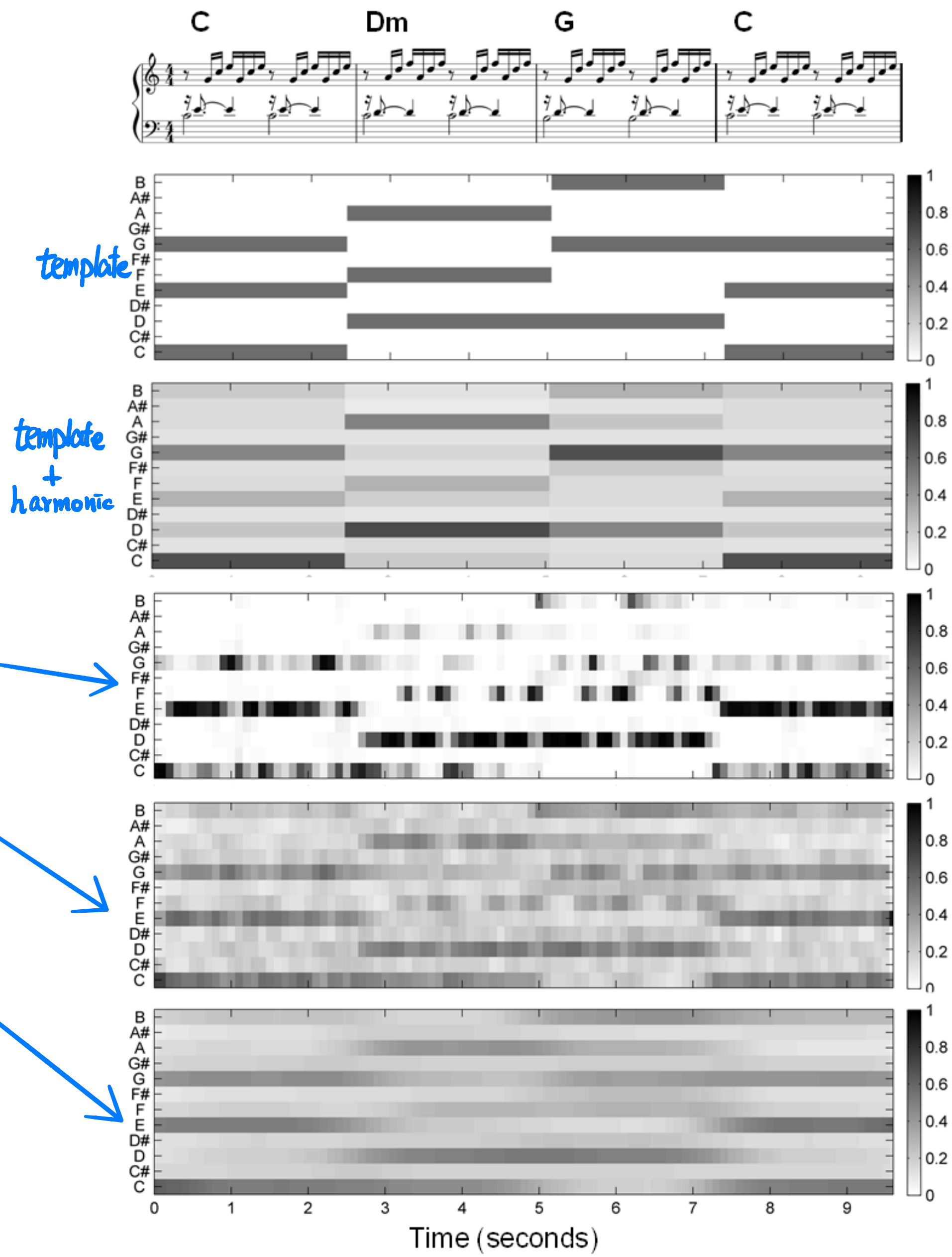
- Logarithmic compression can be viewed as a type of *spectral smoothing*
- *Temporal smoothing*: taking average or low-pass filtering on each of the twelve component of the sequence along time axis
  - Mean filter 平滑
  - Median filter 去極端值 ex: 鼓聲
- Smoothing length L 看幾拍換一次合弦
  - $x_n^L(i) := \frac{1}{L}\sum_{l=0}^{L-1} x_{n+l-\left\lfloor\frac{L-1}{2}\right\rfloor}(i)$
- Experiment evidence:



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Comparison

1) Musical score and annotation
2) Binary chord template
3) Average chord templates obtained from training data
4) Original chroma features obtained from audio recording (feature rate = 10 Hz)
5) Chroma feature from (4) after $\log(1+\gamma|x|), \gamma = 10$
6) Chroma feature from (5) after smoothing ($L = 20$)



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

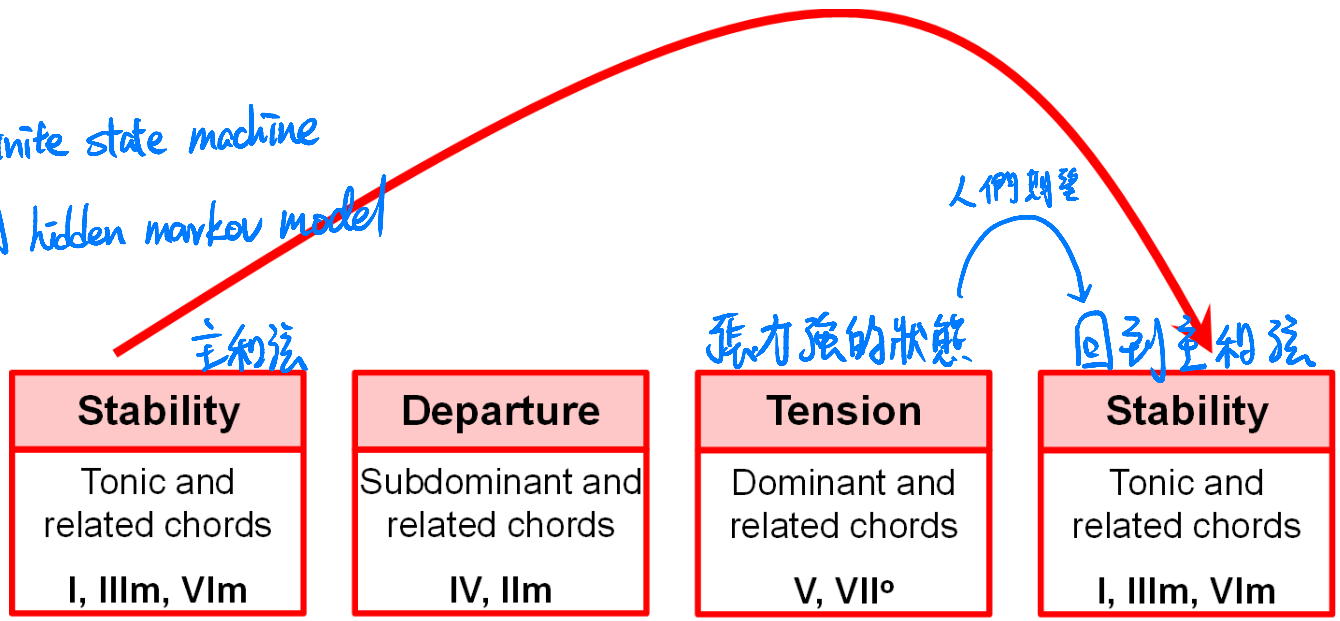
Chord progressions (context)

和弦的進行有規則可循

- Chord progressions are not arbitrary
  - Example 1: I-IV-I-V-I (C-F-C-G-C)
  - Example 2: I-V-VI-III-IV-I-II-V (C-G-Am-Em-F-C-Dm-G)

似 finite state machine

可利用 hidden markov model



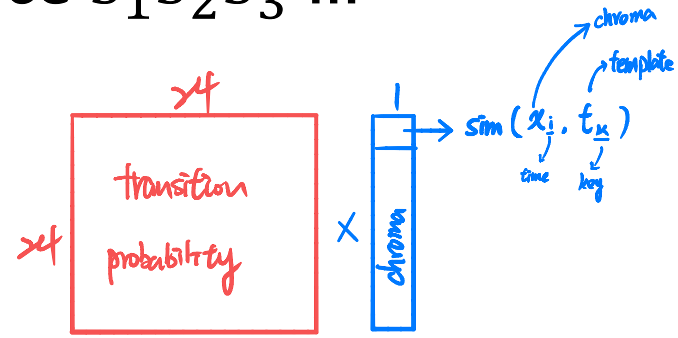
From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

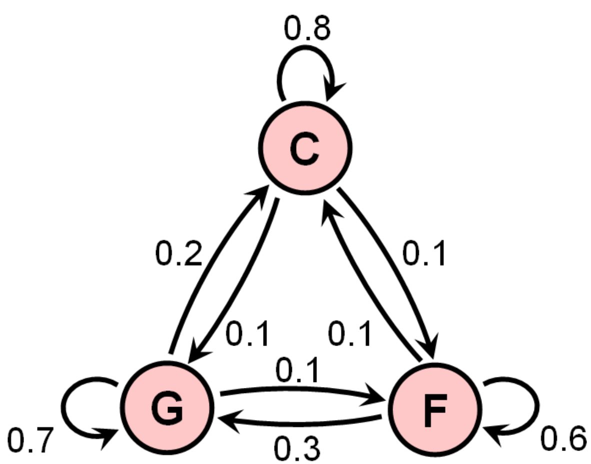
Finite states：24組大小和弦

non-Markov v.s. Markov
符合假設,常見為線性序列

Markov chains of chord progressions

- Markov states $\alpha_1, \alpha_2, \alpha_3$ in a sequence $s_1 s_2 s_3$ ...
- Markov property:
  - $P(s_{n+1} = \alpha_j | s_n = \alpha_i, s_{n-1} = \alpha_k, \ldots)$
    $= \underline{P(s_{n+1} = \alpha_j | s_n = \alpha_i)}$
    假設: t=n+1 的狀態 只和 t=n 的狀態有關





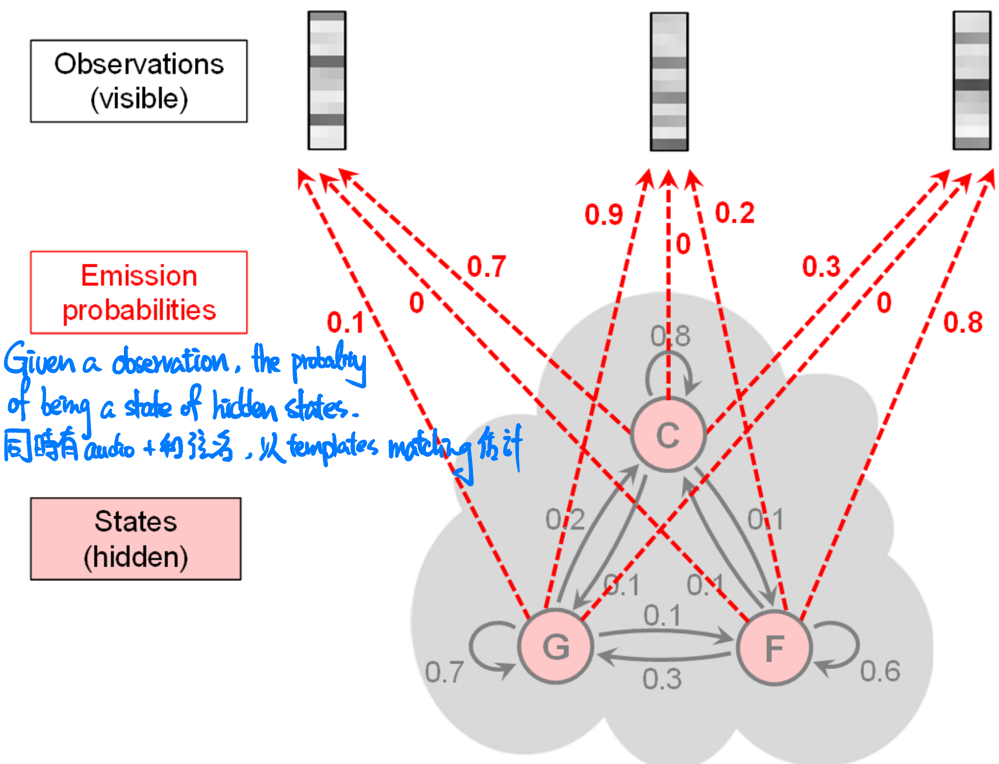
transition probability (可從網上找吉他譜來建)

	α_1 = C	α_2 = G	α_3 = F
α_1 = C	$a_{11} = 0.8$	$a_{12} = 0.1$	$a_{13} = 0.1$
α_2 = G	$a_{21} = 0.2$	$a_{22} = 0.7$	$a_{23} = 0.1$
α_3 = F	$a_{31} = 0.1$	$a_{32} = 0.3$	$a_{33} = 0.6$

From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

HMM model

- Observations
  - Chroma features observation
  - Or template-based result
- Hidden states target
  - “Refined” chord sequence
  - The answer we want
- Transition probability
  - From training data
- Emission probability
  - From training data

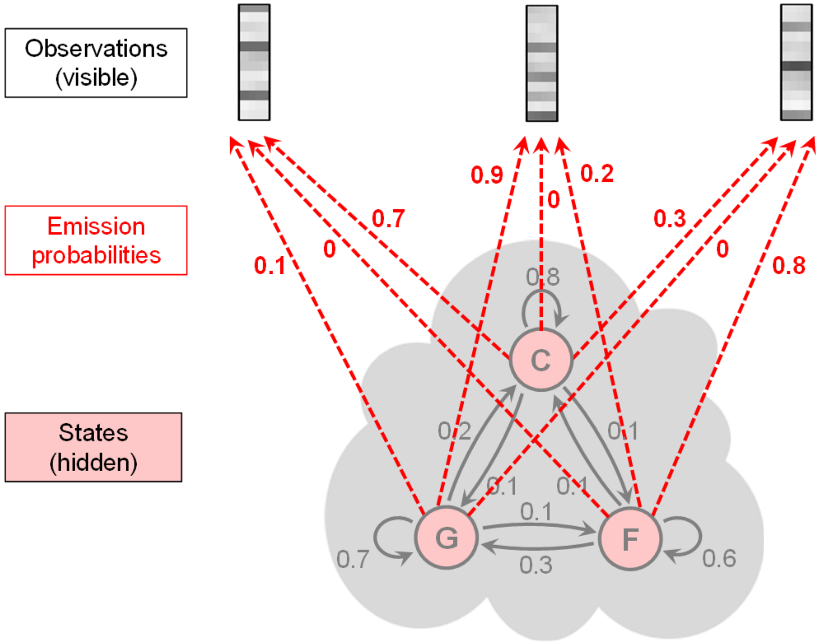


Given a observation, the probalbity of being a state of hidden states.
同时有audio + 和弦名，从templates matching 估计

From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Discrete HMM components

- Map an arbitrary chroma features in the test data to one of a finite set of prototype vectors (codebook)
  - Quantization: map the feature
  - Clustering: train the codebook



Component	Meaning	Reference
$\mathcal{A}$	Set of states α_i for $i \in [1:I]$	(5.18)
A	State transition probabilities a_{ij} for $i,j \in [1:I]$	(5.20)
C	Initial state probabilities c_i for $i \in [1:I]$	(5.22)
$\mathcal{B}$	Set of observation symbols β_k for $k \in [1:K]$	(5.25)
B	Emission probabilities b_{ik} for $i \in [1:I]$ and $k \in [1:K]$	(5.26)

From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

chorma (observation) → chord (state)

A naïve HMM model training method

- I states (e.g., $I = 24$), K observation symbols
- For the training data:

$$c_i = number\ of\ transition\ form\ \alpha_i\ at\ time\ (n = 1)$$

$$a_{ij} = \frac{number\ of\ transitions\ from\ \alpha_i\ to\ \alpha_j}{number\ of\ transitions\ from\ \alpha_i}$$

$$b_{ik} = \frac{number\ of\ transitions\ from\ \alpha_i and\ observing\ \beta_k}{number\ of\ transitions\ from\ \alpha_i}$$

Component	Meaning	Reference
$\mathcal{A}$	Set of states α_i for $i \in [1:I]$	(5.18)
A	State transition probabilities a_{ij} for $i, j \in [1:I]$	(5.20)
C	Initial state probabilities c_i for $i \in [1:I]$	(5.22)
$\mathcal{B}$	Set of observation symbols β_k for $k \in [1:K]$	(5.25)
B	Emission probabilities b_{ik} for $i \in [1:I]$ and $k \in [1:K]$	(5.26)

From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

The uncovering problem of HMM

available in Scikit

- Given:
  - An HMM specified by $\Theta = (\boldsymbol{A}, A, C, \boldsymbol{B}, B)$
  - An observation sequence $O = (o_1, o_2, \dots, o_N)$
- Find:
  - The single state sequence $S = (s_1, s_2, \dots, s_N)$, $s_i \in \boldsymbol{A}$ that "best explain" the observation sequence

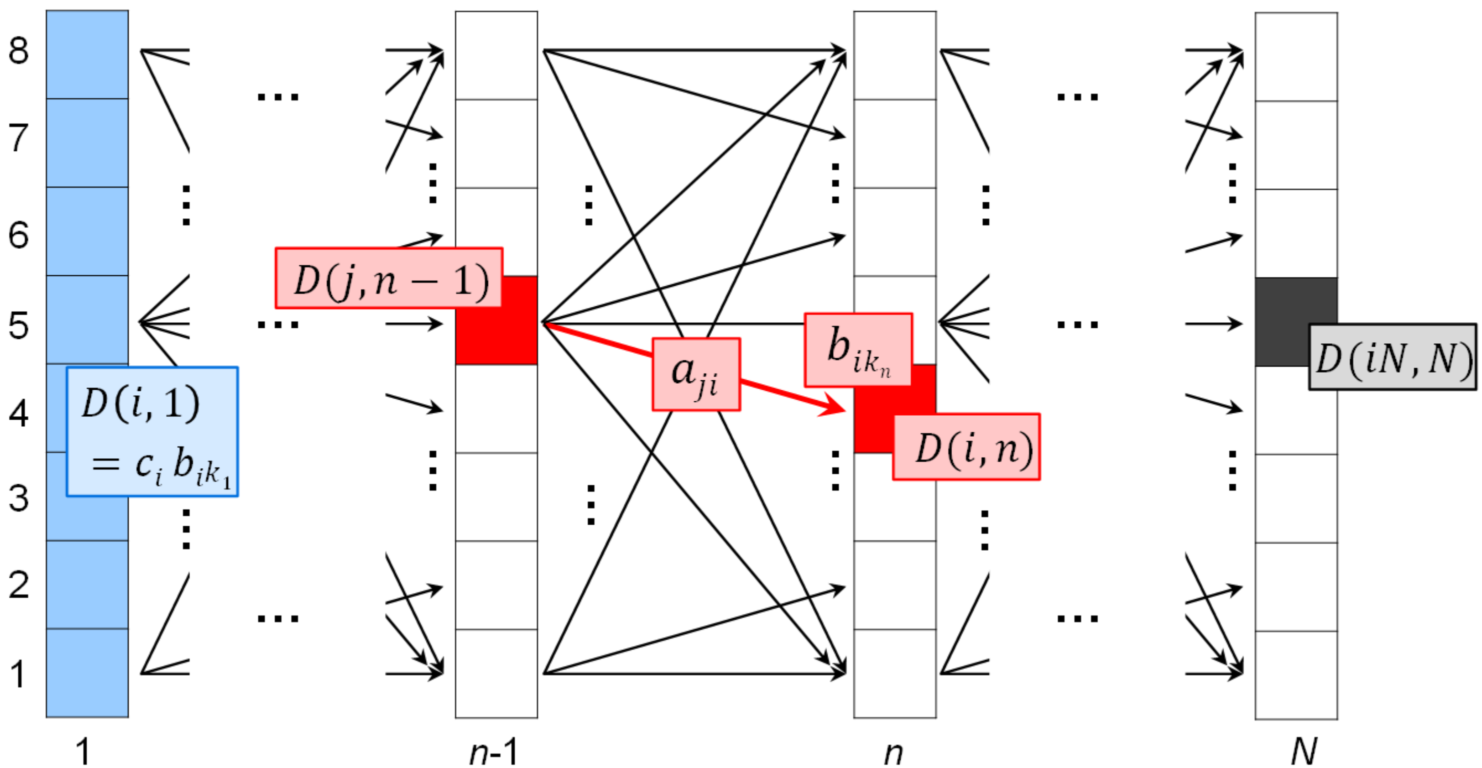
$$S^* = \underset{S}{\operatorname{argmax}} P(O, S|\Theta)$$

  - I states, N time frames -> total I^N possible paths
  - How to solve this problem?

Viterbi's algorithm (1)

- Based on dynamic programming: the optimal result for a problem is built on the optimal result for the sub-problems

Let I = 8,



From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Viterbi's algorithm (2)

Algorithm: VITERBI

Input: HMM specified by $\Theta = (\mathcal{A}, A, C, \mathcal{B}, B)$
Observation sequence $O = (o_1 = \beta_{k_1}, o_2 = \beta_{k_2}, \ldots, o_N = \beta_{k_N})$
Output: Optimal state sequence $S^* = (s_1^*, s_2^*, \ldots, s_N^*)$

Procedure: Initialize the $(I \times N)$ matrix $\mathbf{D}$ by $\mathbf{D}(i,1) = c_i b_{ik_1}$ for $i \in [1:I]$. Then compute in a nested loop for $n = 2, \ldots, N$ and $i = 1, \ldots, I$:

$$\begin{aligned} \mathbf{D}(i,n) &= \max_{j\in[1:I]} \left(a_{ji} \cdot \mathbf{D}(j,n-1)\right) \cdot b_{ik_n} \\ \mathbf{E}(i,n-1) &= \operatorname{argmax}_{j\in[1:I]} \left(a_{ji} \cdot \mathbf{D}(j,n-1)\right) \end{aligned}$$

For backtracking

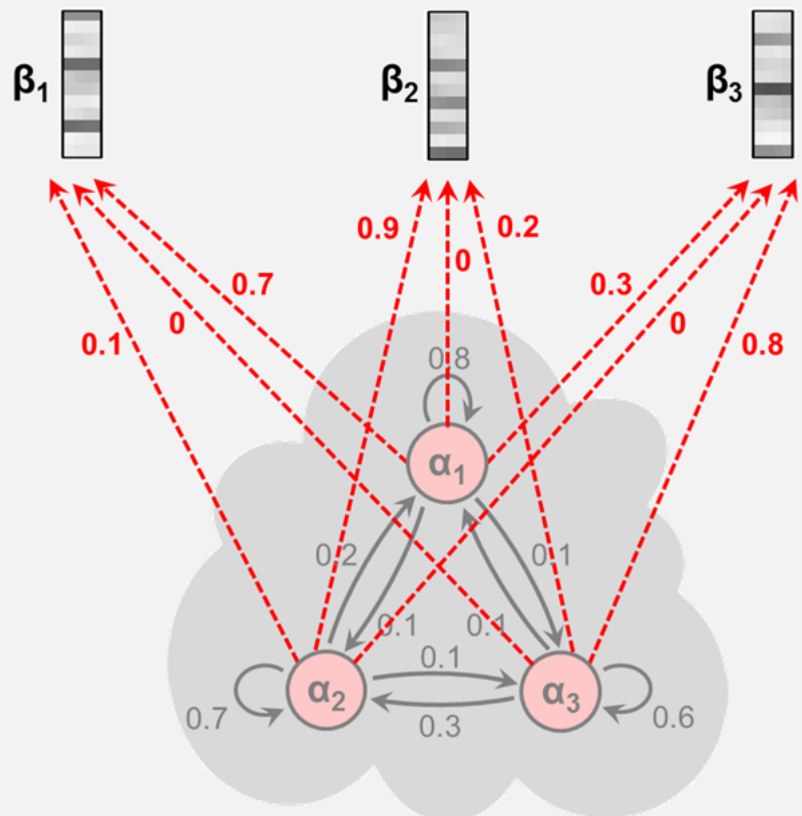
Set $i_N = \operatorname{argmax}_{j\in[1:I]} \mathbf{D}(j,N)$ and compute for decreasing $n = N-1, \ldots, 1$ the maximizing indices

$$i_n = \operatorname{argmax}_{j\in[1:I]} \left(a_{ji_{n+1}} \cdot \mathbf{D}(j,n)\right) = \mathbf{E}(i_{n+1}, n).$$

The optimal state sequence $S^* = (s_1^*, \ldots, s_N^*)$ is defined by $s_n^* = \alpha_{i_n}$ for $n \in [1:N]$.

From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

An example of Viterbi's algorithm (1)



State transition probabilities

A	α_1	α_2	α_3
α_1	0.8	0.1	0.1
α_2	0.2	0.7	0.1
α_3	0.1	0.3	0.6

Initial state probabilities

C	α_1	α_2	α_3
	0.6	0.2	0.2

Emission probabilities

B	β_1	β_2	β_3
α_1	0.7	0	0.3
α_2	0.1	0.9	0
α_3	0	0.2	0.8

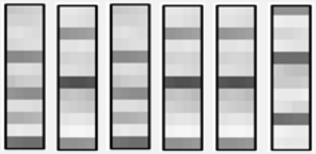
From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

An example of Viterbi's algorithm (2)

Input

Observation sequence

$O = (o_1, o_2, o_3, o_4, o_5, o_6)$



$\beta_1 \quad \beta_3 \quad \beta_1 \quad \beta_3 \quad \beta_3 \quad \beta_2$

Viterbi algorithm

D	$o_1 = \beta_1$	$o_2 = \beta_3$	$o_3 = \beta_1$	$o_4 = \beta_3$	$o_5 = \beta_3$	$o_6 = \beta_2$
α_1	0.4200	0.1008	0.0564	0.0135	0.0033	0
α_2	0.0200	0	0.0010	0	0	0.0006
α_3	0	0.0336	0	0.0045	0.0022	0.0003

E	$o_1 = \beta_1$	$o_2 = \beta_3$	$o_3 = \beta_1$	$o_4 = \beta_3$	$o_5 = \beta_3$
α_1	$\boxed{1}$	$\boxed{1}$	1	1	1
α_2	1	1	1	1	$\boxed{3}$
α_3	1	3	$\boxed{1}$	$\boxed{3}$	3

$\boxed{i_6 = 2}$

Output

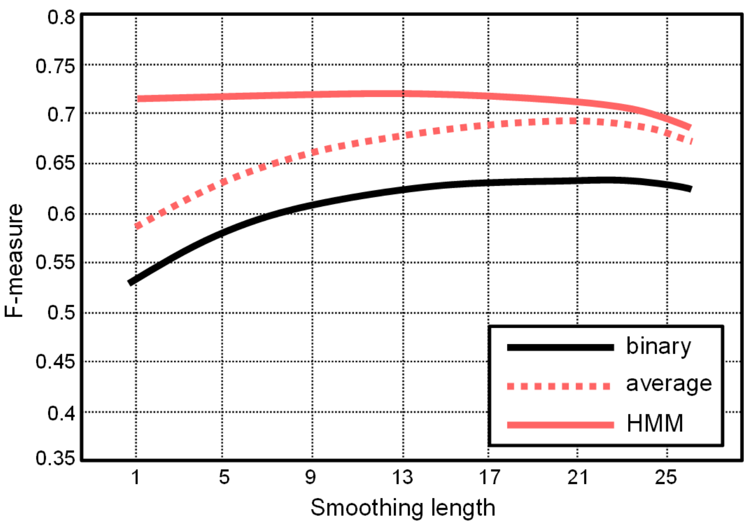
Optimal state sequence

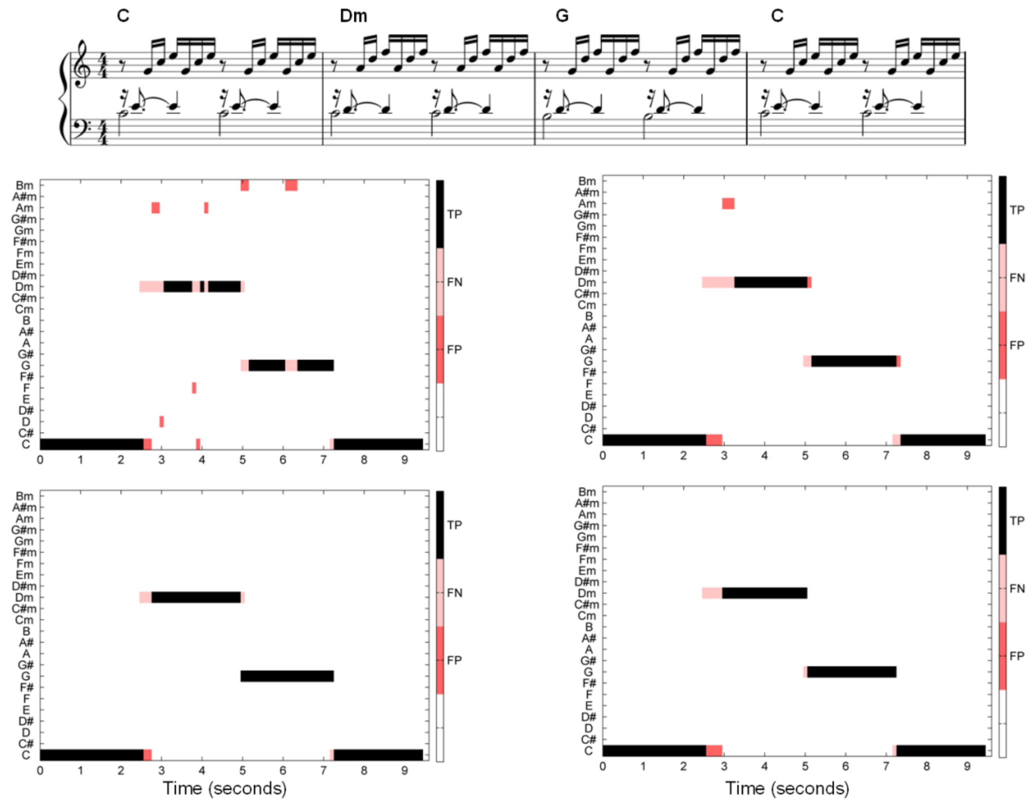
$S^* = (\alpha_1, \alpha_1, \alpha_1, \alpha_3, \alpha_3, \alpha_2)$

From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Result

- Better than temporal smoothing





From: M. Mueller, *Fundamentals of Music Processing*, Chapter 5, Springer 2015

Tools and Datasets

- Chroma toolbox V. 2.0 (in MATLAB)
- Supervised Chord Recognition for Music Audio in Matlab
- MIREX Chord Recognition 有很多MIR的情報

Online Chord Recognition

已成熟很難再improve的research topic

Remaining challenge: key + chord 功能和聲

三和弦已做的差不多，七和弦，特殊和弦難做

- Chordify
- RiffStation
- Chord tell → Youtube to chord

Online Chord Recognition

- Chordify
- RiffStation
- Chord tell