Exploitatie van voorkennis bij het automatisch afleiden van toonaarden en akkoorden uit muzikale audio

Exploiting Prior Knowledge during Automatic Key and Chord Estimation from Musical Audio

Johan Pauwels

Promotoren: prof. dr. ir. J.-P. Martens, prof. dr. M. Leman
Proefschrift ingediend tot het behalen van de graad van
Doctor in de Ingenieurswetenschappen

Vakgroep Elektronica en Informatiesystemen
Voorzitter: prof. dr. ir. R. Van de Walle
Faculteit Ingenieurswetenschappen en Architectuur
Academiejaar 2015 - 2016

UNIVERSITEIT GENT

ISBN 978-90-8578-883-6
NUR 962, 965
Wettelijk depot: D/2016/10.500/15

Abstract

Chords and keys are two ways of describing music. They are exemplary of a general class of symbolic notations that musicians use to exchange information about a music piece. This information can range from simple tempo indications such as “allegro” to precise instructions for a performer of the music. Concretely, both keys and chords are timed labels that describe the harmony during certain time intervals, where harmony refers to the way music notes sound together. Chords describe the local harmony, whereas keys offer a more global overview and consequently cover a sequence of multiple chords.

Common to all music notations is that certain characteristics of the music are described while others are ignored. The adopted level of detail depends on the purpose of the intended information exchange. A simple description such as “menuet”, for example, only serves to roughly describe the character of a music piece. Sheet music on the other hand contains precise information about the pitch, discretised information pertaining to timing and limited information about the timbre. Its goal is to permit a performer to recreate the music piece. Even so, the information about timing and timbre still leaves some space for interpretation by the performer.

The opposite of a symbolic notation is a music recording. It stores the music in a way that allows for a perfect reproduction. The disadvantage of a music recording is that it does not allow to manipulate a single aspect of a music piece in isolation, or at least not without degrading the quality of the reproduction. For instance, it is not possible to change the instrumentation in a music recording, even though this would only require the simple change of a few symbols in a symbolic notation.

Despite the fundamental differences between a music recording and a

symbolic notation, the two are of course intertwined. Trained musicians can listen to a music recording (or live music) and write down a symbolic notation of the played piece. This skill allows one, in theory, to create a symbolic notation for each recording in a music collection. In practice however, this would be too labour intensive for the large collections that are available these days through online stores or streaming services. Automating the notation process is therefore a necessity, and this is exactly the subject of this thesis. More specifically, this thesis deals with the extraction of keys and chords from a music recording. A database with keys and chords opens up applications that are not possible with a database of music recordings alone. On one hand, chords can be used on their own as a compact representation of a music piece, for example to learn how to play an accompaniment for singing. On the other hand, keys and chords can also be used indirectly to accomplish another goal, such as finding similar pieces.

Because music theory has been studied for centuries, a great body of knowledge about keys and chords is available. It is known that consecutive keys and chords form sequences that are all but random. People happen to have certain expectations that must be fulfilled in order to experience music as pleasant. Keys and chords are also strongly intertwined, as a given key implies that certain chords will likely occur and a set of given chords implies an encompassing key in return. Consequently, a substantial part of this thesis is concerned with the question whether musicological knowledge can be embedded in a technical framework in such a way that it helps to improve the automatic recognition of keys and chords.

The technical framework adopted in this thesis is built around a hidden Markov model (HMM). This facilitates an easy separation of the different aspects involved in the automatic recognition of keys and chords. Most experiments reviewed in the thesis focus on taking into account musicological knowledge about the musical context and about the expected chord duration. Technically speaking, this involves a manipulation of the transition probabilities in the HMMs. To account for the interaction between keys and chords, every HMM state is actually representing the combination of a key and a chord label.

In the first part of the thesis, a number of alternatives for modelling the context are proposed. In particular, separate key change and chord change models are defined such that they closely mirror the way musicians conceive harmony. Multiple variants are considered that differ in the size of the context that is accounted for and in the knowledge source from which they were compiled. Some models are derived from a music corpus with key and chord notations whereas others follow directly from music theory.

In the second part of the thesis, the contextual models are embedded in a system for automatic key and chord estimation. The features used in that system are so-called chroma profiles, which represent the saliences of the pitch classes in the audio signal. These chroma profiles are acous-

tically modelled by means of templates (idealised profiles) and a distance measure. In addition to these acoustic models and the contextual models developed in the first part, durational models are also required. The latter ensure that the chord and key estimations attain specified mean durations.

The resulting system is then used to conduct experiments that provide more insight into how each system component contributes to the ultimate key and chord output quality. During the experimental study, the system complexity gets gradually increased, starting from a system containing only an acoustic model of the features that gets subsequently extended, first with duration models and afterwards with contextual models. The experiments show that taking into account the mean key and mean chord duration is essential to arrive at acceptable results for both key and chord estimation. The effect of using contextual information, however, is highly variable. On one hand, the chord change model has only a limited positive impact on the chord estimation accuracy (two to three percentage points), but this impact is fairly stable across different model variants. On the other hand, the chord change model has a much larger potential to improve the key output quality (up to seventeen percentage points), but only on the condition that the variant of the model is well adapted to the tested music material. Lastly, the key change model has only a negligible influence on the system performance.

In the final part of this thesis, a couple of extensions to the formerly presented system are proposed and assessed. First, the global mean chord duration is replaced by key-chord specific values, which has a positive effect on the key estimation performance. Next, the HMM system is modified such that the prior chord duration distribution is no longer a geometric distribution but one that better approximates the observed durations in an appropriate data set. This modification leads to a small improvement of the chord estimation performance, but of course, it requires the availability of a suitable data set with chord notations from which to retrieve a target durational distribution. A final experiment demonstrates that increasing the scope of the contextual model only leads to statistically insignificant improvements. On top of that, the required computational load increases greatly.

Samenvatting

Akkoorden en toonaarden zijn twee manieren om muziek te beschrijven. Het zijn voorbeelden van de ruimere klasse van symbolische (muziek-) notaties die gebruikt worden door muzikanten om informatie over een muziekstuk uit te wisselen. Deze informatie kan gaan van eenvoudige tempoaanduidingen zoals "allegro" tot precieze instructies voor de uitvoerder van een stuk. Concreet zijn toonaarden en akkoorden beiden tijdgebonden labels die de harmonie beschrijven gedurende bepaalde tijdsintervallen, waarbij harmonie slaat op het samenklinken van muzieknoten. Akkoorden beschrijven de lokale harmonie, terwijl toonaarden een meer globaal overzicht geven en dus per definitie een opeenvolging van akkoorden bestrijken.

Alle symbolische notaties hebben met elkaar gemeen dat ze bepaalde karakteristieken van een muziekstuk beschrijven en andere aspecten negeren. Hoe gedetailleerd dit gebeurt, hangt af van de reden waarom de informatie uitgewisseld wordt. Een eenvoudige beschrijving zoals "menuet", bijvoorbeeld, dient slechts om het karakter van een muziekstuk grof te benoemen. Een muziekpartituur daarentegen, bevat precieze informatie over de toonhoogtes, gediscretiseerde informatie met betrekking tot de timing en beperkte informatie over de klankkleur. Haar doel is dan ook om het recreëren van een muziekstuk mogelijk te maken. De informatie over timing en klankkleur moet niettemin nog aangevuld worden met de interpretatie van de uitvoerder.

Tegenover de symbolische notatie staat de muziekopname, die een getrouwe reproductie van het muziekstuk toelaat. Het nadeel van een muziekopname is echter dat deze niet toelaat om één enkel aspect van het muziekstuk apart te wijzigen, tenminste niet zonder de reproductiekwali-

teit te verminderen. Het is bijvoorbeeld niet mogelijk om de instrumentatie in een muziekopname te wijzigen, terwijl dit slechts een eenvoudige verandering van een paar symbolen vereist in een symbolische notatie.

Hoewel een opname dus fundamenteel verschilt van een symbolische notatie, zijn beiden uiteraard nauw aan elkaar gerelateerd. Getrainde muzikanten kunnen muziekopnames (of live-muziek) beluisteren en een symbolische notatie voor het gespeelde stuk uitschrijven. Deze vaardigheid staat dus in theorie toe om een symbolische notatie te maken voor elke opname in een muziekcollectie. In de praktijk zou dit echter veel te arbeidsintensief zijn voor de grote collecties die vandaag beschikbaar zijn via webwinkels of streamingproviders. Het is dus noodzakelijk om dit proces te automatiseren, en dat is precies het onderwerp van deze thesis. Het gaat hier specifiek om het extraheren van akkoorden en toonaarden uit muziekopnames. Een databank met akkoorden en toonaarden maakt toepassingen mogelijk die niet haalbaar zijn met een databank van opnames alleen. Enerzijds kunnen akkoorden op zichzelf gebruikt worden als beknopte weergave van een muzieknummer, bijvoorbeeld om een passende begeleiding voor zang te leren. Anderzijds kunnen toonaarden en akkoorden ook indirect gebruikt worden om andere toepassingen mogelijk te maken, zoals het vinden van gelijkaardige nummers.

Omdat muziektheorie al eeuwenlang bestudeerd wordt, bestaat er heel wat kennis over toonaarden en akkoorden. Zo is geweten dat opeenvolgende toonaarden en akkoorden sequenties vormen die allesbehalve willekeurig zijn. Mensen hebben nu eenmaal een bepaald verwachtingspatroon waaraan muziek moet voldoen om als aangenaam ervaren te worden. Toonaarden en akkoorden zijn ook nauw verwant, een gegeven toonaard impliceert dat bepaalde akkoorden verwacht kunnen worden en gegeven akkoorden impliceren op hun beurt een bepaalde overkoepelende toonaard. Daarom draait een substantieel deel van deze thesis om de vraag of musicologische kennis geïntegreerd kan worden in een technisch kader zodanig dat ze bijdraagt aan het verbeteren van het automatische afleiden van toonaarden en akkoorden.

Het technisch kader dat gebruikt wordt in deze thesis bestaat uit een hidden Markov model (HMM). Dit biedt de mogelijkheid om de verschillende aspecten die komen kijken bij automatische akkoord- en toonaardextractie eenvoudig van elkaar gescheiden te beschouwen. Het merendeel van de experimenten in deze thesis focust op het in rekening brengen van musicologische kennis over de muzikale context en over de verwachte akkoordduur. Technisch gesproken gaat het dan om het manipuleren van de transitieprobabiliteiten in de HMMs. Om de interactie tussen toonaarden en akkoorden samen te beschouwen, stelt elke toestand van een HMM de combinatie van een toonaard en een akkoord voor.

In het eerste deel van deze thesis worden een aantal alternatieven voor de contextmodellering voorgesteld. Meer bepaald worden aparte akkoordwissel- en toonaardwisselmodellen gedefiniëerd zodat het resultaat nauw

aansluit bij de manier waarop muzikanten harmonieleer opvatten. Meerdere varianten worden voorgesteld. Ze verschillen van elkaar in de grootte van de context die in rekening wordt gebracht en in de bron van de musicologische kennis die eraan ten grondslag ligt. Sommige modellen zijn afgeleid van een muziekcorpus met akkoord- en toonaardannotaties, terwijl anderen volgen uit de muziektheorie.

In het tweede deel van deze thesis worden de contextmodellen gebruikt als onderdeel van een systeem dat automatisch toonaarden en akkoorden afleidt uit opnames. De features waarmee gewerkt wordt zijn zogenaamde chromaprofielen. Ze stellen de intensiteit van het audiosignaal per toonklasse voor. Deze chromaprofielen worden akoestisch gemodelleerd met behulp van templates (geidealiseerde chromaprofielen) en een afstandsmaat. Naast deze akoestisch modellen en de contextmodellen uit het eerste deel zijn ook nog duurmodellen vereist. Die laatsten zorgen er voor dat de gewenste gemiddelde duur voor de akkoorden en de toonaarden bekomen wordt.

Met het bekomen systeem worden vervolgens experimenten uitgevoerd die inzicht verschaffen in de bijdrage van elke systeemcomponent tot de kwaliteit van de bekomen akkoorden en toonaarden. Hiervoor wordt de complexiteit van het systeem langzaam verhoogd, te beginnen met een systeem dat enkel een akoestisch model van de features bevat en dat dan eerst met duurmodellen en vervolgens met contextmodellen wordt uitgebreid. Uit de experimenten blijkt dat het in rekening brengen van de gemiddelde akkoord- en toonaardduur essentieel is om aanvaardbare resultaten te bekomen voor zowel akkoord- als toonaardextractie. Het gebruik van contextuele informatie heeft daarentegen een wisselend effect. Enerzijds is de positieve invloed van een akkoordwisselmodel op de gevonden akkoorden beperkt (twee à drie procentpunten), maar wel grotendeels onafhankelijk van de gebruikte variant van het akkoordwisselmodel. Anderzijds heeft dit model een groter potentieel om de toonaarduitvoer te verbeteren (tot zeventien procentpunten), maar enkel op voorwaarde dat de variant van het model goed aangepast is aan het geteste muziekmateriaal. Tenslotte blijkt de invloed van het toonaardwisselmodel in al zijn varianten een verwaarloosbaar effect te hebben.

In het laatste deel van deze thesis worden enkele uitbreidingen van het gepresenteerde systeem voorgesteld en geëvalueerd. Eerst wordt de globale waarde voor de gemiddelde akkoordduur vervangen door variabele waarden die afhangen van de specifieke toonaard-akkoordcombinatie, wat een gunstig effect heeft op de toonaardprecisie. Vervolgens wordt het HMM systeem gewijzigd zodat de a priori akkoordduurdistributie niet langer een geometrische distributie is, maar beter de distributie van de geobserveerde duur in een geschikte dataset benadert. Deze wijziging leidt tot een kleine verbetering van de akkoordprecisie, maar vereist uiteraard dat een passende dataset met akkoordannotaties beschikbaar is om de vereiste duurdistributie uit af te leiden. Het laatste experiment toont aan dat het

vergroten van de reikwijdte van het akkoordcontextmodel slechts statistisch niet significante verbeteringen teweegbrengt. Bovendien stijgt de benodigde rekenkracht aanzienlijk.

Acknowledgments

As I write this, it is awards season, so it only seems fitting that I too take a minute to thank my supporting cast, without whom none of this work would have been possible.

First of all, in the role of the drum teacher in *Whiplash*, my promotor Jean-Pierre Martens. Although his teaching methods were a bit more friendly and conventional than in the film, he still managed to push me to the top of my game. Thanks for all the advice and the fruitful discussions. An honourable mention goes to the fact that he knows surprisingly little about music (but all the more about engineering), which forced me to always make my reasoning mathematically sound. Musical intuition never was a convincing answer for him.

Then, as the complementary half of *The Producers*, my co-promotor Marc Leman. Without him, the field of research at the intersection of music and engineering would never have existed at Ghent University. Thanks for the interesting cooperation and the refreshing insights.

Furthermore, I really enjoyed my time with the *Jets* of ELIS (does that mean IPEM are the *Sharks*?). Thanks to all colleagues, past and present, for the shared moments in and outside the DSSP lab. A special mention goes to *Blues Brother* Matthias, my office mate, who was usually the first victim when I was looking for feedback on an idea.

I'm also very glad that I had the opportunity to take my research *On the road*. Thanks to the people at LaBRI, especially Thomas Rocher, for welcoming me in their warm, sunny offices in Bordeaux. *La vita è bella* at IRCAM in Paris too. Thanks to Geoffroy Peeters and all people there for making it such an amazing place to work in.

There are also many people that did not contribute directly to my pro-

fessional life, but ensured that I could clear my head from time to time to fill it again with bigger and better ideas. I'm talking about the folks *Straight Outta Compton*, the famous neighbourhood in Schoten. Then there are the people from my *School of Rock* (and electrical engineering) and the *Grease* gang spread out over Gent, Paris or wherever you have moved in the meantime. And for *Saturday Night Fever*, I know I can always count on the border-defying group known as room65. Thanks all for your support.

Last but not least is the expanding *Von Trapp family*. Thanks for everything and keep on singing on the stairs.

Johan Pauwels

Contents

Abstract **I**

Samenvatting **V**

Acknowledgments **IX**

Nomenclature **XVII**

List of Figures **XXI**

List of Tables **XXIV**

1 Introduction **1**
1.1 Situating key and audio estimation from audio 1
1.1.1 Different ways of storing music 1
1.1.2 Levels of abstraction in symbolic notation 3
1.1.3 Keys and chords 5
1.1.4 The interdependence between keys and chords . . . 6
1.2 Applications of key and chord estimation 8
1.2.1 The role of symbolic annotations in an application . . 8
1.2.2 Categories of applications 9
1.3 Main contributions 12
1.4 Thesis outline 14

2 Music theoretical background information **15**
2.1 Musical notes and physical tones 15

2.2 Octaves and chromas 17
2.3 Pitch intervals between notes 20
2.4 Chroma intervals 21
2.5 Spatial representations of chroma 21
2.6 Chords and keys 24
2.6.1 Chords 25
2.6.2 Keys 26
2.6.3 The interdependence between keys and chords . . . 28
2.7 Reasoning about keys and chords together 29
2.7.1 Relative chords in a key 29
2.7.2 Diatonic chords in a key 32
2.7.3 The diatonic circle 33

3 Past and current trends in key and chord estimation 35
3.1 An overview of systems according to output 35
3.1.1 Separate key and chord estimation systems 36
3.1.2 Combined key and chord estimation systems 36
3.1.3 Joint estimation with other musicological concepts . 38
3.1.4 Related work in the symbolic domain 39
3.2 A unifying view on key and chord estimation 40
3.2.1 Common frameworks for feature decoding 41
3.2.2 General tendencies 43
3.3 A discussion of system components 44
3.3.1 Feature extraction 44
3.3.1.1 Logarithmic spectrum calculation 45
3.3.1.2 A multitude of chroma calculations 47
3.3.1.3 Feature usage 51
3.3.1.4 Feature timing 52
3.3.2 Feature decoding 53
3.3.2.1 Knowledge-based acoustic models 54
3.3.2.2 Data-driven acoustic models 55
3.3.2.3 Durational aspect 56
3.3.2.4 Contextual aspect 57

4 Modelling musicological knowledge 61
4.1 Introduction 61
4.2 A case for relative models 64
4.2.1 Relative chord change models 65
4.2.2 Relative key change models 66
4.3 Constructing relative chord change models 68
4.3.1 Learning models from a symbolic data set using Kneser-Ney smoothing 68
4.3.2 Deducing models from Lerdahl's chord distance . . . 73
4.3.2.1 Lerdahl's chord distance 74
4.3.2.2 Converting distances to probabilities 76

4.4 Constructing a key change model 77
4.4.1 Lerdahl's regional distance 78
4.4.2 From distance to probability 80
4.5 Data sets 81
4.6 Quantifying model fitness: perplexity 86
4.6.1 Quantifying the benefits of context size 86
4.6.2 Quantifying model specificity 87
4.7 Conclusion 91

5 A probabilistic system for estimating chords and local keys 93
5.1 Architecture of the system 93
5.1.1 Feature extraction 94
5.1.2 Probabilistic framework 95
5.1.2.1 Acoustic models 101
5.1.2.2 Duration models 103
5.2 Experimental evaluation 104
5.2.1 Evaluation measures 104
5.2.2 Impact of the feature extraction 105
5.2.3 Impact of the duration models 108
5.2.4 Impact of chord change models 112
5.2.4.1 Adding a chord change model 112
5.2.4.2 Error analysis 116
5.2.4.3 Without key acoustic model 121
5.2.5 Impact of key change models 123
5.3 Conclusion 125

6 A detailed investigation of duration modelling 127
6.1 Relative chord-specific prior mean durations 127
6.2 Alternative chord duration distributions 130
6.2.1 Introducing explicit duration hidden Markov models 130
6.2.2 Using negative binomial prior chord distributions . . 133
6.2.3 Optimising EDHMM decoding through path pruning 135
6.2.4 Experimental evaluation 136
6.3 Conclusion 140

7 Increasing the scope of the musicological context 141
7.1 A probabilistic trigram framework 141
7.2 Exploiting musicological constraints to reduce computational requirements 145
7.3 Experimental results 148
7.4 Conclusion 150

8 Conclusions and perspectives **153**
8.1 Summary 153
8.2 Conclusions 154
8.3 Future research prospects 155
8.3.1 Extending the chord or key vocabulary 155
8.3.2 Focussing on acoustic modelling and feature design . 156
8.4 Further processing of estimated key and chord sequences . . 157

A Mapping of chord types to four triad types **161**

B List of songs in the SEMA data set **163**

Bibliography **169**

Nomenclature

12-TET	12 tone equal temperament
α_k, α_c	balancing parameters for key acoustic model and chord acoustic model
β_k, β_c	balancing parameters for key change model and chord change model
$\mathbf{C}$	chord vocabulary
$c_n \in \mathbf{C}$	chord variable at time step n, consisting of a root $r \in \mathbf{P}$ and a chord type $p \in \mathbf{T}$: $c_n = (r_n, p_n)$
$\mathbf{C}'$	relative chord vocabulary
$\mathbf{x}_n$	feature/observation at time step n
$c'_n \in \mathbf{C}'$	relative chord variable at time step n, i.e. a chord interpreted in a key
$\mathbf{D}(m), \mathbf{D}(k)$	set of relative/absolute chords that are diatonic in mode m/key k
$\mathcal{D}(a, b)$	pitch interval between note $a \in \mathbf{N}$ and note $b \in \mathbf{N}$
$\overline{\mathcal{D}_c}$	mean Lerdahl chord distance
$\overline{d_c}$	mean chord duration
δ	Viterbi forward variable

$\overline{\mathcal{D}_k}$	mean Lerdahl regional distance
$\overline{d_k}$	mean key duration
$\mathcal{D}_{LC}(a, b; m)$	Lerdahl's chord distance between relative chord a and relative chord b in mode m
$\mathcal{D}_{LR}(a, l, m)$	Lerdahl's regional distance for tonic interval a with source mode l and target mode m
$\mathcal{D}_{\text{up}}(a, b)$	upwards chroma interval between chroma $a \in \mathbf{P}$ and chroma $b \in \mathbf{P}$
$f_{kc2a}, f_{kc2b}, f_{kc2c}$	different variants of the model that calculates the bigram key change probabilities
$f_{kc3a}, \ldots, f_{kc3c}$	different variants of the model that calculates the trigram key change probabilities
$i^{(tr)}$	upwards chroma interval between tonic $a \in \mathbf{P}$ and root $b \in \mathbf{P}$: $i^{(tr)} = \mathcal{D}_{\text{up}}(a, b)$
$i^{(tt)}$	upwards chroma interval between two tonics $a, b \in \mathbf{P}$: $i^{(tt)} = \mathcal{D}_{\text{up}}(a, b)$
$\mathbf{K}$	key vocabulary
$k_n \in \mathbf{K}$	key variable at time step n, consisting of a tonic $t \in \mathbf{P}$ and a mode $m \in \mathbf{M}$: $k_n = (t_n, m_n)$
$\mathbf{M}$	mode vocabulary
$\mathbf{N}$	collection of all notes
N_c	number of (considered) chords
$N_d(m)$	number of diatonic chords in mode m
N_k	number of (considered) keys
N_m	number of (considered) modes
$N_{nd}(m)$	number of non-diatonic chords in mode m
$N_u(m)$	number of unseen relative chords in mode m
N_x	number of observations
$\mathbf{P}$	collection of all chromas
P_{cc2}, P_{cc3}	bigram/trigram chord change probabilities

P_{cd}	bigram chord duration probabilities
P_{kc2}, P_{kc3}	bigram/trigram key change probabilities
P_{kd}	bigram key duration probabilities
$\underline{\mathbf{S}}$	data set, i.e. a collection of music pieces $\underline{\mathbf{s}}_{\mathbf{y}}$
S	smoothing size
$\mathbf{T}$	chord type vocabulary
T_a	analysis shift, time between frames during spectral analysis
T_x	observation shift, time between features

List of Figures

1.1 Flash card reporting a conversation at a bakery 4

2.1 The pitch helix proposed by Shepard and its projection onto the chromatic circle 18
2.2 Chromatic circle 23
2.3 Circle of fifths 24
2.4 The difference between considering a key-chord pair as two separate entities or as a key-relative chord pair where the root is expressed relative to the tonic. The latter representation is easier to move around because it contains only one absolute chroma such that changes to key and chord always stay synchronised. 31
2.5 Constructing diatonic chords 33
2.6 The diatonic circle 34

4.1 "Perdido" by Juan Tizol 63
4.2 Preprocessing of chord labels 69
4.3 Example of counting bigrams and trigrams in an annotation 71
4.4 Lerdahl's basic space for IImin in major 75
4.5 Lerdahl's basic space for Vmaj in major. The numbers in boldface indicate the chroma interval differences with respect to the basic space of IImin in major (seen in figure 4.4) 76
4.6 Lerdahl's basic space for a major mode starting key 79

4.7 Lerdahl's basic space for the major mode target key a perfect fifth higher than the starting key. The numbers in boldface indicate the chroma interval differences with respect to the basic space of a major mode starting key (seen in figure 4.6) 79
4.8 Model perplexities for the major and minor chord change models derived from three different data sets 87
4.9 Test set perplexity of the theoretical model in function of the probability of a diatonic chord 89

5.1 Schematic of the smoothing procedure. The black circles represents the chroma vectors calculated on all frames, and alternating grey circles the resulting smoothed chroma vectors that are used as observations in the probabilistic framework. The smoothing windows used for the computation of the latter are indicated by boxes, here for $S = 3$, where only the shaded boxes need to be computed. The upper pane shows the situation when the observation shift is equal to the analysis shift ($T_x = T_a$), the lower when the observation shift is equal to S times the analysis shift ($T_x = ST_a$). 96
5.2 Influence of smoothing size on the SEMA chord results for two chord acoustic models and two smoothing approaches . 106
5.3 Influence of smoothing sizes on short chords 107
5.4 Influence of acoustic model balance factors on the SEMA set 110
5.5 Influence of the chord acoustic model balance factor on short chords 111
5.6 Chord estimation output on the Isophonics data broken down into categories 117
5.7 Key estimation output on the Isophonics data broken down into categories 118

6.1 Aggregated difference in key and chord scores obtained on the SEMA set by replacing the global mean chord duration by relative chord specific durations retrieved from the SEMA and Isophonics sets 129
6.2 Aggregated difference in key and chord scores obtained on the Isophonics set by replacing the global mean chord duration by relative chord specific durations retrieved from the SEMA and Isophonics sets 129
6.3 Expanded state topologies leading to different combined duration distributions 131
6.4 The class of negative binomial distributions 135
6.5 Chord duration histograms with geometric and best-fitting negative binomial distributions 137
6.6 Influence of the chord duration distribution shape on SEMA data set 138

6.7 The evolution of key and chord results in function of a changing chord duration model. Each negative binomial shape was tested with four chord change model configurations on the Isophonics data set. The baseline geometric duration model is shown as $\rho = 1$. 139

List of Tables

2.1 The most common names for the twelve chromas 19
2.2 Pitch intervals in 12-TET. The first twelve double as chroma intervals. 22
2.3 Conversion between distance in semitones and in circle steps 24
2.4 Common chord types (a) and modes (b) 27
2.5 The degrees and names of the corresponding chroma interval starting from the tonic for two modes 32

4.1 Resulting counts for the example sequence in figure 4.3 . . . 70
4.2 Lerdahl's regional distance for all combinations of modes and chroma intervals between the tonics 80
4.3 Number of retained n-gram tokens in the back-off model per data set 83
4.4 Number of unique n-gram types per data set 83
4.5 Common bigrams in three data sets 84
4.6 Common trigrams in three data sets 85
4.7 Mean validation set perplexities per data set and model order and corresponding free parameter values 88
4.8 Test set perplexities for all combinations of models and test sets 90

5.1 Binary templates for four chord types 101
5.2 Temperley templates for four modes 103
5.3 Chord results on the SEMA set 109
5.4 Key results on the SEMA set 109

5.5 Key and chord score of the configurations that maximise chord (left column) or key score (right column) for different chord change models. The best results per chord change model are set in boldface. 114
5.6 Key and chord score of the configurations without key acoustic model that maximise chord (left column) or key score (right column) for different chord change models. The best results per chord change model are set in boldface. The difference with respect to the corresponding configurations with key acoustic model is displayed in parenthesis. 122

7.1 All pairwise combinations of key and chord variables in a trigram system with their proportion of occurrence 147
7.2 Results of the optimal configurations with different chord change models on the SEMA set (a) and on the Isophonics set (b). The best results per chord change model are set in boldface. The differences with respect to the equivalent bigram configurations are displayed in parenthesis, followed by the percentage of music pieces whose output is changed. 151

1 Introduction

This introductory chapter will explain what exactly is meant by the estimation of keys and chords from audio and what it can be used for. Furthermore, the main contributions of this thesis will be discussed and finally, an overview is given of the remainder of the text.

1.1 Situating key and audio estimation from audio

1.1.1 Different ways of storing music

Listening to music does not require a special skill, it can be done by every hearing person, regardless of his musical education. People are also able to describe in natural language what they have been listening to, but these descriptions are rather vague and generally inconsistent. When two persons are asked to create sounds based on the description made by a third, these sounds will seldom sound remotely similar. However, trained musicians are able to do exactly this, name what they have been hearing (even if it is just in their head) in a way such that the description can be passed along to others who can then consistently replicate what was meant. This kind of accurate representation is called a *symbolic (musical) notation*.

Many types of symbolic notation exist, ranging from simple, coarse descriptions to complex, detailed ones. The simple ones just give a broad characteristic of a piece, whereas the most detailed ones can be used by

a performing musician to recreate the described piece in a way that it is recognisable as such, without requiring that he himself has heard the original beforehand. The most known example is sheet music, which is one of the more detailed examples. Because of the infinite number of subtle variations in music, no representation exists that is accurate enough to reproduce every single nuance. Instead, each type of notation emphasises different aspects of a piece by making different trade-offs between what is more and what is less accurately described. A general approach to keeping a description manageable, is to describe but a limited number of characteristics that are then discretised into a finite number of categories, to constrain the number of symbols. This explains why there is such a multitude of notations: different applications require descriptions that are adapted in content and granularity to different purposes.

Symbolic notation, going back as far as 2000 BC, used to be the only way to store music until the invention of the *audio recording* with Edison's phonograph in 1877. In some ways, audio recordings are the opposite of symbolic notation. Whereas even the most detailed symbolic notation allows for some freedom in interpretation due to its incompleteness, a recording consistently recreates music in exactly the same way, time after time. However, it is a single, holistic description which is not aware of its composing parts. Symbolic notation on the other hand is self-aware. It consists of a description of what is going on in the music piece on multiple levels, with a meaningful hierarchy between its constituting components. This is apparent in the sense that it is easy in symbolic notation to make changes to a single musical aspect without changing the other aspect. E.g. one can change the instrumentation or tempo, while it is hard to impossible to do that in an audio recording. That is why symbolic notation is said to be a *semantically rich* description, i.e. the meaning of the music it represents is readily apparent, while audio recordings are *semantically poor*, i.e. meaning is absent in the representation. Because of their high fidelity and ease of use, audio recordings have become the dominant means to store music for playback since the 1920s. Nevertheless, symbolic music notations persist because the semantically rich information they provide is needed for a number of other applications.

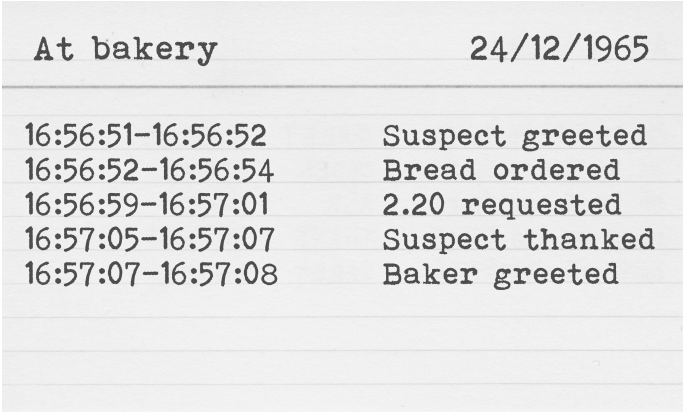
We can conclude that music can be stored either in the symbolic domain or in the audio domain, each with their advantages and disadvantages. Luckily, these domains are not isolated from each other. It is perfectly possible to convert music from one domain to the other. We are especially interested in the conversion of semantically poor, high definition audio recordings into semantically rich symbolic notations, a process called *music transcription*. This conversion can be easily done by humans, using a skill called musical hearing, although some musical training might be required once the intended symbolic notation goes beyond the most trivial ones (e.g. labelling music as either "slow" or "fast"). The idea here is to automate this human capability in order to reap the benefits of automa-

tion: scalability, consistency and cost effectiveness. These properties make it possible to open up a whole new range of applications that would not be feasible when relying on manual labour. Looking beyond the musical domain, other fields of study are also concerned with semantic audio analysis, such as speech recognition and the detection of well-defined events in audio, ranging from goals in sports commentary to gun shots in security footage. On an even higher level, semantic audio analysis is part of the larger scope of content based analysis, where an analysis can be performed on signals other than audio, such as moving or still images, biometric data or a combination thereof. The corresponding applications include facial recognition in pictures and heart-rate monitoring, among others.

1.1.2 Levels of abstraction in symbolic notation

In order to illustrate the different types of symbolic notation, with their distinct levels of abstraction, let us draw a parallel to natural speech. Here too we have two very different types of media to store it. On one hand, we have a symbolic notation, namely text, which is good at capturing meaning, but does not allow to perfectly reconstruct speech because it looses a number of characteristics of the voice, such as intonation. And on the other hand, the same recording technology used for music can also store speech in a perfectly reproducible way. We can also make an exact transcription of what is said in a recording, thereby transitioning between the semantically rich and semantically poor domains. A word for word transcription is the most detailed way to store speech in the symbolic domain, but this level of abstraction is not always appropriate for the intended application. A condensed textual representation, a summary, can be more useful for certain applications. Take the case of a student attending a lecture. When he later wants to study the content of that lecture, it is much more useful for him to have notes that are a synopsis of what has been said, instead of the exact wording of the lecturer. Such notes can exist on different levels of abstraction: the initial notes taken during the lecture, which contain most of the information in shorthand, and a general outline made afterwards, which only contains the most important points. For this example, the final textual representation will still contain a lot of variation, because the topic of a lecture can be anything, but for other textual representations used in more specific cases, limiting syntax and vocabulary of the summary can be helpful.

Let us imagine that a non-specified governmental agency is investigating a money laundering case. A person is suspected of trafficking money to accomplices in cover-up shops. The agency now wants to find out which of the shops on High Street are involved and have planted a bug on the suspect. At one point, the government agent on listening duty might hear the following scene:


At bakery 24/12/1965

16:56:51-16:56:52	Suspect greeted
16:56:52-16:56:54	Bread ordered
16:56:59-16:57:01	2.20 requested
16:57:05-16:57:07	Suspect thanked
16:57:07-16:57:08	Baker greeted

Figure 1.1: Flash card reporting a conversation at a bakery

- "Good morning"
- "One rye bread, please."
- "That will be 2.20 euro."
- <sound of money being handed over>
- "Thanks, have a nice day."
- "Bye."

The agency now wants to keep a record of every visit to every shop. This could easily be done by constantly recording the suspect, but the agency has decided that this is not the most practical way of storage for them. At one point in the future, they would like to compare all the transactions to see if patterns arise in these seemingly innocent visits and if they can be linked to the money laundering. They do not want to listen to all these conversations over and over again, but want the essential information in a readily available form. Therefore they have come up with a strict procedure. The officer on duty needs to summarise every conversation in every shop on a flash card according to a well defined scheme. Then they can pin all these flash cards to a whiteboard and start drawing arrows between them and all other things detectives usually do. As an example, the card that has been created from the aforementioned conversation is displayed in figure 1.1.

As you can see, the transcripts follow a well defined set of rules. Every line of conversation is summarised by a two word "noun-verb" scheme together with precise timing information. The entire dialogue is also described by a single "preposition-noun" scheme that describes the overall setting of the conversation. This type of note-taking system is especially

adapted to its intended use: it provides a good description of the aspects of the scene that are deemed important, namely an overview of the dialogue without getting lost in the exact wording, but completely ignoring other, less important aspects, such as the tone of the voices and the background noises. It is therefore tailored to this specific type of use, but would be insufficient for other cases, such as writing down a comedy sketch, where the exact words and how they are spoken are at least as important as the general meaning of what is said. We could come up with other schemes to describe a dialogue for other use-cases, such as a script for a radio drama, which would focus on other aspects and would require a different level of detail, but this one contains exactly the right level of abstraction for its intended purpose.

1.1.3 Keys and chords

Similar to the rigid dialogue summary scheme, musical notation in terms of chords or keys follows a well-defined syntax and serves a specific purpose. Together, they do not constitute the most thorough notation of music: instrumentation is not included, nor is there an exact description of each separate note, for instance. Sheet music contains more details, although some aspects are still ignored, and can be considered the equivalent of full text. Key and chords describe one specific aspect of a music piece, namely the harmony. All information that does not contribute to the harmony is left out, while all essential information is retained and made more explicit in comparison to a list of notes. Consequently, this type of music description is only applicable in the context of Western tonal music, which is based on harmony. It therefore makes no sense to use chord and key notation with Indian music or contemporary electro-acoustic music, just as you do not tell a joke by only listing key words.

In our flash card example, the "noun-verb" descriptors take on the same role as chords in music: they are concise descriptions of audio segments with a certain time span. In fact, not every point in time has such a description associated with it, e.g. the sound of money has none. The audio segments can vary in duration, and form a first level of abstraction of the exact words that are chosen and how they are spoken. The description is not exact enough to reconstruct the specific wording, but the abstraction is however close enough to the actual events such that an exemplary instance can be given for each exchange, without changing its meaning significantly. Instead of "good morning", "hello" or "good day" could be used, and baguettes or raisin bread could be bought instead of rye bread, all without fundamentally changing the situation. A more detailed description would only be distracting for its specific use.

The scene as a whole on the other hand, can be summarised on a higher time-scale as "at bakery". This is equivalent in abstraction level to a mu-

sical key. Starting from this description, it is nearly impossible to reconstruct the entire conversation: there is too much possible variation. A customer can order multiple items, with multiple price tags, engage himself in smalltalk with the baker, and so on. Nonetheless, it is a valid description of the scene, and it can be used to distinguish itself from other scenes, like "at butcher's" or "at mechanic". The abstraction level is therefore strongly tied to a temporal scale: a group of words in a summary corresponds to a larger span of time than a group of words of the same length in a full text transcription. A more formal definition of keys and chords will follow in the next chapter.

The process of *key and chord estimation from audio* is thus one specific type of music transcription, where the target symbolic notation consists of keys and chords. Although we call it a *transcription* when done by humans, in the context of research towards automating this process, the term *estimation* is often used to underline the imperfect nature of the current technology. The difference between key and chord estimation and related transcription tasks targeting other semantically rich musical notations, such as beat extraction and genre classification, is that we can use domain specific knowledge about harmony to facilitate the task. The many forms of knowledge and the sources they come from will be the topic of the following chapters. Later on in this introductory chapter, we will discuss some of the possible applications of key and chord estimation.

A final note on the estimation process concerns the time scale. In some of the most common symbolic notations, such as sheet music, the duration of the symbols is quantised into a limited number of categories and expressed relative to the overall tempo (crotchets or quarter notes, quavers or eight notes, etc.). In actual performances, a substantial part of the expressiveness of music comes from small deviations from these discretised durations. So this quantisation can be seen as an extra level of abstraction. We do not make this quantisation in this work. Instead, all timing information is expressed exactly in seconds. In this regard, the result of our proposed conversion process is more similar to symbolic MIDI notation. Working with digital systems, a certain discretisation is of course inevitable, but if it is sufficiently precise, the timing can be considered continuous.

1.1.4 The interdependence between keys and chords

A recurring theme in this thesis is the study of how prior knowledge can be used in the estimation of keys and chords from audio. Such knowledge is also deeply ingrained in humans. Through our exposition to music, a certain musical norm is established: we have a clear expectancy of what "real music" is and this notion is much narrower than all the theoretically possible combinations of notes. This is illustrated by the fact that some

music genres such as be-bop or trash metal are not appreciated by everyone. These genres are called "difficult", "unaccessible" or plainly "not music". In reality, these genres simply strain the listener's notion of music too much, and this varies strongly between individuals. Even though there is a personal part to it, some preferences arise that are more universal. Just like a greeting does not happen in the middle of a conversation, there are some rules in music that are rarely violated. The scholarly study of *music theory* is concerned with capturing this subliminal knowledge.

We will try to quantify this knowledge, by capturing parts of it in statistical models and by interpreting parts in the context of the theory of classical harmony. This quantified knowledge will then be used to facilitate the estimation of keys and chords from audio. The function of this extra information is slightly different for keys than for chords. Chords on the one hand, are close to acoustic events, so in theory they can be estimated using just the local acoustic properties. A musicological knowledge model can only help to resolve ambiguities, to discriminate between melody and accompaniment notes, and to recover from errors. Translated to our wiretapping example, suppose that the wireless connection to the suspect is noisy, and at one point the officer hears the following, incomplete line: "One rye **ead, please". It is then very likely that the officer will write down that the suspect ordered "bread", not "head" or "lead", because that is what is expected in the context of a bakery. Neither will he note down the words of the song played on the radio in the background, because it is obvious from the context that this is not relevant. Chord knowledge can similarly resolve noise, present in the signal itself or arising from a faulty initial estimation. Finding a key on the other hand, requires by definition more than local observations, so a musicological knowledge model can be used to decide on the time interval in which the key is constant. In our example, this means that it should be decided when there is a change in location and how the location should be named.

The fact that keys and chords both describe the musical content, but on different abstraction levels, leads to a challenging circularity in modelling harmony. If the key is known, some assumptions can be made about the chords, but the reverse is also true. Suppose that it has already been established that our suspect has entered a bakery, then it is much more likely that the subsequent conversation will contain phrases such as "One baguette, please" or "Would you like a bag with that?". It would be very unlikely to hear a sentence such as "Could you check the oil level?" in such a setting. But upon hearing the latter phrase, it will be clear that it is very likely that the suspect has entered a garage. In itself, all possible phrases can be uttered in all possible situations, but one has to be very absent-minded to ask for a tyre change at a butcher's. There is just a large difference in probability that the latter is actually going to happen. Exactly the same can be said for the relation between keys and chords: any combination is possible, but some are more likely than others. The signi-

ficance of a certain observation in establishing a particular context is also variable, and not necessarily tied to probability. For example, chit-chatting about the weather is common, but not indicative for a specific shop. On the other hand, a line like "One baguette, please" will strongly suggest that the subject is standing in a bakery, and simultaneously weaken the evidence that he is at the car mechanic. Other sentences such as "Would you like a bag with that?" are not very helpful to decide whether the suspect is at a bakery or at a butcher's shop, but can very likely rule out that he is at the mechanic.

1.2 Applications of key and chord estimation

1.2.1 The role of symbolic annotations in an application

A particularity of all but the most trivial symbolic musical notation, is that it is understood by a far smaller percentage of the population than other symbolic notations, such as the alphabet. Natural language-like descriptions such as "fast, energetic music" are still understandable for all, but using the specific musical vocabulary, which includes keys and chords, requires some degree of formal musical education. Therefore the raw output of key and chord symbols is not meaningful for everyone. This allows us to divide applications of key and chord estimation into *direct applications* versus *indirect applications*. For applications of the former category, the interaction with the final product is done in terms of chords and/or keys themselves. In the latter, estimated chord and/or key sequences are only used, without the end-user realising it, as an intermediate representation of a music piece in order to reach a different goal. The purpose of key and chord estimation here is that a recording is first converted into a semantically rich representation before information retrieval techniques are applied, which is much easier on symbolic notation than on audio itself. One of the consequences is that indirect applications have the potential to reach a much larger audience.

In addition to being understood by only a part of the population, being able to read a symbolic notation does not automatically mean that one can listen to music and write down what has been played. This again contrasts with natural language, where any hearing person who can read can also write text transcriptions. Being able to read a symbolic representation evidently is a prerequisite for being able to create it by listening to music. However, training one's hearing such that it can perceive what has to

be represented takes a considerable amount of time, except for some rare, gifted people. So a substantial portion of amateur musicians can read a symbolic notation, but would nevertheless benefit greatly from an automatic generation of that notation from audio. Its appeal lies in the fact that they can then use this automatic transcription to learn how to play any song they have a recording of, instead of needing to look first for a man-made transcription, which might be expensive or unavailable altogether. In our specific case of chord estimation, the end product is a so-called chord sheet, which contains just the sequence of chords that is played in a song. This description is not exact enough to precisely recreate the piece of music, which is deemed necessary for classical music, but it is sufficient as a compact notation for jazz and pop music, which has a larger tradition of interpreting songs differently and making personal adaptations.

A second way to separate the different applications, is the distinction between *small-scale* and *large-scale applications*. The former means that the application only needs to extract chords and/or keys from a single piece of music in order to be useful, while large-scale applications are only able to fulfil their goal if a large set of music has been processed, often the larger the better. Of course, producing already useful results for a single file does not prevent a small-scale application from processing large data sets.

1.2.2 Categories of applications

The combination of the two previously introduced distinctions allows us to split the type of applications into four categories. This grouping will facilitate a systematic exploration of the kinds of applications in the following paragraphs. If any commercial implementations of these types of application already exist, they will be mentioned under the appropriate category. Otherwise, some ideas for potential applications will be outlined. Note that this categorisation of applications is not specifically tied to chord and key estimation, but can be applied to any task of music transcription.

Direct, small-scale applications

Driven by the great number of amateur musicians who can read music notation, but cannot transcribe from audio themselves, the most obvious direct application is automatic chord transcription. A number of commercially available products are therefore competing on this market. Some of these are marketed as stand-alone programs, such as D'Accord's "iChords"[1], Luxand Development's "Chord Pickout"[2], MusProjects' "AnySong Chord

[1] `http://www.daccordmusic.com/eng/site/application.php?idProduto=42`
[2] `http://www.chordpickout.com`

Recognition"[3] , EUMLab's "Chordec"[4], Martian Storm's "Chord Detector"[5]. Others are built as a web service, such as "Chordify"[6] or are integrated in more extensive tools for learning music, such as PG Music's "Chord Tool" (included with "Band-in-a-box"[7]), SuperMegaUltraGroovy's "Capo"[8] or "Riffstation"[9]. A number of programs acknowledge the imperfection of automatic transcription beforehand, and position their application merely as an aid for manual transcription, requiring extra guidance such as the indication of chord boundaries. In addition, almost all programs provide an editor for manual correction of the automatically generated results. Performing a rigorous evaluation on their output is not possible because of the closed nature of their file formats and/or the absence of absolute timing information, but most of them fall into one of the following two cases. Either its makers are closely associated with a research group that has published the underlying algorithm, or it is clear enough from informal experiments that their performance leaves much to be desired for. But even for state-of-the-art research spinoffs, independent reviewers find the performance of the estimation still lacking[10]. Most of the reviewers praise the idea, but not the implementation.

The audio that is being transcribed does not have to come from a third-party. Even when a user is playing himself, and therefore already knows what he is playing, automatic transcription can be useful. Its purpose is then to take notes of what is being played, avoiding the manual labour of writing down chords on paper or entering them into a computer program.

Another use for chords in an direct way can be seen in the music training app "Yousician"[11]. Here, a user is presented with a chord sequence that he tries to play to the best of his abilities. The sound is recorded and converted back into symbolic notation, which then gets compared with the target sequence. The user then gets points according to how well he played. The app keeps track of your progress and turns the learning process into a visually attractive game.

Indirect, small-scale applications

In indirect, small-scale applications, the generated key and chord sequences of a piece of music control another process. One possibility is that the chords or keys control playback of supplemental audio, e.g. for generating automatic accompaniment. This ensures that the accompaniment is

[3] http://www.musprojects.com/anysong-chord-recognition/
[4] http://eumlab.com/
[5] http://www.chord-detector.com
[6] http://chordify.net
[7] http://pgmusic.com/bbwin.htm
[8] http://supermegaultragroovy.com/products/capo/
[9] http://www.riffstation.com
[10] http://www.gizmag.com/riffstation-guitar-learning-review/27705/
[11] http://get.yousician.com/

always in perfect harmony with the played music. The musician can then change his playing on the spot and improvise at will, and the accompaniment will follow. Otherwise, if a prerecorded accompaniment is used, the musician is tied to follow exactly what has been planned beforehand, limiting his freedom of expression. Processes beyond the audio domain can also be controlled, such as the colour and intensity of stage lights or other types of visualisation.

Direct, large-scale applications

The automation of key and chord estimation allows us to build large databases of symbolic music. Enriched with meta-data like the title, composer, genre and year of the piece, such databases can answer many questions posed in musicological research. For example, one can find the most typical chords or keys for a certain composer or genre, study the evolution of chord usage over time or look for relations using knowledge discovery techniques. Another possibility is the retrieval of information that matches a query, such as finding all chord or key sequences in a database that are similar to a given snippet of symbolic music.

A specific example of a retrieval application that has seen multiple commercial implementations is key estimation for deejays. The idea is that a deejay uses automatic key estimation to assign a key label to all of the songs in his collection before the start of his deejay set. He can then use the key of the currently playing song as a query to list all song in his collection with the same key. The songs returned by the program are then all good candidates for the next song, guaranteed to fit well after the current song. The deejay can use this information to make a more informed decision such that he is sure that overlapping parts in the mix will sound pleasant together. Because deejays typically have collections that are so big that labelling them all would require a lot of manual labour and because a great number of them lack the training to do so themselves anyway, they form an ideal public for automatic key estimation. It is therefore not surprising that all major deejay software packages include such functionality as part of their program. Examples are Native Instruments' "Traktor"[12], "Serato DJ"[13], MixVibes' "Cross DJ"[14] or "MixMeister Studio"[15]. In addition, this functionality is also available in some stand-alone, deejay-oriented media managers, e.g. "Mixed In Key"[16], MixShare's "Rapid Evolution"[17], Tagtraum Industries' "BeaTunes"[18] or Pioneer's "rekordbox"[19]. Reviews by inde-

[12] http://www.native-instruments.com/en/products/traktor/dj-software/traktor-pro-2/
[13] http://serato.com/dj
[14] http://www.mixvibes.com
[15] http://mixmeister.com/products-mmstu7.php
[16] http://www.mixedinkey.com
[17] http://www.mixshare.com/software.html
[18] http://beatunes.com
[19] http://rekordbox.com/

pendent deejay websites show that the best of these products achieve a correct key recognition of more than 90 % on typical dance music[20], but this quickly drops to less than 45 % for general pop music[21].

Indirect, large-scale applications

Indirect, large-scale applications also make use of a large database of symbolic music, just like direct, large-scale applications do, but the interaction of the end-user with that database does not involve manipulating keys or chords. These applications can often be seen as the combination of a direct, large-scale application with some additional automation that interprets the key or chord information and derives another property from it.

For instance, a database such as the one used to suggest songs in a similar key to deejays, can be combined with logic that groups songs based on key and possibly other properties, leading to automatic playlist generation. The query is then no longer the key of the currently playing song, but the song itself and it is completely transparent to the user that the resulting list of compatible songs is based on them having the same key. The advantage is that the end-user no longer needs to be able to interpret musical notation to make use of the application, but the challenge lies in the fact that the additional decision logic should approach the same level of quality and creativity as a human.

Another example where queries no longer need to be specified in musical notation, is the retrieval of cover versions of a specified recording. Key and chord estimation are in these cases used at two stages: once for filling a database beforehand with semantically rich information and another time to convert the query recording into the same high-level representation, after which the answer can be found using traditional pattern matching techniques.

1.3 Main contributions

In this thesis, the task of automatically estimating keys and chords from audio is discussed. In particular, the focus lies on exploiting musicological knowledge about keys and chords in the course of that estimation. This is complementary to studying keys and chords as acoustic events, which is equally necessary to come to a working system. The main novelty of this work is situated in that it addresses the following points:

- How can the interdependency between keys and chords be modelled in a consistent way?

[20] `http://www.djtechtools.com/2014/01/14/key-detection-software-comparison-2014-edition/`
[21] `http://www.djtechtools.com/2012/01/26/key-detection-software-showdown-2012-edition/`

- How can knowledge about key and chord sequences coming from different sources be combined?
- How can the resulting knowledge be integrated into a system for automatic key and chord estimation from audio?
- How much do the separate components of the model contribute to the overall performance?
- How can information about expected chord durations be used in the system and how does it improve results?
- How can the context information in consecutive chords be exploited and how does this affects the performance?

The current text is the culmination of previous work that has been published in the following papers, in chronological order of publication:

- Johan Pauwels and Jean-Pierre Martens: "Integrating musicological knowledge into a probabilistic system for chord and key extraction" in *Proceedings of the 128th Convention of the Audio Engineering Society (AES)*, 2010.
- Johan Pauwels, Jean-Pierre Martens and Marc Leman: "Improving the key extraction accuracy of a simultaneous local key and chord estimation system" in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, 2011.
- Johan Pauwels, Jean-Pierre Martens and Marc Leman: "Modeling musicological information as trigrams in a system for simultaneous chord and local key extraction", in *Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, 2011.
- Johan Pauwels, Jean-Pierre Martens and Marc Leman: "The influence of chord duration modeling on chord and local key extraction", in *Proceedings of the IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2011.
- Johan Pauwels and Jean-Pierre Martens: "Combining musicological knowledge about chords and keys in a simultaneous chord and local key estimation system" in *Journal of New Music Research (JNMR)*, 43(3), 2014.

The methods and results presented in these publications have been used to formulate the unifying framework that is presented in this text and that supersedes all previous work.

If the reader is interested in reconstructing how the final proposition emerged, and in finding the motivations behind some of the directions, he is advised to read these preceding publications in the given order. The

reason is that there are some dependencies between the papers. In particular, the original research for the AES paper has been continued and led to the additional results proposed at ICME. Both have been combined and extended to form the JNMR paper. The work in the MLSP and ICMLA papers are both further extensions of this system, but in orthogonal directions, and can therefore be read separately from each other.

1.4 Thesis outline

The remainder of this text is constructed as follows. First, the background knowledge that is necessary to make the rest of this thesis comprehensible is introduced. This part first provides an overview of the music theory concepts and associated terminology in chapter 2. Then a summary of related work on key and chord extraction from audio is given in chapter 3. Special attention is paid to the way the relationship between chords and keys is incorporated in the systems under discussion.

Next, the proposed approach to modelling sequences of chords and keys is discussed in chapter 4. A distinction is made between models learned on a set of symbolic annotations and models derived from a musicological theory. The data sets used in this work are also introduced here. The predictive power of the resulting models is then evaluated in relation to the data sets.

In chapter 5, a system for the simultaneous extraction of chords and keys is proposed. It is introduced gradually, evolving from a simple system that only uses acoustic information to a more complex one encompassing more musically relevant components, among which the models of the preceding chapter. A detailed evaluation after each step allow us to quantise the benefits of each added component.

Subsequently, the duration model of the system resulting from chapter 5 is extended in chapter 6 in two ways. First, the prior chord duration is made dependent on the relative chord-mode combination. Afterwards, a solution is presented that changes the prior chord duration model from a standard geometric distribution to a musicologically more plausible distribution.

A further extension is examined in chapter 7, where the more advanced musicological models as calculated in chapter 4 are incorporated into the system. The necessary adaptation of the search procedure is explained and an evaluation of the resulting system is performed in order to assess how well the theoretically more predictive models translate into an increase in key and chord extraction results.

Finally, some conclusive remarks are drawn in chapter 8 and possible directions for future work are touched upon.

2 Music theoretical background information

Throughout the centuries, a rich language has been developed to describe musical concepts. However, no strict definitions have been established for these terms and often the same terms are reused in multiple situations. They are therefore ambiguous and rely on the context to be fully comprehensible. This might be acceptable for day-to-day, informal conversations, but in a scientific work like this, it is undesirable. Moreover, this loose usage of language makes it unnecessarily complex for novices to get a basic grasp of music theory. Therefore we will start by defining all musical terms used in the remainder of this work, in order to establish a strict frame of reference. We have tried to stray as little as possible from the common consensus, but where deemed necessary, clarity was preferred over conformity, for instance by retaining only one meaning in cases of terminology reuse. Consequently, the reach of these definitions is strictly confined to this text. This section also serves as a musical primer for readers not familiar with music theory. Lastly, mathematical notations will be introduced for some concepts such that they can be used in the more formula-heavy approach described in the following chapters.

2.1 Musical notes and physical tones

As we have already mentioned in the introduction, there are two worlds of music that we would like to reconcile in this work. On the one hand, there is the acoustical side of the story where music is seen as a series of changes in air pressure, which can be fully captured by a music recording. On the

other hand, there is the symbolic side which aims to encode those pressure waves into meaningful symbolical descriptions. Because these symbols need to be descriptive for humans, human perception is an integral part of this notation process.

We see music in the symbolic domain as a sequence of notes, where a *note* is a perceptual unit of music which is completely defined by 4 properties: pitch, duration, loudness and timbre. The *pitch* is the perceived height of the note and allows it to be ordered on a scale from low to high. The *duration* is the amount of time the note is audible (falling above the hearing threshold). The *loudness* of the note is its perceived intensity. Finally, the *timbre* is the colour of the note, grouping all aspects which distinguish notes of the same pitch, duration and loudness from each other.

The acoustic counterpart of a symbolic note is a *(physical) tone*. From now on, we will consider a *music recording* as a sequence of tones. A tone is here defined as a sound composed of one or more sinusoidal components whose frequencies are related to each other by an integer, a so-called *harmonic relation*. It is perceived as a single sound rather than a compound of separate components. Not every sound is a tone however. The sound of a strike on a cymbal or other untuned percussion, for example, does not consist of sinusoidal components. It therefore is not a tone, and consequently is not perceived as a note by humans. On the other hand, all notes have a physical realisation as a tone when they are produced by an instrument or a voice.

Each of the four note properties has an equivalent tone property (Fletcher, 1934), although their relation is not always very straightforward. The pitch of a note is related to the *fundamental frequency* of a tone, defined as the frequency of which every other frequency is an integer multiple. The other sinusoidal components are called harmonics. The duration of a note can be linked to the *duration* of the physical excitation, but taking the hearing threshold into account. The loudness is related to the *energy* of a tone, but this relation is non-linear and depends on the frequency, among others. Equal-loudness contours, first measured by Fletcher and Munson (1933), try to express this for an average listener. The timbre can be linked to the *relation between the energies of the sinusoidal components* and their dependence over time. Some special cases of a tone can then be seen as having a particular timbre, like a tone where the fundamental itself is absent (a virtual fundamental frequency) or a pure tone, where only the fundamental is present.

According to our definition, a tone can have any fundamental frequency, such that there are an infinite number of tones between two frequencies. On the other hand, the number of different pitches that a note can take is well-defined. Although there is no formal limit placed on the pitch range, in practice this is limited by the range of human hearing. In mathematical terms, the number of pitches is countably infinite in theory, but countable in practice. Which fundamental frequencies are considered as leading

to the perception of a note and which ones are not, is determined by a *tuning (system)*. A tuning is a convention about the number and the positions of fundamental frequencies on the frequency scale that get assigned to pitches of notes. It can thus be seen as a discretisation of the frequency axis. Many tunings have been proposed and used through the centuries. Some of them are based more on physical properties than others. Their adoption is also culture dependent. When a tone is produced with a fundamental frequency that is not on a position allowed by the tuning (and the deviation is larger than the just noticeable difference), it is perceived as out-of-tune.

2.2 Octaves and chromas

Both in acoustics and in musicology, the term *octave* is used in virtually the same way. Two tones are said to be an octave apart when their fundamental frequencies have a ratio of 2:1. These tones lead to the perception of two notes and the distance between them is also named an octave. Two notes that are an octave apart are perceived to be more similar than any other note pair, a phenomenon which is called the *octave circularity* or *octave equivalence*. This phenomenon is so dominant that it is reflected in all tuning systems. Concretely, this means that a tuning is defined as the division of an octave into a number of sections, expressed as proportions of the octave. This division is then replicated over all octaves[1]. For this reason, Shepard (1964) proposed to represent pitch perception as a two-dimensional helix structure rather than a one-dimensional line. This pitch helix can be seen in figure 2.1. We will call the vertical and the angular dimension the *register* and the *chroma* respectively. Notes whose pitches are one or multiple octaves apart are thus said to have the same chroma, or alternatively, they belong to the same *pitch class*. This bi-dimensional representation of pitch, such as "$C4$" or "c'", has been already apparent in Western musical notation long before Shepard's research. We can clearly distinguish separate notations for chroma ("C" or "c") and register ("4" or "$'$").

Since the 18th century, the *12-tone equal temperament (12-TET)* is the de facto standard tuning for Western music, so logically, it is also the tuning used in this work. It is obtained by dividing an octave into twelve parts, equally distributed on the logarithmic frequency axis. Consequently, there are twelve different chromas. The frequencies of the 12-tone equal tem-

[1] In theory, nothing stops you from coming up with a tuning system where the spacing of notes is different from octave to octave, or even a system that ignores the octave equivalence altogether, but this is so counterintuitive to human perception (and the underlying physical harmonic series) that this is not done in practice.

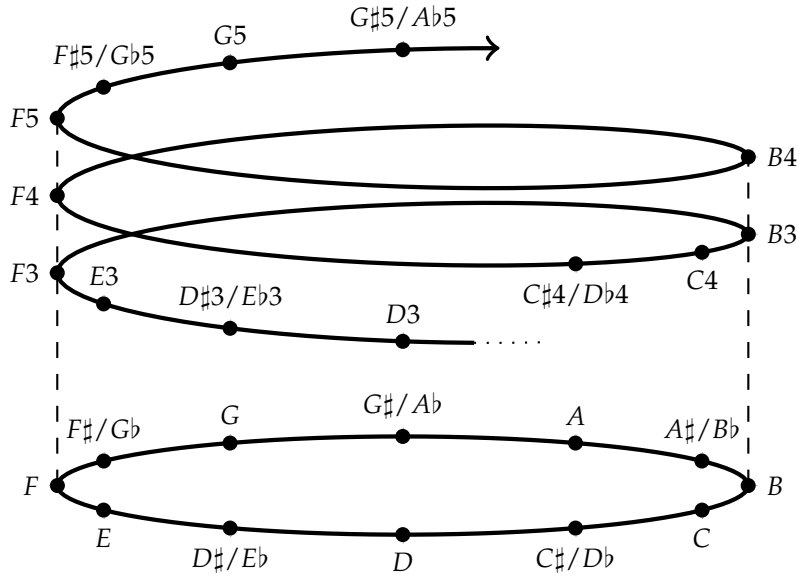


Figure 2.1: The pitch helix proposed by Shepard and its projection onto the chromatic circle

perament are

$$f = 2^{\frac{z}{12}} f_{\text{ref}}, \ \forall z \in \mathbb{Z} \tag{2.1}$$

Nowadays, the reference frequency f_{ref} from which the other notes are derived is set to 440 Hz most of the time, which even became an ISO standard (1975), but historically this has varied widely.

The distance between two adjacent frequencies is called a *semitone*. The major benefit of the 12-TET tuning is that it keeps the distance between two adjacent notes constant. This greatly simplifies the calculation of distances between notes, a simplification on which we will rely heavily for the following definitions. This also means that some of those definitions do not hold for other tunings.

Chromas can be represented using a simple index as follows

$$index = \left(12 \log_2 \left(\frac{f}{f_{\text{ref}}}\right)\right) \bmod 12 \tag{2.2}$$

where the reference frequency gets assigned the index 0 and the others range up to 11. Most commonly however, a letter notation is used. Strikingly enough, this letter notation does not assign a single, unique name to each chroma. Rather, seven out of the twelve chromas are assigned a letter from A to G, starting with A for the chroma of index 0. In addition, two modifiers are introduced: the *sharp* ♯ which increases the chroma by

index	sharps	naturals	flats
0		A	
1	$A\sharp$		$B\flat$
2		B	$C\flat$
3	$B\sharp$	C	
4	$C\sharp$		$D\flat$
5		D	
6	$D\sharp$		$E\flat$
7		E	$F\flat$
8	$E\sharp$	F	
9	$F\sharp$		$G\flat$
10		G	
11	$G\sharp$		$A\flat$

Table 2.1: The most common names for the twelve chromas

one semitone and the *flat* $\flat$ which decreases the chroma by one semitone. The collective musical name for these modifiers is *accidentals*. Combining *natural names*, defined as names without accidentals, with modifiers permits us to give a name to the remaining five chromas, but it also implies that one chroma can have multiple names[2], as can be seen in table 2.1. The reason that these five chromas can only be named indirectly is entirely due to historical reasons. Under no circumstances should this be interpreted as a distinction between more and less important chromas. Moreover, just because a chroma can be named using a simple natural name, does not mean that it is always called that way, e.g. a "B" chroma is still called "$C\flat$" in some cases. Which name is used when is determined by a set of music theory rules that take the context into account, but a priori all names are used, because they imply different musical relations. Two different letter notations referring to the same chroma are said to be *enharmonically equivalent*. They only differ in their *spelling*. In the remainder of this work, the spelling will be ignored under certain circumstances. If this is the case, every chroma name can be replaced by an equivalent name.

The register of a note is often expressed by referring to a reference octave. Therefore, the pitch axis is divided into sections of one octave wide which get assigned a number. Curiously, the convention is to let these sections range from one C to the next C, lower boundary included, rather than from A to A. The register of a note is then indicated by giving the number of the octave which encloses its natural note[3]. According to the ASA

[2]It is also possible to apply the modifiers multiple times which leads to a whole new set of alternative names for the chromas, e.g. $B\flat\flat$ as an alternative name for the chroma with index 0. This causes every chroma to have multiple possible names, even though it does not appear so in the table because it lists only the most common ones.

[3]The fact that the register number is determined based on its natural note name, before an

scientific pitch notation (Young, 1939), the octave ranging from 261.6 Hz to 523.2 Hz gets number 4 assigned to it. It thus ranges from $C4$ to $C5$, such that the reference note of 440 Hz is called $A4$. Another convention is the Helmholtz notation, where the reference octaves are not indicated by numbers but by typographic changes to the chroma letters, e.g. "C" or "c''".

Mathematically, the collection of all notes will be represented as $\mathbf{N}$ and the collection of all twelve chromas as $\mathbf{P}$, such that $a \in \mathbf{N}$ and $b \in \mathbf{P}$ means that the variable a represents a note and b a chroma.

2.3 Pitch intervals between notes

The pitch distance between two notes is called a *(pitch) interval* (and is unrelated to the mathematical notion of interval). We have already encountered the most important interval, the octave, in the previous section. The other intervals are divided into those that are smaller than an octave, which we call *simple intervals*, and those that are larger, called *compound intervals*. The latter are called this way because they can be represented as the sum of one or more octaves and a simple interval. We notate the pitch interval between two notes a and b as $\mathcal{D}(a, b), \forall a, b \in \mathbf{N}$.

Intervals can be expressed in a number of ways, much like a distance between two points can be expressed in multiple units, such as centimetres or inches. A first option is to count the number of semitones between the notes, e.g. $\mathcal{D}\left(E\flat 4, G4\right) = 4$. This representation is not able to take the difference between enharmonically equivalent notes into account however: both $\mathcal{D}\left(E\flat 4, G4\right) = 4$ and $\mathcal{D}\left(D\sharp 4, G4\right) = 4$. Although for some uses, like ours, this might be an advantage rather than a disadvantage. A more general option is to give the interval a name. This name consists of two parts: a *generic (interval) number* and its *quality*. The generic number counts the number of natural chroma names, thus without accidental, that lie between the two notes, start and end note included. For example, the generic number of the interval between $D\sharp 4$ and $G4$ is 4 (D-E-F-G, four natural chroma names in between), whereas it is three for the interval between $E\flat 4$ and $G4$. The quality can be either *perfect*, *major*, *minor*, *diminished* or *augmented*, which gets abbreviated as P, M, m, d or A, and it depends on the number of semitones between the notes. The quality therefore serves to distinguish between intervals with the same generic number, but where the accidentals of the constituting notes differ. For instance, the intervals between $D\sharp 4$-$G4$ and $D\sharp 4$-$G\sharp 4$ both have generic number four, but they are a diminished fourth ($d4$) and a perfect fourth ($P4$) respectively. Not every combination

optional modifier is applied, implies that notes of the same pitch do not necessarily have the same register number, a B3 note has the same pitch as C♭4 for instance.

of generic number and quality gives a valid interval, however. A table of allowed combinations can be found in table 2.2 together with the equivalent distance in semitones. The final names for our example intervals are thus $\mathcal{D}\left(E\flat 4, G4\right) = M3$ and $\mathcal{D}\left(D\sharp 4, G4\right) = d4$. It can be seen from these examples that the multiple spellings for naming a note cause intervals to have multiple possible names too, known likewise as *enharmonically equivalent* intervals. Note that these are aurally indistinguishable because the constituting notes are the same, only named differently[4]. Analogous to note names, we will encounter situations in which the spelling of intervals is ignored in the remainder of the text.

2.4 Chroma intervals

A pitch distance can also be defined between chromas. But because the register information is lost, the relative positions of both chromas on the frequency axis becomes ambiguous. In our definition of the *upwards chroma interval* $\mathcal{D}_{\text{up}}(a, b), \forall a, b \in \mathbf{P}$ between two chromas a, b, we always assume that the second chroma b falls within the octave above a. The distance between them is therefore always a simple interval: $P1 \leq \mathcal{D}_{\text{up}}(a, b) < P8$ when expressing the chroma interval as a name or $0 \leq \mathcal{D}_{\text{up}}(a, b) < 12$ when expressing it as a number of semitones (the top half of table 2.2). A chroma interval thus becomes directional, $\mathcal{D}_{\text{up}}(a, b) \neq \mathcal{D}_{\text{up}}(b, a)$, unlike a pitch interval, whose direction is determined by the pitch values itself[5], $\mathcal{D}(a, b) = \mathcal{D}(b, a), \forall a, b \in \mathbf{N}$.

2.5 Spatial representations of chroma

From the spatial organisation of notes as a helix and the definition of chromas, it directly follows that chromas can be represented on a circle that arises from the projection of the helix on the ground plane. This can be seen in figure 2.1. The order of the note axis inside an octave is preserved and every chroma appears only once. It does not matter which chroma is used as the starting point, they all lead to the same result. This arrangement of chromas is called the *chromatic circle*. The distance in semitones between

[4]This is the main instance where the choice for 12-TET tuning greatly simplifies music theory. It both explains the popularity of 12-TET and the reason why there are multiple names for the same notes and intervals in the first place: they are not necessarily the same in other tunings.

[5]To be complete, we can also define the downwards chroma interval $\mathcal{D}_{\text{down}}(a, b)$ where b is assumed to fall within the octave below a. The upwards and downwards chroma interval then form symmetrical pairs $\mathcal{D}_{\text{up}}(a, b) = \mathcal{D}_{\text{down}}(b, a)$, but we only require the former to understand the remainder of this text.

semitones	interval names		
0		pure unison ($P1$)	diminished second ($d2$)
1	augmented unison ($A1$)	minor second ($m2$)	
2		major second ($M2$)	diminished third ($d3$)
3	augmented second ($A2$)	minor third ($m3$)	
4		major third ($M3$)	diminished fourth ($d4$)
5	augmented third ($A3$)	perfect fourth ($P4$)	
6	augmented fourth ($A4$)		diminished fifth ($d5$)
7		perfect fifth ($P5$)	diminished sixth ($d6$)
8	augmented fifth ($A5$)	minor sixth ($m6$)	
9		major sixth ($M6$)	diminished seventh ($d7$)
10	augmented sixth ($A6$)	minor seventh ($m7$)	
11		major seventh ($M7$)	diminished octave ($d8$)
12	augmented seventh ($A7$)	pure octave ($P8$)	diminished ninth ($d9$)
13	augmented octave ($A8$)	minor ninth ($m9$)	
14		major ninth ($M9$)	diminished tenth ($d10$)
15	augmented ninth ($A9$)	minor tenth ($m10$)	
16		major tenth ($M10$)	diminished eleventh ($d11$)
17	augmented tenth ($A10$)	perfect eleventh ($P11$)	
18	augmented eleventh ($A11$)		diminished twelfth ($d12$)
19		perfect twelfth ($P12$)	diminished thirteenth ($d13$)
20	augmented twelfth ($A12$)	minor thirteenth ($m13$)	
21		major thirteenth ($M13$)	diminished fourteenth ($d14$)
22	augmented thirteenth ($A13$)	minor fourteenth ($m14$)	
23		major fourteenth ($M14$)	diminished double octave ($d15$)

Table 2.2: Pitch intervals in 12-TET. The first twelve double as chroma intervals.

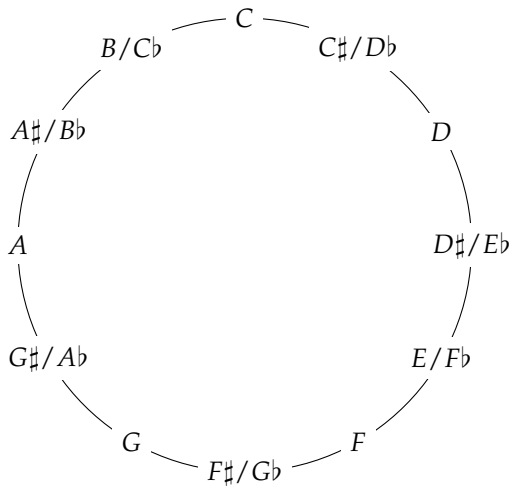


Figure 2.2: Chromatic circle

two chromas can be counted by traversing in a clockwise direction along the circumference of this circle, given that two chromas are spaced one semitone apart. A graphical depiction can be seen in figure 2.2.

The chromatic circle is not the only spatial organisation of chromas that is perceptually motivated. An alternative configuration is based on the interval of a perfect fifth ($P5$), whose notes sound very well together and therefore are perceptually close, second only to an octave relationship. This can be explained by looking at the ratio of two consecutive harmonics of a tone. They form a sequence of the form $\frac{2}{1}, \frac{3}{2}, \frac{4}{3}, \ldots$ and the ratio $\frac{3}{2}$ is almost (for 12-TET) or even exactly (for some other tunings) equal to the ratio of the fundamental frequencies of two notes a perfect fifth apart. Because a perfect fifth equals seven semitones, this ratio for 12-TET is equal to

$$\Delta f_{P5} = \frac{2^{\frac{i+7}{12}} f_{\text{ref}}}{2^{\frac{i}{12}} f_{\text{ref}}} = 2^{\frac{7}{12}} \approx 1.498 \; \forall i \in \mathbb{Z} \tag{2.3}$$

As a consequence, two notes a perfect fifth apart will have a large number of overlapping harmonics, which gives the impression that they are perceptually close. When we form a chain of chromas increasing by a perfect fifth from one to the next, all chromas are encountered exactly once before arriving back at the starting chroma. The circular figure that appears in this way is called the *circle of fifths*, shown in figure 2.3. Adjacent chromas are said to be one step on the circle of fifths apart, or one *circle step* for short.

The circle of fifths gives us another way to numerically express a chroma interval, in addition to the distance in semitones which is counted on the

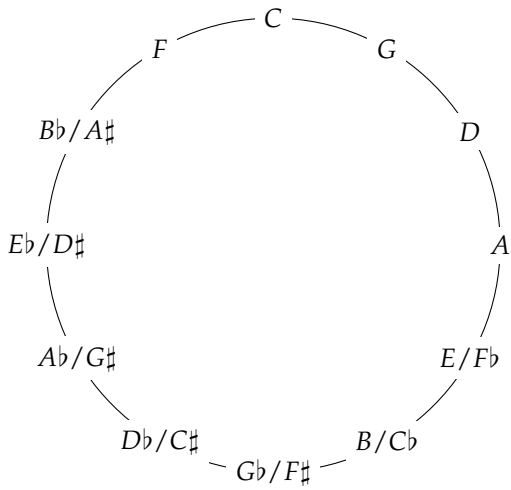


Figure 2.3: Circle of fifths

semitones	0	1	2	3	4	5	6	7	8	9	10	11
circle steps	0	7	2	9	4	11	6	1	8	3	10	5

Table 2.3: Conversion between distance in semitones and in circle steps

chromatic circle. We can likewise count the number of circle steps on the circle of fifths, again in the clockwise direction for an upwards chroma interval. This number does not distinguish between enharmonically equivalent chromas, like the number of semitones in between them. These two representations must be seen as different numerical expressions of the same musical distance, as they can unambiguously be converted from one to another. Each of them emphasises a different perceptual relation and a corresponding spatial organisation. A conversion table can be found in table 2.3.

2.6 Chords and keys

In a music piece, notes do not appear in isolation, but together. The study of how notes sound together is called *harmony*, from the Latin "harmonia", signifying agreement. It encompasses both the study of notes that sound simultaneously, forming a *chord*, and of notes sounding sequentially, forming a *melody*. These two terms are also called the vertical and the horizontal components of harmony, in allusion to the horizontal time axis. Most West-

ern music contains both aspects of harmony, where chords provide an accompaniment over which a melody is played. However, the distinction between which notes belong to a chord accompaniment and which ones belong to the melody, can be questionable in some cases. These two aspects converge in the study of the broader musical context, established by both chords and/or melody notes in succession. This broader musical context is called a *key*.

2.6.1 Chords

We define a *chord* as an unordered set of 2 or more simultaneously sounding chromas coming from at least 3 different notes. A chord is thus a compact representation of a collection of simultaneously sounding notes, where the register and number of played notes is lost. The extra information needed to expand a chord into a collection of notes, i.e. the number of notes of each pitch class and their register, is called the *chord voicing*. A subset of the voicing information is the *inversion* of a chord. It just states the chroma of the lowest note in the chord, without giving its exact register nor specifying whether it is repeated in any higher registers. In our definition of chord estimation, the goal is only to obtain the chord information from a recording, not the inversion or complete voicing.

Suppose we have the collection of notes $\{C4, E4, G4, C5\}$. It consists of 4 notes with 3 different chromas, so it complies with our definition of a chord. Specifically, the chord would be $\{C, E, G\}$. The order of the chromas does not matter, but we naturally need to write them from left to right in some order, so $\{E, G, C\}$ or $\{C, G, E\}$ would be equivalent notations. Counterexamples of a chord are $\{C3, C4, C5, C6\}$ or $\{C4, E4\}$, where the reason for excluding them as chord is the fact that the collection of notes only contains one chroma, or that there are only two notes respectively. The voicing information to get back to the example note collection we started from is $\{C \to 4, 5; E \to 4; G \to 4\}$, but other voicings of $\{C, E, G\}$ lead to other note collections such as $\{E3, C4, E4, G4\}$ or $\{G6, C7, G7, C8, E8\}$. This clearly illustrates the loss of information associated with labelling a collection of notes as a chord, but the voicing information is not necessary for the study of harmony. So, this is a deliberate choice to get rid of superfluous information.

Without loss of generality, one chroma (which does not necessarily need to be a part of the chord, but in most cases it is) can be picked as a reference chroma and all chord chromas can be expressed as distances to this reference. In this representation, the reference chroma is called the *root* of the chord and the set of chroma intervals is called the *chord type*. These two components make up the name of a chord. If we choose C as the root of our example chord $\{C, E, G\}$, the associated chord type is $\{P1, M3, P5\}$. Other possible root-chord type combinations that describe the same set of

chromas are $E\,\{P1, m3, m6\}$, $G\,\{P1, P4, M6\}$ and $D\,\{M2, P4, m7\}$. Which chroma is chosen as the root depends on the context, so in absence of any contextual information, all representations are equally valid. In practice however, some chord types appear more often than others and if a chord can be written as one of those common chords, it most likely will. We designate the collection of all considered chords as $\mathbf{C}$ and of that of all chord types as $\mathbf{T}$. A variable that represents a chord is therefore symbolised as $c \in \mathbf{C}$, a root chroma variable as $r \in \mathbf{P}$ and a chord type as $p \in \mathbf{T}$, such that $c = (r, p)$.

Chords can be divided into categories according to the number of chromas they are made of. Three-chroma chords are called *trichords* and four-chroma chords are called *tetrachords*. Both trichords and tetrachords contain a subset of common chords fulfilling an extra constraint, namely those formed by stacking intervals of a third on top of each other. These chords are said to be *triadic* and are perceived as very pleasant combinations of chromas. Therefore most music consists predominantly of triadic chords. A triadic trichord is called a *triad* and similarly, a triadic tetrachord is a *tetrad*. Triadic chords can be easily recognised because the generic numbers of their composing chroma intervals form the sequence $1, 3, 5, (7)$.

Representing a chord type as a collection of chroma intervals might be the most unambiguous and complete way, but it is also rather long and cumbersome. The most common chroma interval collections therefore get a name assigned to them. Some of the most popular chord type names are listed in table 2.4a together with their abbreviations as defined by Harte et al. (2005). Our example collection of chromas $\{C, E, G\}$ will therefore most commonly be interpreted as a Cmaj chord, because the alternatives for the root lead to chord types that are absent from this list of common chord types. These other representations are not wrong however, and could be preferred in specific circumstances, given enough support from the context. It is not always that obvious to decide what is the most appropriate chord representation. For example, the chromas $\{A, C, E, G\}$ can be interpreted as either an Amin7 or a Cmaj6 chord, which are both common. The context will be decisive in choosing one over the other in this case.

2.6.2 Keys

The broader musical context called *key* introduces a certain hierarchy to the chromas, centred around a principal chroma, called the *tonic*. This tonal centre is perceived as the most stable chroma over a period of time. It acts as a centre of balance around which other chromas shift. Therefore the tonic is more often than not the last note of a music piece, because it has a sense of finality and completion. Given this tonic, it is decided for every other chroma whether it *belongs* to the key (which is always true for the tonic) or not. The subset of chromas that belong to a key form the basic

chord type name	abbreviation	chroma intervals (as names)	chroma intervals (in semitones)
major	maj	{P1, M3, P5}	{0, 4, 7}
minor	min	{P1, m3, P5}	{0, 3, 7}
diminished	dim	{P1, m3, d5}	{0, 3, 6}
augmented	aug	{P1, M3, A5}	{0, 4, 8}
major 7th	maj7	{P1, M3, P5, M7}	{0, 4, 7, 11}
dominant 7th	7	{P1, M3, P5, m7}	{0, 4, 7, 10}
minor 7th	min7	{P1, m3, P5, m7}	{0, 3, 7, 10}
half diminished 7th	hdim7	{P1, m3, d5, m7}	{0, 3, 6, 10}
(full) diminished 7th	dim7	{P1, m3, d5, d7}	{0, 3, 6, 9}
major 6th	maj6	{P1, M3, P5, M6}	{0, 4, 7, 9}
minor 6th	min6	{P1, m3, P5, m6}	{0, 3, 7, 9}

(a)

mode name	chroma intervals (as names)	chroma intervals (in semitones)
major	{P1, M2, M3, P4, P5, M6, M7}	{0, 2, 4, 5, 7, 9, 11}
minor	{P1, M2, m3, P4, P5, m6, m7}	{0, 2, 3, 5, 7, 8, 10}

(b)

Table 2.4: Common chord types (a) and modes (b)

building blocks for constructing melodies and chords in that key. Chromas not belonging to the key are therefore far less likely to appear in music played in that key than chromas that do belong to it. Similar to a chord type, the collection of chromas that belong to a key is represented as a set of upwards chroma intervals starting from the tonic. Together they form the *mode* of the key. The most common modes are listed in table 2.4b with their name.

The complete name of a key is formed by combining the letter notating of the tonic with the mode name. Example keys are therefore "$G\sharp$ major" and "B minor". Unfortunately, there is some overlap between the mode and the chord type names, but we will try to minimise confusion by always abbreviating chord type names, e.g. "maj", and always writing mode names in full, e.g. "major". We represent the collection of all keys as $\mathbf{K}$ and the collection of all modes as $\mathbf{M}$. A key variable is mathematically represented as $k \in \mathbf{K}$, and its tonic and mode as $t \in \mathbf{P}$ and $m \in \mathbf{M}$, such that $k = (t, m)$.

2.6.3 The interdependence between keys and chords

Chords and keys are examples of symbolic labels that convey a certain meaning. Both describe aspects of harmony, i.e. they both interpret notes by aggregating certain aspects and ignoring others. The two terms are distinguished from each other by their difference in time scale, and accordingly, in abstraction level. A chord describes a blend of notes at a single point in time, so it covers multiple simultaneously sounding notes. Therefore, there is a direct relation between the physical tones that are sounding at that single moment and the chord label. A key on the other hand, is concerned with longer time spans containing multiple notes in succession, not necessarily overlapping in time. It is therefore a higher level descriptor that cannot be used to represent a single time instance, only a sequence of notes.

This difference in abstraction and temporal level between chords and keys can be illustrated by the following observation. Speaking in the strictest sense, it is not possible to play any chord or key, only notes can be played by generating the corresponding physical tones. Neither chords nor keys contain enough information to do this unambiguously. However, when asked to produce a certain chord, a musician will fulfil this request without hesitation. He will then play a number of simultaneously sounding notes that are exemplary for that chord label, but there is no unique answer. Its exact manifestation will vary depending on the musician's background and instrument. Other options are equally valid due to the incomplete chord description, but all will sound reasonably similar. You can therefore play *a* chord of a given label, but not *the* chord.

On the other hand, when asked to play a certain key, a musician will say that this is impossible. No single combination of notes is exemplary enough to represent the myriad of possible variations. Moreover, establishing a sense of key requires a longer exposure over time. In other words, a chord can be created by a single action of a musician, but a key requires multiple actions. Musicians therefore do not speak about playing *a* key with a certain label, only about playing multiple notes or chords *in* a key. So when creating music from symbolic notation, a key has multiple orders of magnitude more possible appearances with a larger variety than a chord, but the latter still has a non-trivial amount of different appearances.

The concepts of keys and chords are strongly intertwined. On the one hand, a key indicates a preference over certain chromas, such that it is more likely that those chords are played that contain only chromas belonging to the key. These chords are said to be *diatonic* with respect to the key. On the other hand, a sequence of chords gives a strong indication of what the encompassing key might be[6].

At times, a key or a chord will be absent from a music piece. We will designate these special cases as the *none* key and *none* chord. The silence at the beginning and end of a piece is an obvious example of instants without key or chord, but there are some instances where music is audible, yet no chord or key is present. For instance, a melody sung by a single voice has a key but no chords, and a drum solo has neither. We can thus segment a music piece into regions in which the key or chord stays constant and label those regions. The resulting two sequences provide a compact representation of the harmony of the piece. These timed sequences are the objective of key and chord estimation algorithms. Just the succession of chords or keys in a piece, without timing information, are also called the *chord changes* and the *key changes*. A key change is also referred to as a *modulation*, so there are $N - 1$ modulations in a sequence of N keys.

2.7 Reasoning about keys and chords together

2.7.1 Relative chords in a key

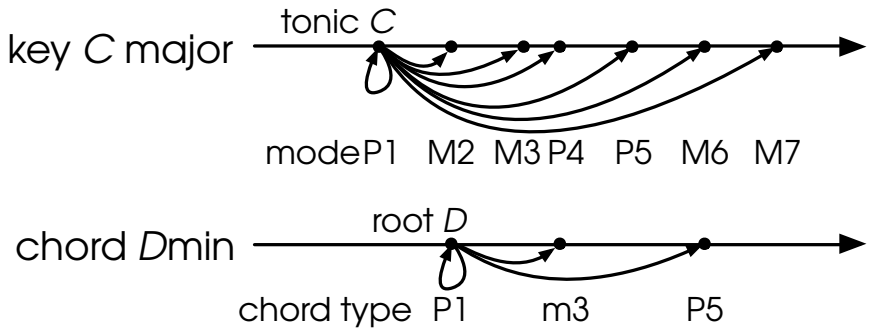
In order to use the most informative representation of music, we always consider keys and chords together in this work. We assume that every

[6]The constituting chromas of all chords in a sequence form a non-uniform histogram in which one can identify the division between chromas that belong to the key and those that do not belong. This becomes clearer when the chord sequence gets longer. Another indication are the chroma intervals between subsequent roots, where certain fixed patterns are strongly indicative for a certain key.

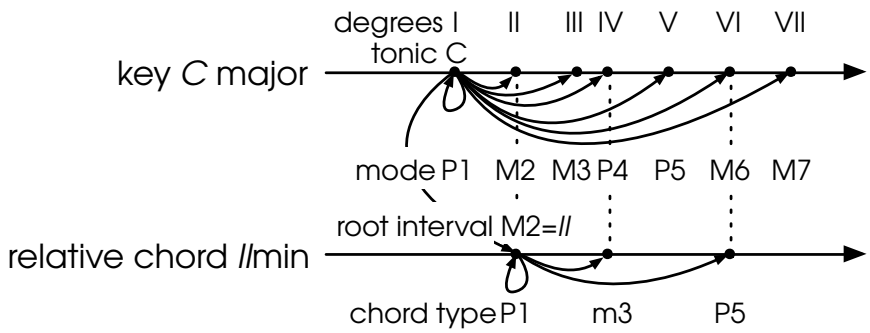
point in time can be labeled with a key k and a chord c (where the key and chord can also be "none"). From the definitions it follows that a key-chord pair (k, c) can equivalently be described as a quadruple (t, m, r, p) of tonic t, mode m, root r and chord type p. The tonic and root represent chromas having an absolute interpretation, whereas the mode and chord type are collections of chroma intervals that are relative to the tonic and the root. An example key-chord combination is depicted in figure 2.4a. We can now, without losing information, replace the absolute root by the chroma interval i between the tonic and the root: $i = \mathcal{D}_{\text{up}}(t, r)$, such that $(t, m, r, p) = (t, m, i, p)$. We will call the combination (i, p) the *relative chord* c'. Now every component of the key-chord combination, except the tonic itself, is expressed in relation to the tonic. The mode and root as distances to the tonic and the chord type as distances to the root (itself is expressed as a distance to the tonic). The corresponding key-relative chord combination of our example is given in figure 2.4b. We thus end up with four different, equivalent representations of the same key-chord combination which can be easily derived from each other: $(k, c) = (t, m, r, p) = (t, m, i, p) = (k, c')$. In cases where confusion can arise between a chord c and a relative chord c', we might underline the difference by calling the former an *absolute chord*.

The reason for introducing the concept of a relative chord, is that this makes the relation between key and chord more explicit. More specifically, it makes it easier to take *transposition invariance* into account. By this we mean the phenomenon that the identity of a music piece stays largely intact when all notes are moved over the same number of semitones up or down (and consequently the keys and chords are also moved over the same distance) (Cuddy and Cohen, 1976). It is easy to see that moving each note a fixed number of semitones is easier in a key-relative chord combination (k, c') than in a key-chord combination (k, c) because the former requires only a single change to the tonic, whereas both the tonic and the root need to be synchronously adapted in the latter representation. For the aforementioned reason, one is mostly concerned with the changes of chords relative to the key when studying harmony. The absolute chroma of the tonic therefore does not matter very much and any musicological knowledge about chord sequences would best be expressed in terms of $(m, i, p) = (m, c')$. This line of thought reflects the way scholars have been analysing harmony for centuries.

The root chroma interval i can be expressed like any chroma interval through its name or its numerical distance, but for upwards chroma intervals starting at the tonic of a given key, a special naming scheme is customary. If the root interval is diatonic in the key, it is represented by its generic number stylised as a Roman number. If the root interval is non-diatonic, it is derived from these Roman numbers by adding modifiers, much like chroma names are derived from the natural chroma names. This representation therefore depends on the mode, which means that it can



(a) A key-chord combination expressed separately with absolute tonic and root



(b) A key-relative chord combination with the root expressed relatively to the tonic

Figure 2.4: The difference between considering a key-chord pair as two separate entities or as a key-relative chord pair where the root is expressed relative to the tonic. The latter representation is easier to move around because it contains only one absolute chroma such that changes to key and chord always stay synchronised.

	I	II	III	IV	V	VI	VII
major	P1	M2	M3	P4	P5	M6	M7
minor	P1	M2	m3	P4	P5	m6	M7

Table 2.5: The degrees and names of the corresponding chroma interval starting from the tonic for two modes

only be used when the key is known. Moreover, the Roman numbers refer to different chroma intervals according to the mode in which they are interpreted. When referring to a chroma relative to the tonic of a key, this chroma is called a *degree* of the key. A list of the corresponding chroma interval names for the degrees of the major and minor mode can be found in table 2.5. Because the major and minor mode differ in the quality of their third and sixth chroma interval, the IIIrd and VIth degree also differ. In our example in figure 2.4b, the root chroma interval M2 can therefore be written as *II*. It is also easy to see that the *D*min chord is diatonic in *C* major, because the chord is entirely composed of key degrees.

2.7.2 Diatonic chords in a key

Degrees take on a principal role in the study of harmony. More specifically, the diatonic triadic chords created on the degrees of a key are very commonly used. They are constructed by considering each of the degrees in turn as the root of a chord, and by stacking thirds on top of this root such that all chromas in the resulting chord are diatonic in that key. For instance, if we want to construct the diatonic triad on the IIIrd degree of the major mode, its root chroma interval must be $M3$, starting from the tonic. By stacking thirds on this root, we get the chord consisting of the chroma intervals $\{M3, P5, M7\}$ upwards from the tonic. This can easily be found by alternating chroma intervals in the first row of table 2.5, starting from the desired root. These chroma intervals are now expressed in reference to the tonic, which can technically be considered as the chord $I\{M3, P5, M7\}$, where the tonic is chosen to be the root. It is however more convenient to change the root to be III, as was intended from the beginning, giving us $III\ \{P1, m3, P5\}$ or IIImin. Therefore the diatonic triad on the IIIrd degree in a major key is always a min chord. The whole process is illustrated in figure 2.5. The resulting chord type depends on the quality of the chroma intervals that form the chord, which themselves are determined by which degree is chosen as the root and therefore varies from degree to degree. The triad on the IVth degree of the major mode, for instance, is always a maj one.

The diatonic chords on the key degrees are closely related to the concept of *harmonic function*. Music can be seen as a constant shift of balance between

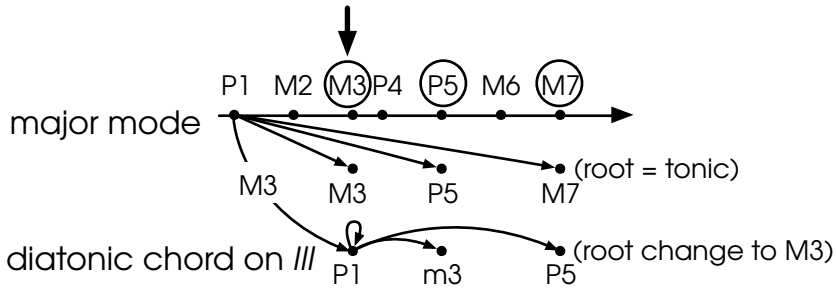


Figure 2.5: Constructing diatonic chords

tension and relaxation (Bigand et al., 1996), between stability and surprise. Some chords are very effective in creating a relaxing effect on the listener, while others cause a more unnerving effect, and this alternation is used to make music interesting. The diatonic chords on the degrees now have a well-defined correspondence to this whole spectrum of effects on the listener, e.g. in the major mode, the chord on the Ist degree is always perceived very stable, whereas the one on the VIIth builds up tension. The *function* of a chord therefore refers to the purpose of that chord in this spectrum of emotional effects. An important part of the study of harmony is to identify the harmonic function of all chords. This can then provide a deeper insight into the music, as it is a sort of explanation of what is going on in the piece. Furthermore, it also gives an indication of how variations of the piece can be created without fundamentally altering the piece. Classically trained composers rely on this reasoning while writing music, as do jazz musicians when they reharmonise a piece to make it novel and more interesting.

2.7.3 The diatonic circle

By now it is clear that the degree representation is strongly connected to the concept of diatonicity. There is one spatial representation that underlines this, which is called the diatonic circle (of fifths). It can be seen in figure 2.6. It is formed by placing the degrees on a circle with a distance of a fifth between adjacent degrees. In contrast to the previous circular constellations, the labels on the circle are no longer the twelve chromas, but the seven degrees of a mode. Another difference with the circle of fifths in figure 2.3, is that the distances clockwise along the circle are not always perfect fifths, but rather those fifths that are required to end up in the next diatonic degree. This depends on the specific mode, because the chroma intervals corresponding to the degrees are also mode-dependent. For example, the distances on the major mode diatonic circle are perfect fifths for

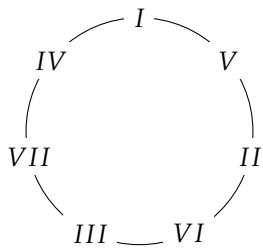


Figure 2.6: The diatonic circle

all except between VII and IV, which is $\mathcal{D}_{\text{up}}(\text{M7}, \text{P4}) = d5$, a diminished fifth. Similarly, on the minor mode diatonic circle all clockwise distances are perfect fifths, except between II and VI, which is $\mathcal{D}_{\text{up}}(\text{M2}, \text{m6}) = \text{d5}$.

As a final note, it should be stressed that the interpretation of a chord as a relative chord in a key is purely a music theoretical concept, intended to reason about the effect that a certain chord sequence has on the listener, which is directly related to his expectancy. It does not have an equivalent acoustic phenomenon. In other words, the key of a piece has no implication on the sound of a chord. One can therefore only speak about a relative chord when the key at that time is known.

3 Past and current trends in key and chord estimation

In this chapter, we review the existing literature about chord and key estimation. Therefore we first divide the existing systems based on the output they produce and focus especially on the way the interdependence between the two is taken into account. Later on, we make abstraction of the specific output generated by the system and discuss both key and chord estimation in a unifying framework. By abstracting away the label type, we can focus on the parallels between key and chord estimation. We can then more easily study the specific components that make up a system and their role in the estimation process.

3.1 An overview of systems according to output

We first make the distinction between systems that handle either keys or chords separately and those that estimate them together. The first group will be called *separate key and chord estimation systems* and they are naturally divided into *key-only estimation systems* and *chord-only estimation systems*. The second group are the *combined key-and-chord estimation systems*. The latter systems incorporate both keys and chords somewhere in the process, either as inputs, as outputs or in between. A first subcategory herein is that of *sequential key-and-chord estimation systems*, where the output of the key estimation is used as supplementary input to the chord estimation, or vice versa. We also include all systems that start from symbolic key or chord

labels, irrespective of how they are obtained (automatically or manually). They can be considered as just the second part of a sequential system. In contrast, *simultaneous key-and-chord estimation systems* estimate both types of symbolic labels in one integrated process. The system we will propose in the subsequent chapters is one of this last subtype.

Another differentiator, specific to key estimation systems, is whether the global key or local keys are estimated. In the former case, a single key label is assigned to the whole music piece. For the latter, the piece is divided into segments of a constant key and each one is labelled. Global estimation has the advantage that it is easier, as there are more observations to base the decision on and no segmentation needs to be done. Its disadvantage is that in songs with key changes, it will inevitably lead to a loss of information, namely the secondary keys and the location of the key changes.

3.1.1 Separate key and chord estimation systems

Automatic key and automatic chord estimation started out as two separate fields. The early attempts at key recognition (Purwins et al., 2000; Pauws, 2004; Gómez and Herrera, 2004) were greatly influenced by earlier efforts on symbolic key estimation (Takeuchi, 1994; Vos and Van Geenen, 1996; Temperley and Sleator, 1999). These, in turn, were developed in close relation with experiments on musical perception, partly in the hope that successful automatic estimation methods would also offer an insight into the mechanisms governing human key recognition. Seminal in this regard was the work of Krumhansl and Kessler (1982).

In parallel, the first systems for chord estimation were being developed (Fujishima, 1999; Nawab et al., 2001; Su and Jeng, 2001). Especially the method of Fujishima (1999) gained a lot of traction, because of the type of features and feature modelling it introduced. The work of Harte and Sandler (2005); Oudre et al. (2011) for instance, are direct descendants of this. Starting with Sheh and Ellis (2003), information about chord changes starts to be taken into account. Their work can be considered as a first attempt at modelling the wider musicological context. Alternative formulations were proposed by Bello and Pickens (2005); Papadopoulos and Peeters (2007); Burgoyne et al. (2007).

3.1.2 Combined key and chord estimation systems

The first systems that exploit the interdependence between keys and chords gradually started to appear from 2004 on. Note that we do not consider a

simple grouping of independent key and a chord estimation algorithms into one system, as in Ryynänen and Klapuri (2008), as a "combined" system. There needs to be some interaction between the two algorithms to be qualified as such. One possible approach to link a key estimation and a chord estimation system together consists of supplying the output of one system to the other. Because either direction of the key-chord relation results in valid information, both key-after-chord and chord-after-key approaches are possible.

For key-after-chord systems, the idea is that a key can easily be derived from a sequence of chords. The chords estimated in the first stage can therefore be used as features for the key estimation. This avoids the need for lower-level spectral features. The key estimation procedure itself is either rule-based (Shenoy and Wang, 2005; Maddage, 2006) or based on an HMM with individual chords (Papadopoulos and Peeters, 2012) or chord pairs (Noland and Sandler, 2009) as observations. The resulting output can be both global (Shenoy and Wang, 2005) or local keys (Maddage, 2006; Noland and Sandler, 2009; Papadopoulos and Peeters, 2012). In most cases, only the best chord sequence produced by the first stage is used in the second stage, meaning that any errors made in the first stage propagate to the second one. We call this a *deterministic* dependency between the chord and key estimation process. To overcome this disadvantage, Papadopoulos and Peeters (2012) do not just use the best chord sequence as input for the key estimation, but the probabilities of each chord label at each frame. These probabilities are the intermediate results of the chord estimation in the first stage. They establish that this *probabilistic* use of the chord results outperforms the deterministic approach of using only the optimal sequence. Another disadvantage of key-after-chord systems is that the chord information can be used for the key estimation, but not the other way around. Shenoy and Wang (2005); Maddage (2006) resolve this drawback by making the process iterative. They refine the first-stage chords by exploiting the newly found key information in a third stage. This refinement is done through heuristics based on music theory.

Chord-after-key systems make use of the fact that a given key gives a strong indication towards which chords can be expected. The key information can then be used to restrict the number of chords that are still eligible [Zenz and Rauber, 2007, the second chord iterations of Shenoy and Wang, 2005; Maddage, 2006], to create context models that depend on the key (Sailer and Rosenbauer, 2006; Khadkevich and Omologo, 2009b) or to consider only those parts of the spectrum that correspond to diatonic tones in the given key (Wang and Wechsler, 2013). The last system is an example of local key estimation, the other systems limit themselves to finding the global key. Similar to key-after-chord systems, chord-after-key systems cannot benefit from knowledge about the chords during the key estimation phase. All key errors therefore propagate into the chord estimation step, especially because only deterministic information is considered in

the given examples. Wang (2013) proposes an iterative system to overcome this problem, where he first determines all the local keys present in the song without pinpointing their exact location. This information is then used to restrict the parts of the spectrum that are considered for chord estimation. Finally, the estimated chords and their boundaries are employed to locate the exact boundaries of the local keys in a third stage.

Estimating keys and chords simultaneously resolves most of the problems inherent to a sequential approach. By delaying all hard decisions until both the optimal key and chord sequences are jointly determined, all key-chord label pairs are considered at all times. The dependency between keys and chord is inherently probabilistic. Therefore there is no chance of a preemptive rejection of a chord or a key that would become optimal after considering additional information. The greatest disadvantage of such an integrated approach is of course its increased complexity, both conceptually and computationally. One possibility to restrict the complexity is to consider only the global key of a piece, like in the system of Yoshioka et al. (2004), which is rule-based. More recent systems commonly use a probabilistic framework, mostly HMMs. We can distinguish both global key (Lee and Slaney, 2008) and local key (Burgoyne and Saul, 2005; Catteau et al., 2007) approaches. The differences between these systems lie in the way keys, chords and their interaction are modelled. This will be discussed in more detail in section 3.3.

3.1.3 Joint estimation with other musicological concepts

Apart from keys and chords, other musical concepts can be labelled in the estimation procedure as well. Similar to using the interdependence between keys and chords, one can try to exploit the dependency between such a third concept and keys or chords. The extra information can then help to restrict key or chord labels to more specific positions and combinations.

One example of such a concept is the metric position, i.e. the position of a beat within a measure. Here the premise is that a chord change is more likely to happen at some positions in the measure than at others. An intuitive example would be that it is more likely to change chords on the first beat of a measure than on the second. This dependency has been taken into account to form a combined metric position/absolute chord estimation system (Papadopoulos and Peeters, 2011), as well as a larger simultaneous estimation of keys, chords and metric position (Mauch and Dixon, 2010b). Also in the iterative sequential key-and-chord system of Maddage (2006), a metric structure estimator early in the chain can revise chord changes based on their metric position.

Another musical concept that is intertwined with chords is the bass line.

The bass note that is played together with a chord often gives an indication of the chord itself. Because the bass note is the lowest note by definition, its fundamental frequency cannot be obscured by interfering harmonics of any other note in the spectrum. It can therefore be estimated comparatively easy. The combination of these two qualities makes the bass line a valuable contribution to a chord context model. The bass line information has both been included deterministically as a preceding step (Yoshioka et al., 2004), or probabilistically integrated into a hypothesis-search-based simultaneous bass-line, global key and chord estimation system (Sumi et al., 2008) or into a completely probabilistic bass-line and chord system (Itoyama et al., 2012).

3.1.4 Related work in the symbolic domain

It is not uncommon that researchers start with a symbolic key and/or chord estimation system and later extend it such that it can process audio as well. For instance, Chew (2002) presented a system for symbolic local key-only estimation based on a new geometric model of tonality, called the spiral array, which later gained the ability to handle audio signals (Chuan and Chew, 2005). The sequential key-after-chord system of Noland and Sandler (2009) was also first conceived to work with symbolic data (Noland and Sandler, 2006). The simultaneous local key and chord audio estimation of Rocher et al. (2010) was preceded by work of the same authors on symbolic key estimation (Robine et al., 2008) and chord estimation (Rocher et al., 2009). The final examples of symbolic/audio system pairs are the key-only estimation systems of Hu and Saul (2009b) and Hu and Saul (2009a), or the chord-after-local-key systems in Wang and Wechsler (2012) and Wang and Wechsler (2013). In most of these cases, the adaptation from symbolic system to audio system was achieved by replacing the feature extraction stage such that the audio equivalent of the symbolic features is calculated. For a sequential key-after-chord system such as the one by Noland and Sandler (2009), this means that the symbolic chords used as input for the key estimation are replaced by chord labels estimated from audio. Other systems usually substitute the symbolic notes that are used as input for their spectral estimates. The remainder of the system can then stay the same.

These examples already show that there is a close link between audio and symbolic estimation systems, but there are other examples of work on symbolic music that has not been reused by the original authors themselves, but served as inspiration to others. We have already mentioned the links between the early audio key-only systems and their symbolic counterparts, but also the simultaneous key-and-chord estimation of Raphael and Stoddard (2003) preceded all efforts on its audio equivalent. Finally, some notable works on chord-only estimation in the symbolic domain are

the preference-rule-based system of Temperley and Sleator (1999) and the graph-search approach of Pardo and Birmingham (2002).

3.2 A unifying view on key and chord estimation

In general, key estimation and chord estimation are conceptually similar. Both are concerned with labelling the harmonic content of audio. There is of course the difference in time-scale, but the same techniques can easily be reused. Therefore we will discuss them together in the remaining part of this chapter. Before we can explore the differences and similarities between various systems, we first describe a categorisation of the different components that form a general key and/or chord estimation system. The input signal is always an audio file, but it is transformed to a feature vector sequence by means of a *feature extraction* stage. The feature stream represents the audio information in a shape that is more adapted to the next stage. To this end, some facets of the information are accentuated and some are ignored, but no supplementary information is created. The calculated features form a new time-changing signal vector with a certain feature rate, which is independent of the sample rate of the audio input. This feature rate determines the maximal time precision or granularity of the final output labels. In the subsequent *feature decoding* stage, the output labels are estimated based on these features. This involves both segmenting the feature stream into sections in which the label stays constant and naming those sections. The way this is done can be characterised according to three aspects, which we will call the *acoustic*, *durational* and *contextual* characteristics of the feature decoding. The acoustic characteristic covers the method of testing how well a candidate label is supported by the feature stream. Both the durational and the contextual aspect deal with techniques that favour certain "expected" labels over others. These techniques can incorporate musicological knowledge similar to the human expectancies of music. The durational aspect encompasses techniques that favour certain label durations over others and the contextual aspect involves looking at the broader musical context to promote certain label combinations over others.

The different aspects of a key and/or chord estimation system are not isolated from each other. There are strong dependencies between them. Least dependent on the other aspects is the contextual aspect of the feature decoding. It is possible to ignore all context information and to decide on a label by only considering locally available information. The contextual aspect can therefore be absent in a system. On the other hand, the durational aspect is strongly connected to the feature extraction stage. They

work together to decide on the segmentation of the feature stream and consequently need to be matched to each other. Even if no component of the system takes explicit care of the durational aspect, an intrinsic preference for certain label durations is still present and at least partly determined by the feature extraction rate. If the feature rate is close to the label change rate, every feature vector can be labelled independently, whereas in the case of a feature rate that is much higher than the label change rate, the system will have to take into account that subsequent labels are correlated. The feature rate also puts a lower bound on the label duration and it determines the precision of the label localisation. Feature extraction and the durational aspect of the decoding are therefore complementary. Lastly, the requirement of a match between the feature extraction stage and the acoustic aspect of the feature decoding is quite obvious. In order to test for the accordance between the features and a label, each label requires an internal representation of what the features would ideally look like. This representation is specific to a particular feature extraction method, such that the label representations and the feature set always needs to be considered together. As a result, it is very hard to evaluate the feature extraction step in isolation.

Global key estimation can take an extra constraint on the key label into account in either the feature extraction stage or the feature decoding phase. The former approach calculates a single feature vector to represent the whole duration of a music piece, instead of a stream of feature vectors. This single feature vector can then be calculated with high precision because lots of data are available (Pauws, 2004; Gómez and Herrera, 2004). The second strategy is to calculate a sequence of local features as before, but to further restrict the durational aspect of the decoding such that only a single key covering the whole feature sequence is generated (Zhu et al., 2005; İzmirli, 2005b). By definition, no contextual aspect is present in both types of system. The first approach also needs no durational aspect.

3.2.1 Common frameworks for feature decoding

The different aspects of the feature decoding stage do not always have a one-on-one correspondence with the components of the technical framework that is used for decoding. We will now list some of the more common frameworks that are used to implement the feature decoding and point out how their components relate to the individual aspects.

A first technique consists of making a frame-based estimation of the probability of each symbolic label at each time step, accounting for the acoustic aspect. The local probabilities are then optionally smoothed over time (Fujishima, 1999; Harte and Sandler, 2005; Oudre et al., 2011). The acoustic probabilities can be calculated by a variety of models, discussed

later. The smoothing provides the durational aspect by promoting labels that stay constant for the duration of the smoothing window. If it is not present, the durational aspect is implicitly provided by the feature rate. It is less common to take the context into account with this set-up, although the system by Cheng et al. (2008) shows that it is possible to perform a greedy search where the acoustic probabilities are augmented with probabilities for the next chord label, based on the the previously found chords.

The most commonly used framework for key and chord estimation is a hidden Markov model (HMM) (Sheh and Ellis, 2003; Bello and Pickens, 2005; Noland and Sandler, 2009; Papadopoulos and Peeters, 2012). The acoustic aspect is again taken care of by a frame-wise calculation of observation probabilities. These are then probabilistically combined with transition probabilities to decode the globally optimal label sequence. The elements on the diagonal of the transition matrix provide the durational aspect and the off-diagonal elements the contextual aspect. When the off-diagonal elements all have the same value, such as in Chai and Vercoe (2005), no contextual aspect is present.

In a conditional random field (CRF), the different aspects are harder to isolate. The whole sequence of observations is modelled together here, meaning that the observations are explicitly assumed to depend on each other. All aspects are therefore strongly intertwined. An example of a linear-chain CRF for chord estimation can be found in Burgoyne et al. (2007). Another example of a system where the different decoding aspects cannot be isolated, is the convolutional neural network of Humphrey and Bello (2012).

In contrast to other high-level discussions of estimation systems, such as in Cho and Bello (2014), we do not distinguish an extra filtering step between feature extraction and feature decoding. We argue that this filtering (with optional resampling) is an integral part of the feature calculation. After all, only the result of the feature extraction stage, possibly including filtering, is conceptually important for the decoding of the features. Whether the features are calculated directly over a certain window size with the desired feature rate or whether they are calculated by smoothing over multiple finer windows and/or downsampling from a higher rate, is an implementation issue. It could influence the quality of the resulting features, but is not conceptually relevant for the rest of the system. Similarly, Cho and Bello (2014) also split the feature decoding stage into two parts: a so-called pattern-matching step followed by a filtering step. We feel that this distinction is too specific, as it does not accommodate systems that integrate these two steps.

3.2.2 General tendencies

Comparing the performance of different estimation systems is not straightforward. One reason is the difference in chord vocabularies generated by the various systems. Other factors are the distinct evaluation measures, the variation in test sets between papers and the extent to which the systems are optimised for a certain set. Of paramount importance for levelling out some of these differences is the yearly Music Information Retrieval EXchange (MIREX)[1]. It is an informal competition to which researchers can submit a system which then gets tested on the same data set and evaluated with the same measures. Since 2008 there has been a chord estimation task. Top performing algorithms estimate chords correctly around 70% to 80% of the time, depending on the data set, with the reservation that the chord evaluation vocabulary is limited and that the data sets are publicly available. There is no local key estimation task, but a global key estimation task has been organised in 2005 and again from 2010 on. A top performing system recognises the key correctly for 87% of the files, where the test set is secret, but synthesised from MIDI. The number of submissions is usually considerably less than for the chord estimation task (average number of participating teams is 3.4 versus 7, some teams have multiple submissions).

When analysing the chord estimation submissions through the years, a couple of tendencies can be discovered. First of all, the number of participating teams is steadily declining, from 11 to 3, although the task is far from solved[2]. Second, submissions rely increasingly on machine learning techniques. This is undoubtedly the result of an increase in available training data. Early systems had to be based on musicological knowledge simply because of the scarcity of labelled data. Furthermore, the complexity of chord estimation systems is rising. This is partly because they are more and more integrated into larger systems that estimate chords jointly with keys, metric position and/or bass line and use more of the context (see section 3.1.3).

Due to the fewer submissions for the key estimation task, it is harder to find global trends. Here too the number of participants declines, from a maximum of 6 to 1. The submitted systems generally estimate only keys and are rather basic. One reason is that the focus on global keys does not encourage more complex systems that can handle key changes.

Evaluating the performance of a system is one thing, explaining why a certain score is achieved is another. If we look at the top performers for chord (Cho, 2013; Khadkevich and Omologo, 2013b) or key estimation (Purwins et al., 2000; İzmirli, 2005a), they have in common that they

[1] `http://www.music-ir.org/mirex/wiki/MIREX_HOME`

[2] The number of participants at MIREX is far from the definite measure for the popularity of chord estimation research. There are multiple research groups that have simply never submitted their system. Nonetheless, these numbers give an indication of the general trend.

estimate keys or chords separately. It seems that the supplementary information present in the interaction between keys and chords has not been effectively exploited by any of the joint systems. We are particularly interested in examining to what extent both tasks can reinforce each other. Therefore we will propose our own joint system for key and chord estimation in chapter 5, where we carefully study the importance of the relations between keys and chords and other decoding aspects. We will formulate this system in terms of an HMM, because it allows us to easily separate the different decoding aspects.

3.3 A discussion of system components

After having discussed entire key and chord estimation systems, we now look at their constituting components in more detail. Unless absolutely necessary, we make abstraction of the fact whether a component has been used for key or chord estimation. After all, an improvement in the feature extraction proposed in a key estimation system might also be beneficial to a chord estimation system. This way, we can more easily compare the innovative parts across different systems.

3.3.1 Feature extraction

An overwhelming majority of key and chord estimation systems uses chromagrams as the acoustic features. This is a time-frequency representation much like the short-time Fourier transform (STFT). A time signal is split into segments, which are then converted to frequency spectra. The spectrum is calculated on a logarithmic frequency scale and collapsed into a single octave by grouping together the logarithmic bins that are spaced an integer number of octaves apart. This type of transform was independently suggested by Fujishima (1999) and Wakefield (1999). The appeal of this transformation is immediately obvious when we consider the music theoretical knowledge presented earlier: notes are logarithmically spaced on the frequency axis and only their chromas are required to determine to which key or chord they contribute. The definition of the chroma transformation therefore comes very naturally as a mathematical formalisation of musicological knowledge. The result is therefore a 12-dimensional vector, although sometimes multiples of 12 are also used to maintain a greater spectral resolution (Sheh and Ellis, 2003; Chai and Vercoe, 2005). Collapsing the bins into one octave makes the features more robust against all possible variations of the chord voicing, which we do not want to discern. The process also reduces the sensitivity to localised errors in the frequency analysis, because the bins are smoothed over all octaves.

There are however also examples of systems that use other features. In some of the pioneering work, alternative features such as complete spectrograms (Nawab et al., 2001), wavelets (Su and Jeng, 2001) or mel-frequency cepstral coefficients (MFCCs) (Sheh and Ellis, 2003) were tried. İzmirli (2005a) compared full spectra and chromas for key estimation and found that chroma representations outperformed spectra in his system. In spite of this, the usage of complete spectra has been picked up again recently with the advent of deep learning techniques, which preferably deal directly with spectrograms. The underlying idea behind these machine learning algorithms is to keep the features relatively simple such that the most optimal internal representation of a label can be automatically discovered, without steering the system towards to human-influenced chroma features (Humphrey and Bello, 2012; Boulanger-Lewandowski et al., 2013). A directly contrasting approach is to restrict the features to the so-called "tonal centroids", which have only six dimensions (Harte et al., 2006; Lee and Slaney, 2008). Like in the chroma transform, octave information is ignored, but the vector elements now represent coordinate pairs in each of the three circularities of the Tonnetz space (Cohn, 1998). Because this is a geometrical model of the pitch space, small deviations in the coordinates will only cause small changes in perception. Therefore this representation is compact as well as robust to external noise. Another way to reduce the dimensionality of the spectral representation is to keep only a smaller orthogonal set of basis functions that is determined through the use of principal component analysis (Morman and Rabiner, 2006; İzmirli, 2006; Nichols et al., 2009; Boulanger-Lewandowski et al., 2013). In contrast to tonal centroids, this reduction is data-dependent. A final alternative for the use of chroma features has already been mentioned when discussing related symbolic estimation systems. Key or chord estimation can also be performed by first transcribing all notes present in the signal and then feeding these note estimates into a symbolic system. Naturally, this approach suffers from a propagation of errors in the polyphonic transcription step. Because note estimation is very much an unresolved problem in itself, this approach (Katayose et al., 1988; Martin, 1996; Kashino and Hagita, 1996; Kashino et al., 1998) has been mostly abandoned once the first systems working directly on audio came into being. The more recent system of Mearns et al. (2011) is the proverbial exception that proves the rule.

3.3.1.1 Logarithmic spectrum calculation

Irrespective of whether the final features are spectrograms, chromagrams or tonal centroids, the first stage of a feature extractor is a time-frequency analysis. The most widespread time-frequency transformation, the STFT, provides spectra with bins that are equidistantly spaced on a linear frequency scale. Because music notes are distributed on a logarithmic frequency scale, the linear bins of the spectra then need to be reassigned. This

redistribution can be seen as a filter bank with logarithmic spacing and both the shape and the width of the filters varies considerably across implementations. The filters can be rectangular to let each linear bin within a local area contribute equally, or peaked around their centre, such that linear bins closer to a theoretical note position contribute more. A number of shapes are compared in the study by Cabral et al. (2005). These filter banks can sum to unity (required for perfect reconstruction), such that each frequency contributes equally to the logarithmic spectrum, or not, such that some sort of equalisation is performed around theoretical note positions.

The resampling and interpolation associated with the Fourier transform followed by a filter bank creates of course additional noise due to rounding errors and quantisation, which is undesired. Therefore an alternative that directly calculates a logarithmic spectrum would be very useful. The concept of such a transformation, the constant-Q transform (where the quality Q stands for the ratio of the centre frequency to the bandwidth of a filter), has already been introduced to the music domain in 1991 by Brown. Here, the time signal is directly analysed by a filter bank with frequency-dependent analysis windows instead of fixed length windows such as for the STFT. Unfortunately, a direct implementation was not efficient, which lead to the widespread use of an efficient algorithm that is based on the STFT (Brown and Puckette, 1992). Therefore the constant-Q transformation is in those cases, although conceptually different, in practice the same as an STFT followed by a logarithmic filter bank. Nonetheless, it has found widespread adoption in chroma calculation due to the appeal of its underlying idea, but it should be regarded as one of the many variations of the previously outlined variations in filter width and shape.

Recently however, an alternative formulation of the constant-Q in terms of non-stationary Gabor frames has been developed that is both efficient and perfectly invertible (Holighaus et al., 2013). Unfortunately, the typical level of detail reported in literature is generally not enough to know whether the implementation of the constant-Q transform was genuine or just a two-pass approach based on the STFT. Therefore it is hard to assess the influence of this factor in past and future papers. Some overview papers (Varewyck et al., 2008; Stein et al., 2009; Jiang et al., 2011) compare different chroma implementations, and do contrast constant-Q transforms to interpolated STFT transforms, but the difference between constant-Q implementations has never been examined. At the moment, only the system of Humphrey and Bello (2012) is known to use the Gabor-based implementation, but whether this increases the recognition performance has not yet been established. The recent availability of a toolbox that offers an accessible implementation of this improved algorithm (Schörkhuber et al., 2014) may promote the usage of constant-Q implementations that both conceptually and practically avoid the drawbacks of the current way of calculating logarithmically spaced spectra.

If the spectra are not used directly as features themselves, the logar-

ithmic spectrum sometimes has a resolution that is a multiple of 12 logarithmic bins per octave (Harte and Sandler, 2005; Gómez, 2006a; Zhu and Kankanhalli, 2006). The extra spectral precision can then be used to increase the accuracy of the optional steps described in the next paragraphs.

3.3.1.2 A multitude of chroma calculations

Although the idea of using a chromagram has achieved broad consensus, there exists a multitude of ways to calculate it. We can informally conjecture that a chroma vector represents the salience of each pitch class in the signal, but what does "salience" stand for? And what does "note" mean if we can only analyse frequencies? Even at the most basic level, no answers to these questions have been agreed upon. To illustrate this in its simplest manifestation, the spectral properties that are summed per octave can be magnitudes (Harte and Sandler, 2005), squared magnitudes (Fujishima, 1999), log-compressed magnitudes (Morman and Rabiner, 2006) or perceptually weighted magnitudes (Pauws, 2004; Schuller et al., 2009a). Throughout the years, many signal processing techniques have been used to solve specific issues with the simplest chroma calculation schemes. These will be discussed in the following paragraphs.

DETERMINING THE REFERENCE FREQUENCY A first problem is that a single reference frequency for the notes on the frequency axis is assumed. As said before, this is often considered to be 440 Hz, but in practice this can vary slightly. This can simply be the result of instruments that are tuned differently, but it can also arise during the post-production process, e.g. due to a different tape playback speed during mastering. The result is in either case that harmonic peaks are not present where they are expected to be, leading to a spectrum that is less clear. In order to compensate for a constant deviation of the reference frequency from 440 Hz, algorithms have been developed that estimate the actual reference frequency as a first step of the feature calculation, such that the spectrum can be centred around the true harmonic peaks. These methods mostly use the magnitudes of a high-precision spectrogram to find the deviation from the standard reference frequency. This deviation can be found through interpolating a pitch peak histogram (Harte and Sandler, 2005; Bello and Pickens, 2005; Zhu and Kankanhalli, 2006), from the biggest single peak (Ellis and Poliner, 2007), by comparing the energy at all theoretical note positions for an enumeration of deviation candidates (Peeters, 2006a), or through circular statistics (Dressler and Streich, 2007). In contrast to these methods, Khadkevich and Omologo (2009a) use both magnitudes and phases in a similar way as the frequency estimation in a phase vocoder (Flanagan and Golden, 1966). A wider discussion of the different approaches and their characteristics can be found in the overview paper by Degani et al. (2014). Besides a different reference frequency, a tuning can also differ in other ways from the impli-

citly assumed 12-tone equal temperament tuning. A study of these aspects can be found in Lerch (2006).

HANDLING HIGHER HARMONICS A second complication of simply folding logarithmic bins into one octave, is that the higher harmonics of a note will add evidence to chromas that are not necessarily present in the signal. For instance, the third harmonic of a note will lead to the presence of a chroma a perfect fifth above (or a perfect fourth below) the chroma of the fundamental. This does not correspond with our music theoretical notion that each note should only lead to one chroma. Therefore a variety of approaches have been developed to mitigate the contribution of higher harmonics to the chromagram. A first method is to make use of a Harmonic Product Spectrum, which emphasises fundamental frequencies by multiplying the spectrum with increasingly more compressed versions of itself (Morman and Rabiner, 2006; Lee, 2006a). The same goal is achieved with the related Harmonic Peak Subtraction of Peeters (2006b), where the spectrum is multiplied by a score that is high for fundamental frequencies supported by higher harmonics. Another option is to apply pitch-tracking techniques to link higher harmonics to their fundamental frequencies. Multiple techniques have been proposed for this. Some keep only those peaks in the spectrum for which a sufficient amount of supporting higher harmonics can be found, leading to a sparse chroma representation (Sailer and Rosenbauer, 2006; Varewyck et al., 2008). Others perform a partial transcription by replacing frequency values with pitch saliences indicating the probability of being a fundamental frequency(Ryynänen and Klapuri, 2008) or with the activation patterns that best reconstruct the spectrum as a combination of idealised note bases, found as the result of a non-negative least squares decomposition (Mauch and Dixon, 2010a). A last way to deal with the contribution of higher harmonics to the logarithmic spectrum, is to use machine learning techniques to learn the optimal mapping from spectrum to chroma vector. İzmirli and Dannenberg (2010) use a multi-layer perceptron, whereas Glazyrin (2013) proposes stacked denoising auto-encoders to this effect. As mentioned before, Boulanger-Lewandowski et al. (2013) directly feed spectrograms into a deep belief network, but they constrain their network to generate a chromagram representation as an intermediate target, so the first layer of their network can also be seen as a machine learned chroma transform. Similarly, Humphrey et al. (2012) use a deep belief network for finding the optimal mapping from spectrum to tonal centroid features instead of chromas. Whether the removal of harmonics is beneficial for the entire chord or key estimation system depends on the feature decoding, but removing them has at minimum the advantages that the resulting chromagram is more suitable for visualisation.

REDUCING TIMBRE DEPENDENCY The third issue with chroma calculation is its robustness to timbre changes. The idea here is to calculate the chromas in such a way that the impact of the spectral envelope that characterises different instruments or voices is reduced. Because this difference in timbre is manifested as a difference in the ratio of harmonics, making chromas robust against timbre changes is somewhat related to the previous problem of handling the contributions of higher order harmonics. Pitch-tracking techniques that retain only a sparse number of fundamental frequencies, such as those of Sailer and Rosenbauer (2006); Varewyck et al. (2008), also implicitly remove timbre information. When the higher harmonics are not entirely removed, the strategy is often the opposite: rather than reducing the higher harmonics in the signal, their amplitudes are put on a fixed level such that the timbral variance between instruments decreases. This approach is called spectral whitening. Two traditional signal processing techniques achieve spectral whitening: high-pass cepstral liftering (Morman and Rabiner, 2006; Müller and Ewert, 2010) and background spectrum subtraction (Catteau et al., 2007; Mauch and Dixon, 2010a). A more simple method consists of binarising the spectrum on the basis of a threshold (Zhu and Kankanhalli, 2006; Cranitch et al., 2007). In the feature extractors of Morman and Rabiner (2006); Mauch and Dixon (2010a), the spectral whitening is subsequently followed by one of the methods that aim to reduce the higher harmonics discussed above.

MINIMISING ATONAL SIGNAL COMPONENTS The last complication that arises when computing chromas is that not every sound in the signal is tonal. Notably untuned percussion produces sound that is wide-banded and noise-like (in the signal processing sense). It can therefore mask other, harmonically significant sounds. A trivial first step is to consider only the frequency range that tonal instruments can produce. Frequencies outside this range are guaranteed to be noise. Furthermore, some algorithms have been proposed to reduce the impact of percussive sounds in the range where tonal and atonal instruments overlap. In the work of Ono et al. (2008), used by Ueda et al. (2010); Ni et al. (2012), the spectrogram is iteratively decomposed into a harmonic and a percussive part based on the assumption that the spectrogram gradients in the horizontal, respectively the vertical, direction form independent Gaussian distributions. The resulting harmonic component is then transformed back to the time-domain to generate a signal from which a better chromagram can be calculated. As such, it can be easily combined with any method for chroma calculation. Schuller et al. (2009b) suggest a similar decomposition of the signal into a harmonic and a percussive part using non-negative matrix factorisation. However, they remark that using the resulting harmonically enhanced signal does not improve the performance of their key estimation system.

Apart from these methods that really try to separate the harmonic and percussive part into two different signals, there are also techniques to high-

light the harmonic part, without paying attention to the significance of the residual signal. Gómez (2006a) uses a transient detector as a preprocessing step to exclude noisy regions from the spectral analysis. Peeters (2006a) achieves the same goal by running the audio file through a sinusoidal analysis/resynthesis algorithm first. Another approach is proposed by Glazyrin and Klepinin (2012). They use a technique borrowed from image processing to keep only the horizontal edges in a spectrogram by means of a Prewitt filter. Lastly, Wang and Wechsler (2013) perform audio denoising by means of an undecimated wavelet transform.

IMPROVING SPECTRAL PRECISION Besides trying to avoid the noise coming from non-harmonic contributions to the signal, other attempts have been made to make the spectral analysis more precise. Zhu and Kankanhalli (2006) search for local maxima in a high-resolution spectrogram to handle vibrato and they keep only those partials that form a diatonic set, starting from the strongest peak. This technique is called consonance filtering. Ellis and Poliner (2007) make use of the phase derivative to improve the frequency resolution to a sub-bin precision. This method is known as instantaneous frequency estimation and it is especially relevant for low frequencies where notes are spaced much more closely together on a linear scale. Similarly, Khadkevich and Omologo (2013b) use a spectral reassignment technique based on the complex STFT values to redistribute the energy centred on the bins over the intervals between bins and to filter out noisy components. A simpler way to achieve sub-bin resolution is quadratic interpolation of the bins, as proposed by Gómez (2006a).

LEVERAGING REPETITION Apart from the methods that try to compensate for one of the pitfalls inherent to chroma or spectral calculation, other enhancements have been proposed as well. For instance, all techniques that try to make use of repetition in the signal fall under this category. It often happens in music that some parts are repeated. If we could detect these repetitions, the number of observations on which we can base our estimation would increase and therefore be more robustness against random noise. A possible drawback is that minor variations in the repetitions get lost, because they are also considered to be noise. Mauch and Dixon (2009) look for fixed-length sections around the diagonal of a self-similarity matrix constructed by calculating Pearson correlation coefficients between all beat-synchronised segments. The corresponding chromagram sections are then averaged and the resulting labels are then used for all instances. Cho and Bello (2011) use recurrence plots to this end, which are thresholded matrices of similarity between all fixed length sequences of frames instead of between all isolated frames. According to them this has the advantage that smaller units of repetitions can be identified. Therefore the averaging can also take place over repetitions of single chords instead of over repeated chord sequences, which increases its applicability. Finally Glazyrin

and Klepinin (2012) also compute a beat-synchronised self-similarity matrix, just like Mauch and Dixon (2009), but they weigh the different repetitions according to their similarity before averaging them into the final features. In this way, they adhere to the work of Cho and Bello (2011).

3.3.1.3 Feature usage

A lot of key or chord estimation systems, especially the earlier ones, calculate just a single feature sequence which is subsequently decoded. An alternative which became widespread after its introduction by Mauch and Dixon (2008), is to extract a secondary chromagram from the lower frequency region of the spectra, often called a *bass chromagram*. This supplementary 12-dimensional chromagram is then either simply concatenated to the primary chromagram to form 24-dimensional observations (Mauch and Dixon, 2008; Ni et al., 2012) or used as an independent stream (Khadkevich and Omologo, 2011; Mauch and Dixon, 2010b). The original motivation for extracting a bass spectrogram was that the lowest note in the spectrum acts as a strong indicator towards the chord and that it permits to find the chord inversion. It has been shown, however, that employing a bass spectrogram is ineffective in estimating chord inversions (Pauwels and Peeters, 2013). Considering that the bass cutoff frequency is arbitrarily predetermined, and no mechanism is put into place to ensure that the bass chromagram contains exactly one tone, this is entirely logical. If multiple notes fall into this bass range, their order is therefore lost through the intrinsic character of the chroma transform and if no tone is present in this region, the inversion estimation is based on random noise. Nevertheless, a bass chromagram adds extra observations such that the subsequent feature decoding is less susceptible to mistakes in the primary chromagram (due to masking effects or noise for instance). It is explicitly demonstrated by Khadkevich and Omologo (2011) that such an increase in the number of observations can lead to better chord estimation results. This idea can be extended further by increasing the number of chroma streams extracted from the same spectrogram up to four (Cho, 2013). The original premise that bass notes can aid in estimating chords, and specifically chord inversions, is valid though, but it requires separate handling of the lowest notes, as done by the systems described in section 3.1.3. In general, all systems that co-estimate keys and chords with another musical concept, require additional features specific to this third concept, for example the output of a bass line estimator.

The approach of Rocher et al. (2010) also works with multiple observation streams, but here the multiple streams do not cover different frequency ranges, but multiple time windows. In their simultaneous key and chord estimation system, the observations used for the chord estimation are chromas averaged over three different windows, whereas the key estimation only uses the largest time window. A parallel can be drawn to the

deep-learning system of Boulanger-Lewandowski et al. (2013), where the inputs of the layers are complemented with their aggregated means and variances over different windows. As much as eight different windows are fed into the system, where not only the size of the window can vary, but also the offset, such that windows no longer need to be all centred around the time instant that is being analysed. However, because of the nature of their machine learning system, it is not guaranteed that all windows play a significant role in determining the output.

3.3.1.4 Feature timing

Features are calculated locally such that label changes can be precisely located. Their timing is determined by the size of the windows over which the features are determined and the time shift between subsequent windows. We have already touched on the connection between the feature timing and the durational aspect of the feature decoding. So, before we start exploring the latter, we will quickly review the options for segmenting the audio stream. A first possibility is to use an arbitrary, fixed feature rate. This rate should be high enough to permit a precise localisation of label changes, but the windows should be long enough to contain a sufficient number of periodic signal repetitions. The solution is to employ overlapping windows with a rate as high as the required temporal precision. An increase in rate comes at an increased computational cost though.

A significant reduction in computational time can be achieved by synchronising the features to the metrical grid. For example, a popular practice is to make the features beat-synchronous, as pioneered by Shenoy et al. (2004). The underlying assumption is that labels only change precisely on the position of a beat and that they consequently are constant within every interval between two consecutive beats. If this is the case, the feature rate can be increased without the loss in temporal precision it normally causes. Usually, the time between beats is also considered long enough for the signal to be stable, such that the inter-beat interval is taken as the window size (creating disjoint segments), but this is not necessarily so (Glazyrin and Klepinin, 2012). A more conservative speedup with less strict assumptions can be achieved by tracking a subdivision of the beat, but for key estimation, tracking a higher metric level, such as measures (Hu and Saul, 2009b), is also arguable. As mentioned before, one possible way to apply the global key constraint is to make the feature analysis window as long as the entire music piece, creating a single feature vector instead of a feature stream.

In practice, if a feature stream is following a metric level, its calculation is rarely done straightaway over the corresponding metrical windows. Because the size of these windows is variable due to changes in metre, the features are usually calculated over fixed size windows with a high rate and then filtered over the desired time period, with optional resampling. The result is that it is not always trivial to determine exactly which au-

dio samples have contributed to a particular feature segment and to what extent.

3.3.2 Feature decoding

Each of the three aspects of the feature decoding can be determined through musical knowledge or through machine learning. Note that the technical framework used for the models is not necessarily discriminating for this division. For instance, the observation probabilities in the HMM of Bello and Pickens (2005) are set manually based on musical theory, whereas they are trained using the Baum-Welch algorithm by Sheh and Ellis (2003). Neither do all aspects of one system need to be specified in the same way. While the observation probabilities of Bello and Pickens (2005) are set on the basis of knowledge, their transition matrix is trained without supervision. The division between knowledge-based and data-driven techniques is not always strict though. For instance, many machine learning techniques need an initialisation step, in which musicological knowledge can be used. On the other hand, the used musicological knowledge can emerge from an analysis of a large corpus. We will consider a method as data-driven if it contains an (iterative) refinement of its parameters on a data set somewhere in the process. In knowledge-based methods, the parameters are set through human reasoning about readily interpretable parameters, eventually complemented by a small set of free control parameters.

In practice, we will mostly discuss the components of systems in which the acoustic aspect is taken care of by a framewise acoustical model. They represent the large majority of systems in the literature. Alternative systems consist of a monolithic, all-integrated decoding step in which the different aspects cannot be discerned. Few things can be said about them, except for the fact that they use machine learning to optimally match their parameters to the used features.

It is common for data-driven acoustic models of keys and chords to transpose the training labels to one common tonic, respectively root. This increases the number of examples per model and ensures that the prior probability of the acoustic models stays equal for each tonic or root. The resulting generalised mode or chord type model is then transposed back to each of the chromas, such that all key or chord models are circularly shifted versions of each other. Although widespread, this step is not universally applied, as the system of Ni et al. (2012) proves. Relating the acoustic models of the same type or mode to each other in this way could be considered an example of musicological knowledge in a data-driven method. In knowledge-based models, this type/mode relation arises naturally.

3.3.2.1 Knowledge-based acoustic models

Musicological knowledge in knowledge-based models is usually represented in the form of idealised chroma templates (also named profiles) for all distinct labels. Consequently, these templates have the same dimensions as the chroma vectors and are derived from music theory or from perceptual experiments. The simplest templates are binary, reflecting the inclusion or exclusion of a certain chroma in a chord (for chord templates), or the chroma's diatonicness in a key (for key profiles). For instance, a *G*maj chord template would have its ones corresponding to the *G*, *B* and *D* chromas and zeros for the remainder. Such binary templates has been used by the early chord estimation system of Fujishima (1999) and by many others since. For key estimation, it is more popular to use templates that stem from music perception experiments. Very influential in this regard were the tone-probing experiments of Krumhansl and Kessler (1982). A number of human subjects rated how well single notes of all chromas fit in a given key, which is established by first playing a characteristic sequence of notes or chords in that key. The results were found to be highly similar for all keys with the same mode (demonstrating transposition invariance). Consequently, a single tone profile relative to the tonic was derived for each mode (major or minor) by averaging the profiles over the tonics. The resulting templates were then used for symbolic key estimation, and later also found their way to audio key estimation, in the work of Purwins et al. (2000). İzmirli (2005a) established that they work better than binary profiles. Based on Krumhansl and Kessler (1982)'s work, Temperley (1999) proposed a variant that has been specifically designed for automatic key estimation. An alternative way to obtain key templates is through statistical analysis of corpora of symbolic music (Chai, 2005; Noland and Sandler, 2009). A good comparison between all aforementioned templates and some other variants is made by Gómez (2006b).

We have already established that an appropriate acoustic model should be matched with the features it processes. The previously described templates all assume theoretical chromas, where one note leads to just a single chroma. Therefore they are a good match for chroma extraction methods that suppress the harmonics of the notes. An orthogonal approach, is to take the existence of the harmonics explicitly into account in the acoustic model. The appropriate features therefore still need to contain them, otherwise there would be a strong mismatch between the features and the feature decoding. Accounting for harmonics can be combined with any of the preceding theoretical chroma profiles. First, a theoretical shape of the amplitudes of the harmonics has to be hypothesised, commonly an exponential decay a^i with the harmonic index i going up to n harmonics. For every chroma, the resulting series is then scaled by the theoretical chroma template value. The theoretical harmonics are finally folded into one octave, similar to the chroma calculation, and contributions to the same chroma

are added. In case of a binary chord or key template that contains a C, this would give a value for the C chroma of $a + a^2 + a^4 + 2^8 + \ldots + a^n$ and for the G chroma of $a^3 + a^6 + a^{12} + \ldots + a^n$. The other notes in the key or chord with overlapping harmonics can add even more to these values. This way of template calculation has been first applied by Gómez (2006a) for key estimation and by Papadopoulos and Peeters (2007) for chord estimation. Instead of hypothesising a harmonic distribution, İzmirli (2005a) used real harmonic ratios that were measured on a collection of monophonic piano sounds.

The resulting templates are used as parameters in the acoustic models. The most prevalent model consists of calculating the distance between a chroma vector and all templates. The distance is then inversely proportional to the probability of the corresponding label. Oudre et al. (2011) studied a variety of distance measures that can be used in this context. Alternatively, the templates can be used to set the means of multivariate Gaussians, as done by Bello and Pickens (2005); Mauch and Dixon (2010b), or half-Gaussians (Catteau et al., 2007). The advantage of knowledge-driven acoustic models is that no labelled training data is needed. This also reduces the risk that the model is too tailored to one specific data set.

3.3.2.2 Data-driven acoustic models

A number of different machine learning techniques and associated models have been used to create optimal representations for specific feature calculation methods. A first technique is the learning of parameterised distributions such as multivariate Gaussians (Sheh and Ellis, 2003), Gaussian mixture models (GMMs) (Morman and Rabiner, 2006; Burgoyne et al., 2007) or Dirichlet distributions (Burgoyne and Saul, 2005) through expectation-maximisation (Dempster et al., 1977). Another option is to train a topic model without supervision such as latent Dirichlet allocation (LDA) (Hu and Saul, 2009a) or an infinite Gaussian mixture model (Wang, 2013). This contrasts to the application of discriminative models, such as support vector machine (SVM) classifiers (Morman and Rabiner, 2006; Weller et al., 2009; Schuller et al., 2009a).

Another category of methods is based on the use of artificial neural networks. Both unsupervised learning, with self-organising maps for standalone key (Purwins et al., 2000) and chord (Su and Jeng, 2001) estimation, and supervised learning systems have been applied. An example of the latter is the chord classifier of Zoia et al. (2004), where a three-layer perceptron (MLP) is trained for each chord label. As for supervised key estimation, Sun et al. (2009) use a time-delay neural network that can detect key modulations. Finally, the most recent approaches are based on deep-learning. They incorporate different types of neural networks, such as stacked denoising auto-encoders (Glazyrin, 2013) or deep restricted Boltzmann machines (Boulanger-Lewandowski et al., 2013). For a comparison of a num-

ber of machine learning techniques to each other and to a knowledge-based acoustic model based on their applicability to key estimation, see Gómez and Herrera (2004).

In order to link knowledge-driven and data-driven acoustic models, it can be enlightening to interpret the trained models in a musicological way. For instance, the means of a multivariate distribution trained over a set of chromas can easily be seen as a kind of template containing musicological knowledge. It could be used with a distance metric to act more or less like a profile as used for knowledge-based models. Moreover, a musicological interpretation of the generated models is even required for strictly unsupervised training systems, where the resulting models need to be manually assigned to musical labels before they can be used in a decoder.

3.3.2.3 Durational aspect

Algorithms that model sequences of observations directly, such as CRFs or convolutional neural networks, take care of both the acoustic and durational aspect. In the majority of systems however, the acoustic models process single frame observations, assuming independence between subsequent frames. The result is that there is nothing to prevent excessive fragmentation in the output, leading to spurious short labels or oscillations between labels. A realistic sequence consists of the opposite, clusters of labels with clear cuts. Duration models, which come in different degrees of complexity, can help to prevent this fragmentation. The necessity of a duration model for framewise acoustic models can be somewhat reduced by using delta features to compensate for the independence assumption. The features are then supplemented with their time derivatives, as done by Ueda et al. (2010). This only accounts for local correlation however, and is no substitution for proper duration modelling.

As we have already pointed out, the simplest way of taking the expected duration of the labels into account is to do this deterministically by filtering the estimated label sequences (Harte and Sandler, 2005) or filtering the output of the acoustic model (Oudre et al., 2011). Both mean and median filters over a fixed amount of frames have been used. The exact amount can be made dependent on the tempo by using the output of a beat-tracker. An extreme case is when the global key is determined, the filter window then encapsulates the whole music piece and takes the mean of all local estimations (Zhu et al., 2005; Cranitch et al., 2007).

When an ergodic HMM is used as the system's framework, the smoothing happens probabilistically. The self-transition probabilities on the diagonal of the transition matrix regulate how easy or how difficult it is to change between labels. They are either manually set using musicological knowledge (Burgoyne and Saul, 2005) or trained (Sheh and Ellis, 2003). In contrast to a filtering of output labels, the Viterbi (1967) algorithm used in an HMM decoder does not just consider the best output of each frame for

the temporal smoothing. It rather considers all options for each frame, also the locally suboptimal, to find the sequence of labels that optimally combines the observations and the requirement of temporal stability. A drawback of modelling duration with an HMM, is that a standard HMM implies that state durations follow a geometric distribution. This downside can be alleviated by the use of an extended duration HMM, where the duration distribution can be trained as well (Chen et al., 2012). In hypothesis-search based systems such as those of Yoshioka et al. (2004); Sailer and Rosenbauer (2006); Sumi et al. (2008), chord duration is part of the information that determines whether the hypothesised labels are retained and further expanded.

Finally, the feature decoding can also be explicitly split into two stages. First, the signal is segmented into regions in which the label stays constant. Afterwards, the features are averaged over the found segments and labelled. The final label segmentation is in this case already determined before any acoustic model is used. This presegmentation of the features can be accomplished by harmonic change functions, such as the ones of Goto and Muraoka (1999); Harte et al. (2006); Li and Bello (2007). The systems of Morman and Rabiner (2006) for chord estimation and İzmirli (2007) for key estimation demonstrate that such an approach works in practice.

3.3.2.4 Contextual aspect

Besides modelling the duration of the labels, more musicological knowledge can be applied to estimate the likelihood of a label in the present context (determined by the surrounding labels). This way, one tries to take into account that music is not composed of labels that follow each other in a random order. Certain sequences are more likely to appear than others. For example, many of the rules that are taught in music theory relate to particular sequences of labels that are either strongly favoured or discouraged. Much like the acoustic models, the context models can be based on a musicological theory or trained on an annotated data set using machine learning techniques.

The most common way to integrate context information into a system is to use an HMM as feature decoder. The off-diagonal probabilities, or change probabilities, of the transition matrix can then take care of the contextual aspect. Examples for the case of chord estimation are the construction of a data-driven model on an annotated data set (Sheh and Ellis, 2003) or the use of a doubly-nested circle-of-fifths (Bello and Pickens, 2005) as a knowledge-based model of the distance between subsequent chords. As was the case for acoustic models too, the distinction between data-driven and knowledge-driven approaches is not always strict. The doubly-nested circle-of-fifths is used by Bello and Pickens (2005) to initialise the training process, but it can also be used on its own, as proven by Papadopoulos and Peeters (2007). Similar procedures exist for local key estimation. For

instance, the key transitions of Noland and Sandler (2006) are based on Krumhansl's key profile correlations (Krumhansl, 1990) (not to be confused with Krumhansl's probe tone ratings used for the acoustic model). They test these key transitions both as they are and as training initialisation.

A logical extension of the distinct key and chord change models is a change model in terms of relative chords in a key. After all, we know from music theory that harmony is mostly analysed as the movement of chord degrees in a key. Therefore we need the key while we recognise the chords. Two types of systems are suitable for this: either a sequential chord-after-key estimation or a simultaneous key-and-chord estimation system. Sequential chord-after-key systems use the generated key deterministically. The states of the chord-producing HMM then represent relative chords in the found key, which keeps the complexity constant in comparison to a chord-only decoder (Khadkevich and Omologo, 2009b). If keys and chords are recognised simultaneously, the states of the HMM represent key-chord combinations. The optimal key and chord sequences are then jointly decoded. This leads to a strong increase in the number of states and the size of the transition matrix. The number of states increases even more when concepts such as metric position or bass line are taken into account as well. To keep the number of variables tractable, the transition probabilities can be decomposed into smaller parts. The different approaches of context modelling in an HMM distinguish themselves by the relations they model and whether the information they encode is knowledge-based or data-driven. Burgoyne and Saul (2005) first proposed a model with key-chord states based on a theoretical classification of triads in a key. Catteau et al. (2007) used the same state space and derived their probabilities from Lerdahl (1988)'s theory on tonal distance. The states of Papadopoulos and Peeters (2011) were composed of chords and metric positions, with transitions set by expert logic. Mauch and Dixon (2010b) extended the search space to combinations of keys, chords (including inversions) and metric positions and used a dynamic Bayesian network (DBN), which is a generalisation of an HMM. The DBN parameters were set using musicological knowledge. This configuration inspired Ni et al. (2012) to create an entirely data-driven equivalent, although the DBN was reformulated to its equivalent HMM and the metric position was dropped as a variable. The required computation can optionally be limited by prematurely pruning unlikely key-chord combinations, as in Rocher et al. (2010), where the used knowledge is also based on Lerdahl's theory. Another way of reducing the complexity is to consider only the global key, such that a majority of entries in the transition matrix can be set to zero. This can equivalently be seen as constructing a separate key-dependent HMM for each key, instead of one large HMM. This approach was adopted by Lee and Slaney (2008) in their data-driven system.

A drawback of the usage of a standard HMM, is that it can only account for the immediate context, whereas we know from music theory that

looking at a wider context can be much more enlightening. For instance, a *D*min-*G*7-Cmaj sequence is more representative of a C major key than either *D*min-*G*7 or *G*7-Cmaj sequences. Khadkevich and Omologo (2009b) therefore used a two-pass decoding of the bigram HMM by rescoring the lattice produced in the first pass with a trigram or tetragram model probabilities.

A few non-HMM feature decoders also contain a contextual aspect. In hypothesis-search systems, context is one of the elements to determine whether a candidate label is retained and expanded in the search, just like the durational aspect is another factor. The score for each frame can depend on the fitness of a chord in a key (Sailer and Rosenbauer, 2006), or the chord transitions in a key and the compatibility between bass-line and chord Yoshioka et al. (2004); Sumi et al. (2008). The greedy-search decoder of Cheng et al. (2008) even uses higher order models up to tetragrams to this effect.

Finally, it should be remarked that sequential systems can contain two context models, one for each stage. The first context model is necessarily formulated in terms of either keys or chords. The results of the first stage allow the second context model to be expressed as relative chords in a key, as described previously, but this is not required. The second model can very well be formulated in terms of either chords or keys, whichever has not been used in the first stage, or even be absent altogether. In order to be considered a sequential system, the second stage decoder needs to use some information that is produced by the first stage, but it suffices that this information is used for the acoustic modelling. For instance, in the key-after-chord systems of Noland and Sandler (2009); Papadopoulos and Peeters (2012), the chords estimated in the first stage are used as features, but the states of the HMM that forms the second stage represent just keys. The chord information is therefore only used for the acoustic models, and the context aspect is the same as for a key-only estimation system. Actually, the transition matrix of such a system is perfectly interchangeable with the one of a key-only estimation system with chroma features, for example the one of Peeters (2006a). Going further, the system of Maddage (2006) also uses estimated chords as features in its second stage, without considering any context. The same can of course be said of all systems that determine a global key after estimating the chords (Shenoy and Wang, 2005; Weil et al., 2009). In contrast, simultaneous key-and-chord estimation systems can estimate keys without relying on a key acoustic model, as Burgoyne and Saul (2005) have demonstrated. The keys are then entirely derived from the context model in combination with the simultaneously estimated chords.

4 Modelling musicological knowledge

One of the main objectives of this thesis is to establish whether the addition of prior musicological knowledge can improve the estimation of keys and chords from audio. This background knowledge is what humans, musically schooled or not, acquire through exposition to music. It is thus culturally determined at least to some extent, although certain physical observations provide a very elegant explanation for some phenomena (Cook and Hayashi, 2008). The question of nature-versus-nurture therefore remains unsolved. This, however, is a question for musicologists, experimental psychologists and the like, and will not be answered in this work. We make abstraction of where this knowledge comes from, and are only interested in making a statistical model of it, such that it can be used in a probabilistic framework. In this chapter, and the subsequent chapters that build further on this one, musicological knowledge is interpreted in the strict sense of knowledge about key sequences, chord sequences and their interaction. As we have seen in the previous overview of the literature, other musical concepts such as metric structure and repetition can be included in a model of musicological knowledge, but we focus on the combination of keys and chords because relatively few works have explored their relation in depth.

4.1 Introduction

Both the temporal dependency and the intertwining relation between keys and chords that we have tried to make clear with words in the previous chapters, can be mathematically expressed as the joint probability $P(K, C)$.

Here $C = c_1, c_2, \ldots, c_J$ is the sequence of chords up to a certain point J in time and $K = k_1, k_2, \ldots, k_J$ is the sequence of corresponding keys up to that point. The used indices $j = 1, \ldots, J$ increase each time a chord changes, so by definition $c_j \neq c_{j\pm1}, \forall j$. The chord sequence therefore determines the segmentation and we assume that the chord segments form indivisible units of time. Key changes can therefore only occur together with chord changes, but they happen less frequently than chord changes, so $\exists j : k_j = k_{j\pm1}$. To illustrate exactly on which information we rely to build our model, we present an example. This will also serve to explain the further processing steps. In figure 4.1a, we can see the first eight measures of the jazz standard "Perdido"[1] in standard musical notation. We do not take the melody into account, so we only copy the chord sequence and discard timing information. The key labelling is not explicitly written out in the sheet music, but we do add it in our representation of the piece. The indexed result in table 4.1b is the raw format of the information that we will use to build our model with. Note that the derivation of this format from sheet music is just done for illustrative purposes. One of the reasons for using this specific sub-selection of musical information is precisely that it is easier to obtain than complete sheet music and that it is easier to reason about because it is more compact.

A first approximation we make in order to come to a model of feasible proportions, is to assume that there is a limit to how far in time a chord or key can exert influence on the following chords and keys. We assume a horizon that limits the influence of previous keys and chords to the $n-1$ most recent ones. Earlier chords or keys therefore become insignificant. This can be expressed as the $(n-1)$-th order Markov property

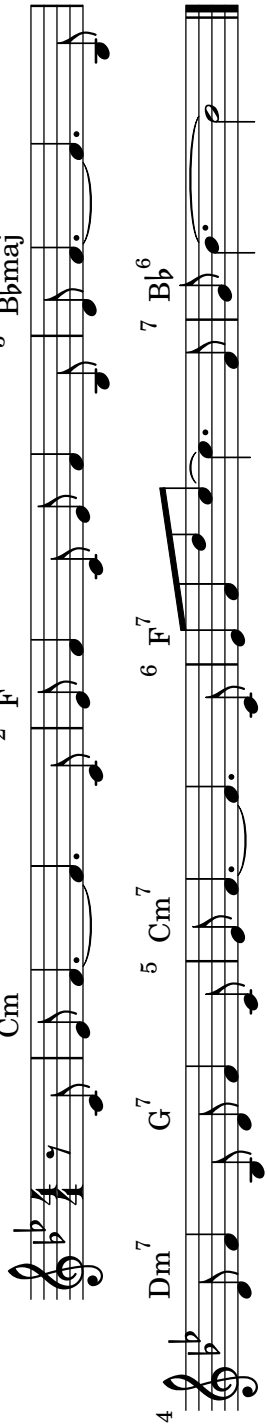
$$P(K, C; n) = \prod_{j=1}^{J} P\left(k_j, c_j | k_{j-1}, c_{j-1}, \ldots, k_{j-n+1}, c_{j-n+1}\right) \tag{4.1}$$

The conditional variables are called the *context*, therefore the *context size* is $n-1$. For the remainder of the text, we will use a more compact style of notation $P\left(k_j, c_j | k_{j-1}^{j-n+1}, c_{j-1}^{j-n+1}\right)$ for $P\left(k_j, c_j | k_{j-1}, c_{j-1}, \ldots, k_{j-n+1}, c_{j-n+1}\right)$. By applying Bayes' rule, the joint probability can be decomposed even further into

$$P(K, C; n) = \prod_{j=1}^{J} \underbrace{P\left(k_j | c_j, c_{j-1}^{j-n+1}, k_{j-1}^{j-n+1}\right)}_{\text{key transition probability}} \underbrace{P\left(c_j | k_{j-1}^{j-n+1}, c_{j-1}^{j-n+1}\right)}_{\text{chord change probability}} \tag{4.2}$$

We name the two resulting terms respectively the *key transition probability* and the *chord change probability*. The difference in naming comes from the

[1] As is typical for jazz standards, many different harmonisations and interpretations of this melody exist. For the purpose of this example, we just assume that this particular version and its harmonic annotation is given as irrefutable ground truth.



(a) Sheet music

key sequence	*B*♭ major			*C* major		*B*♭ major		
chord sequence	Cmin7	*F*7	*B*♭maj7	*D*min7	*G*7	Cmin7	*F*7	*B*♭maj6
index *j*	1	2	3	4	5	6	7	8

(b) Key and chord sequence

Figure 4.1: "Perdido" by Juan Tizol

fact that a change in index j implies a chord change, but not necessarily a key change. A key transition from segment j to $j+1$ can either be a key change or a key self-transition. We now want to separate the key change probabilities from the key self-transition probabilities because they represent different musicological concepts. This way, the resulting key change probability will be analogous in scope to the chord change probability. We can then propose similar ways to model these probabilities later on in the chapter. To this end, we apply the law of total probability to the key transition probability

$$P\left(k_j|c_j, k_{j-1}^{j-n+1}, c_{j-1}^{j-n+1}\right) = \begin{cases} P\left(k_j = k_{j-1}|c_j, k_{j-1}^{j-n+1}, c_{j-1}^{j-n+1}\right) & \text{if } k_j = k_{j-1} \\ \left(1 - P\left(k_j = k_{j-1}|c_j, k_{j-1}^{j-n+1}, c_{j-1}^{j-n+1}\right)\right) & \\ \quad \underbrace{P\left(k_j|c_j, k_{j-1}^{j-n+1}, c_{j-1}^{j-n+1}, k_j \neq k_{j-1}\right)}_{\text{key change probability}} & \text{if } k_j \neq k_{j-1} \end{cases} \tag{4.3}$$

In this formulation, we can recognise a separate *key change probability*. The term $P\left(k_j = k_{j-1}|c_j, k_{j-1}^{j-n+1}, c_{j-1}^{j-n+1}\right)$ is concerned with the duration of a key in terms of the number of chords it contains. It will be treated in the following chapters. In the next sections, we will define numerical models for both change probabilities that map the variables to probability values in the range $[0, 1]$. We thus require both a key change model f_{kc}

$$f_{kc} : \left(k_{j-n+1}, c_{j-n+1}, \ldots, k_{j-1}, c_{j-1}, k_j, c_j,\right) \mapsto [0, 1] \tag{4.4}$$

and a chord change model f_{cc}

$$f_{cc} : \left(k_{j-n+1}, c_{j-n+1}, \ldots, k_{j-1}, c_{j-1}, c_j\right) \mapsto [0, 1] \tag{4.5}$$

Not all of these variables need to be engaged in these models however. A part of the modelling can include deeming some context variables to be conditionally independent, and therefore not contributing to the model, or some of these variables can be tied or transformed to make some musicological relations more explicit. This is exactly what we will discuss next.

4.2 A case for relative models

We are not looking at chord and key sequences in isolation, but we are interested in their interaction instead. We will therefore make the relation between the two more explicit by using the musicologically more informative representation of relative chords in a key, as introduced in chapter 2.

This also reflects more closely the way scholars analyse harmony by hand. To recapitulate, we define a key k as the combination of a tonic t and a mode m and chord c as the combination of a root r and a type p. A key-chord pair is therefore equivalent to the quadruple of tonic, mode, root and type $(k, c) = (t, m, r, p)$. By expressing the root r as the chroma interval $i^{(tr)}$ between the tonic and the root: $i^{(tr)} = \mathcal{D}_{\text{up}}(t, r)$, we arrive at the definition of a relative chord c' with respect to a key k. Therefore we can equivalently express a key-chord pair as a key-relative chord pair $(k, c) = (t, m, r, p) = \left(t, m, i^{(tr)}, p\right) = (k, c')$.

4.2.1 Relative chord change models

To calculate the chord change probabilities $P(c_j | k_{j-1}^{j-n+1}, c_{j-1}^{j-n+1})$, we will reason in terms of relative chords and key modes. In particular, we express all chords $c_{j-n+1}, \ldots, c_j$ as relative chords $c'_{j-n+1}, \ldots, c'_j$ with respect to key k_{j-1}. We then assume that the chord change probability does not depend on the whole key k_{j-1}, but only on its mode m_{j-1}. This allows us to build distinct transition models for keys that differ only in mode, but not in tonic, for which distinct idiomatic chord sequences exist. On the other hand, the shift-invariance between keys that differ in tonic, but not in mode is also taken into account. This is our main motivation for modelling keys and chords together. It allows for a reasoning that closely follows the principles used in scholarly analysis of harmony. Systems such as the ones by Sheh and Ellis (2003); Bello and Pickens (2005); Papadopoulos and Peeters (2007) only estimate chords, and cannot make these distinctions. For sequences of two chords, we end up with just a mode and a pair of relative chords as a representation for the local harmony:

$$f_{cc2} : \left(c'_{j-1}, c'_j, m_{j-1}\right) \mapsto [0, 1] \tag{4.6}$$

For sequences of three chords or more, we have more context information at our disposal, but all chords are still interpreted as relative chords in the most recent key of the context k_{j-1}, of which we again only consider the mode m_{j-1}. For a trigram model (where $n = 3$), this amounts to

$$f_{cc3} : \left(c'_{j-2}, c'_{j-1}, c'_j, m_{j-1}\right) \mapsto [0, 1] \tag{4.7}$$

We advocate that the interpretation of the chords $c_{j-n+1}, \ldots, c_j$ in the context of the already hypothesised key k_{j-1} and not in the context of their corresponding keys $k_{j-n+1}, \ldots, k_j$ is in accordance with common practice in harmonic analysis. As such, it is more than just a simplification to reduce the number of parameters. Let us illustrate this with an example. Consider chords 4–8 of our example song in table 4.1b, giving us the fairly common chord sequence: *D*min7–*G*7–Cmin7–*F*7–*B*♭maj6. Its last three chords

(Cmin7–*F*7–*B*♭maj6) form a $II - V - I$ movement in the key of *B*♭ major. The first two chords *D*min7–*G*7 do not fit in this key, but they form a $II - V$ in the key of *C* major. We will now take a closer look at the chord change from *G*7 to Cmin7, since that is the only instance where $k_{j-1} \neq k_j$. First of all, even though both chords are labelled with different keys, it does not make sense to interpret them in their respective keys (a movement from a V in *C* major to a II in *B*♭ major in this case). In order to reason about this transition, we must interpret both chords in one and the same key. This can either be the key of the starting chord (a V to I movement in *C* major) or in that of the target (a VI to II movement in *B*♭ major). We now argue that it is musically more intuitive to interpret the transition between *G*7 and Cmin7 as a chord transition in *C* major and the transition between Cmin7 and *F*7 as a transition in key *B*♭ major because of the forward movement in music: the past context prepares the ear for what is coming, so the new chord c_j will be interpreted in the key of established context, irrespective of whether that new chord also marks the start of a new key or not. In practice, this means that each transition between the $n-1$ chords that form the context and the current chord should be interpreted in the key k_{j-1} of the last context chord. We will propose different chord change models in section 4.3, but all of them share this principle. They will therefore all have the same input domains, $\left(c'_{j-1}, c'_j, m_{j-1}\right)$ in case of bigram models and $\left(c'_{j-2}, c'_{j-1}, c'_j, m_{j-1}\right)$ in case of trigram models.

4.2.2 Relative key change models

For the key change probabilities $P\left(k_j | c_j, k_{j-1}^{j-n+1}, c_{j-1}^{j-n+1}, k_j \neq k_{j-1}\right)$, we will also perform a number of transformations. Unlike the chord change probabilities, we choose to define multiple models with different input domains. A first, simplified bigram model is obtained by considering the previous key to be the dominant factor in determining the current key. We therefore assume that the key k_j is conditionally independent from the chords c_j, c_{j-1}, given the context key k_{j-1}, such that the chords do not contribute to the probability:

$$P\left(k_j | c_j, k_{j-1}, c_{j-1}, k_j \neq k_{j-1}\right) = P\left(k_j | k_{j-1}, k_j \neq k_{j-1}\right) \tag{4.8}$$

The shift-invariance of music, which means in this case that transposing both keys over the same amount is perceptually insignificant, is here taken into account by making the model only a function of the chroma interval $i_j^{(tt)}$ between the two tonics t_j and t_{j-1}, instead of considering the two tonics themselves. The simplest bigram model is therefore expressed in terms of

$$f_{kc2a} : \left(i_j^{(tt)}, m_j, m_{j-1}\right) \mapsto [0, 1] \tag{4.9}$$

Two more complex models are derived from this by simply imposing extra conditions on its applicability, i.e. if the conditions are satisfied the model values are the same as f_{kc2a}, otherwise they are 0. The aforementioned conditions depend on the diatonicity of the chords in their key. Concretely, we derive 2 models

$$f_{kc2b} : \left(k_j, k_{j-1}, c_j\right) \mapsto [0,1] = \begin{cases} 0 & c_j \notin \mathbf{D}\left(k_j\right) \\ f_{kc2a} & c_j \in \mathbf{D}\left(k_j\right) \end{cases} \tag{4.10}$$

and

$$\begin{aligned} f_{kc2c} : \left(k_j, k_{j-1}, c_j, c_{j-1}\right) &\mapsto [0,1] \\ &= \begin{cases} 0 & c_j \notin \mathbf{D}\left(k_j\right) \vee c_{j-1} \notin \mathbf{D}\left(k_{j-1}\right) \\ f_{kc2a} & c_j \in \mathbf{D}\left(k_j\right) \wedge c_{j-1} \in \mathbf{D}\left(k_{j-1}\right) \end{cases} \end{aligned} \tag{4.11}$$

For f_{kc2b} we add the condition that when a new key is started, the first chord in the new key should be diatonic (in that key). The underlying motivation is that in order to communicate a new key to the listener, the played chord should be exemplary enough for this new key. This exemplarity is expressed by the binary condition of diatonicity. More differentiating ways to express the exemplarity of a chord for a key can be easily conceived, but those are left for future work. The f_{kc2c} model keeps this condition, but additionally requires that also the last chord of a key should be diatonic, such that the transition between two keys is clearly underlined.

Trigram models are formed very similarly, the extra variables are again used to impose more conditions that need to be satisfied before a non-zero value will be assigned. The larger context will be used to restrict the key change rate. Remember that the index j follows the chord changes, so when analysing a sequence of three keys, this does not imply that there are two key changes present. On the contrary, that would mean that the middle key would only last one chord and this does not comply with the definition that a key is a content label on a higher temporal level than a chord. A harmonic analysis returning such results would therefore be undesirable, and that is why we explicitly forbid this. The simplest trigram model, where we again consider the key k_j to be conditionally independent of all chords, is therefore:

$$f_{kc3a} : \left(k_j, k_{j-1}, k_{j-2}\right) \mapsto [0,1] = \begin{cases} 0 & k_{j-2} \neq k_{j-1} \\ f_{kc2a} & k_{j-2} = k_{j-1} \end{cases} \tag{4.12}$$

When we add the contribution of the chords again, we get the more complex models

$$f_{kc3b} : \left(k_j, k_{j-1}, k_{j-2}, c_j\right) \mapsto [0,1] = \begin{cases} 0 & k_{j-2} \neq k_{j-1} \vee c_j \notin \mathbf{D}\left(k_j\right) \\ f_{kc2a} & k_{j-2} = k_{j-1} \wedge c_j \in \mathbf{D}\left(k_j\right) \end{cases} \tag{4.13}$$

and

$$f_{kc3c} : \left(k_j, k_{j-1}, k_{j-2}, c_j, c_{j-1}\right) \mapsto [0,1] = \begin{cases} 0 & k_{j-2} \neq k_{j-1} \vee c_j \notin \mathbf{D}\left(k_j\right) \vee c_{j-1} \notin \mathbf{D}\left(k_{j-1}\right) \\ f_{kc2a} & k_{j-2} = k_{j-1} \wedge c_j \in \mathbf{D}\left(k_j\right) \wedge c_{j-1} \in \mathbf{D}\left(k_{j-1}\right) \end{cases} \tag{4.14}$$

All together, we can see that we only need to define one model f_{kc2a}, from which we can derive a number of alternatives. One possible definition of f_{kc2a} will be explained in section 4.4.

4.3 Constructing relative chord change models

We will consider three types of relative chord change models: a uniformly distributed model that serves as a base-line, data-driven models learned from multiple annotated data sets, and a theoretical model based on Lerdahl's chord distance (Lerdahl, 2001, p.55). Irrespective of the model type, we first need to decide which chord and key vocabularies $\mathbf{C}$ and $\mathbf{K}$ to use. The vocabulary sizes are N_c and N_k, such that $\mathbf{C} = \{C_1, C_2, \ldots, C_{N_c}\}$ and $\mathbf{K} = \left\{K_1, K_2, \ldots, K_{N_k}\right\}$. In practice, the chords are obtained by distinguishing N_P chord types and 12 possible roots, such that $N_c = 12N_P$. Similarly, a number of modes N_M and 12 possible tonics define $N_k = 12N_M$ keys. For the remainder of this text, we will use four triads – maj, min, dim and aug – leading to a $N_c = 48$ and limit the modes to major and minor, giving $N_k = 24$. This selection is in line with earlier work (Harte and Sandler, 2005; Burgoyne et al., 2007) and provides a maximal harmonic discrimination with a minimum number of chord types and modes.

4.3.1 Learning models from a symbolic data set using Kneser-Ney smoothing

Given a corpus of symbolic key and chord labels, we can learn the probabilities of key and chord sequences, similar to the way humans build up certain expectations after listening to music. If a certain combination appeared

Original chords	*B*sus4	*F*♯maj	*E*maj/*F*♯	*E*maj	*B*maj7
Reduced chords	*X*	*F*♯maj	*E*maj	*E*maj	*B*maj
Merged chords	*X*	*F*♯maj	*E*maj		*B*maj

Figure 4.2: Preprocessing of chord labels

a lot, it will likely appear again. An easy way to exploit this information is to count all occurrences of a certain sequence in a corpus and to use its relative frequency as a measure of its probability. The drawback of this approach is that the resulting probabilities will be too specific to the data set. In particular, any unseen sequence gets a zero probability, even though it is entirely possible that the sequence appears in another data set. To alleviate this problem, we construct a backoff model with Kneser-Ney smoothing (Kneser and Ney, 1995). This technique, originally developed for natural language models for speech recognition, derives the strictly positive probability of an unseen combination from the probabilities of its lower order subsequences. The procedure that leads to these probabilities, along with some chord-specific adaptations, is detailed below for $n = 1$ to 3.

PREPROCESSING First of all, the chord and key annotations in the data set are reduced to the vocabularies of the model. This mapping is based on the presence or absence of certain chroma intervals in the chord. Complete details can be found in appendix A. If no musically meaningful mapping can be found for a chord or if no chord is annotated, it is replaced by the unknown symbol "X". For example, a sus4 chord cannot reasonably be mapped to one of the 4 triads in our example, so it is replaced by X. As a final preprocessing step, identical labels in succession, whether due to the vocabulary reduction (e.g. maj7 and 7 chord types both mapping to maj) or whether present in the original annotation, are merged. An example of this process for a sequence of chord labels can be seen in figure 4.2.

RELATIVE CHORD COUNTING The next step is simply counting occurrences of relative chord and mode combinations. This implies that we are no longer working with key and chord labels of vocabularies $\mathbf{K}$ and $\mathbf{C}$, but with the corresponding relative chord vocabulary $\mathbf{C}' = \left\{C'_1, C'_2, \ldots, C'_{N_c}\right\}$ and mode vocabulary $\mathbf{M} = \left\{M_{1,} M_2, \ldots, M_{N_M}\right\}$. To obtain sequences of relative chords in a mode, we slide a window of length n over the chords and process the chords in that window. If all $n-1$ chords of the context are in the same key, all n chords are interpreted in the context key and the count of the relative chord sequence is increased. If there is a key change in the context (only possible when $n \geq 3$), the n-gram gets discarded because it is not representative enough. In case there is an X anywhere in the n-gram, the n-gram is ignored as well. For the case of $n = 1$ where there is

unigram counts	
cnt $(II\text{min}, \text{major})$	2
cnt $(V\text{maj}, \text{major})$	2
cnt $(I\text{maj}, \text{major})$	2

(a)

bigram counts	
cnt $(II\text{min} - V\text{maj}, \text{major})$	2
cnt $(V\text{maj} - I\text{maj}, \text{major})$	2
cnt $(I\text{maj} - VI\text{min}, \text{major})$	1

(b)

trigram counts	
cnt $(II\text{min} - V\text{maj} - I\text{maj}, \text{major})$	2
cnt $(V\text{maj} - I\text{maj} - VI\text{min}, \text{major})$	1

(c)

Table 4.1: Resulting counts for the example sequence in figure 4.3

no context, a chord c_j is always interpreted in its own key k_j. This process is illustrated for unigram, bigram and trigram counts in figure 4.3, where the reason for rejecting an n-gram is also stated if applicable. The result is that each combination of relative chords and modes $\left(c'_1, \ldots, c'_{n-1}, c'_n, m\right)$, where $c'_1, \ldots, c'_{n-1}, c'_n \in \mathbf{C}'$ and $m \in \mathbf{M}$, has a non-negative counter associated with it: cnt $\left(c'_1, \ldots, c'_{n-1}, c'_n, m\right) \geq 0$. The non-zero counts for our example can be found in table 4.1. Using Kneser-Ney smoothing, these counters are subsequently used to get strictly positive probabilities, also for combinations that were not encountered in the corpus. The idea behind this technique is that the probabilities of all encountered combinations are set slightly below their ratio of occurrence in the data set. The freed probability mass is then distributed over the unseen combinations in proportion to the probabilities of their subsequences. This procedure works iteratively starting from unigram probabilities.

UNIGRAM MODEL (N=1) The calculation of the unigram probabilities still starts out as the proportion of the counts to all relative chords in the data set:

$$P\left(c'_n | m\right) = \frac{\text{cnt}\left(c'_n, m\right)}{\sum_{c' \in \mathbf{C}'} \text{cnt}\left(c', m\right)} \tag{4.15}$$

Depending on the choice of data set and chord vocabulary, it is definitely possible that a particular relative chord is never encountered and therefore gets probability zero. However, due to the shift of key that is used for interpreting chords in key k_{j-1} in higher order n-grams, this does not preclude it from appearing in a longer chord sequence. An example of this is the VImin relative chord in figure 4.3, which arises as the interpretation of Amin in Cmajor in the bigram and trigram case, but not in the unigrams. Note that the unigram probabilities are required to be strictly posit-

Preprocessed keys	Cmajor			Gmajor			
Preprocessed chords	*D*min	*G*maj	*C*maj	*A*min	*D*maj	*G*maj	*X*
Relative unigrams	*II*min	*V*maj	*I*maj	*II*min	*V*maj	*I*maj	*X*
Relative bigrams	*II*min	*V*maj					
		*V*maj	*I*maj				
			*I*maj	*VI*min			
				*II*min	*V*maj		
					*V*maj	*I*maj	
						*X*in bigram	
Relative trigrams	*II*min	*V*maj	*I*maj				
		*V*maj	*I*maj	*VI*min			
			key change in context				
				*II*min	*V*maj	*I*maj	
					*X*in trigram		

Figure 4.3: Example of counting bigrams and trigrams in an annotation

ive when they are needed as back-off for higher order probabilities. Therefore we introduce an extra smoothing step specifically for the unigrams in case the (mode-specific) number of unseen relative chords $N_u(m) > 0$. We add a fixed amount $\underline{U}$ to each of the unigram counters in order to make them strictly positive:

$$P\left(c_1'|m\right) = \frac{U + \text{cnt}\left(c_1', m\right)}{N_c U + \sum_{c' \in \mathbf{C}'} \text{cnt}\left(c', m\right)} \tag{4.16}$$

where N_c is the size of the chord vocabulary.

We can make this amount U musicologically more meaningful by relating it to the fraction P_u of the total probability mass that is reserved to distribute uniformly over the unseen unigrams. As a result, every unseen unigram get assigned a probability of $\frac{P_u}{N_u(m)}$, and all unigrams present in the data set get a higher probability:

$$\frac{P_u}{N_u(m)} = \frac{U}{N_c U + \sum_{c' \in \mathbf{C}'} \text{cnt}\left(c', m\right)} \iff$$
$$U = \frac{P_u}{N_u(m) - N_c P_u} \sum_{c' \in \mathbf{C}'} \text{cnt}\left(c', m\right) \tag{4.17}$$

BIGRAM MODEL ($n = 2$) From the bigram probabilities on, we have the possibility to back off to the model of one order below in case there are not enough occurrences of a certain bigram to reliably estimate its probability. The Kneser-Ney smoothing we use takes two parameters: the *pruning limit* L, which gives the minimum number of times an n-gram needs to occur in order to be deemed representative enough, and the *discount* D, which gives the amount that is subtracted from the counts to create probability mass for the unseen combinations. They are linked by the requirement $D \leq L$. The bigram probabilities are then calculated as

$$P\left(c_2'|c_1', m\right) = \begin{cases} \dfrac{\text{cnt}\left(c_1', c_2', m\right) - D}{\sum\limits_{\substack{c' \in \mathbf{C}': \\ \text{cnt}\left(c_1', c', m\right) > L}} \text{cnt}\left(c_1', c', m\right)} & \text{cnt}\left(c_1', c_2', m\right) > L \\ \gamma_1\left(c_1', m\right) P\left(c_2'|m\right) & \text{cnt}\left(c_1', c_2', m\right) \leq L \end{cases} \tag{4.18}$$

We can see that the frequent bigrams get probabilities that are roughly equal to their proportion of occurrence (after subtraction of the discount), but less frequent bigrams fall back on the probabilities of the unigram model, weighted by a so-called *backoff factor* γ_1. This backoff factor is calculated such that the total probability is preserved: $\sum_{c'} P\left(c'|c_1', m\right) = 1$. This

requirement is satisfied by the following equation:

$$\gamma_1\left(c'_1, m\right) = \frac{1 - \sum\limits_{\substack{c' \in \mathbf{C}': \\ \mathrm{cnt}\left(c'_1, c', m\right) > L}} P\left(c' | c'_1, m\right)}{1 - \sum\limits_{\substack{c' \in \mathbf{C}': \\ \mathrm{cnt}\left(c'_1, c', m\right) > L}} P\left(c' | m\right)} \tag{4.19}$$

TRIGRAM MODEL AND HIGHER ($n \geq 3$) Probabilities for the trigram model and higher follow the same pattern as the bigram: frequently observed combinations are calculated as a discounted proportion of the occurrences and infrequent ones are based on the lower order probability weighted by the backoff factor γ_{n-1}

$$P\left(c'_n | c'_1, \ldots, c'_{n-1}, m\right) = \begin{cases} \dfrac{\mathrm{cnt}\left(c'_1, \ldots, c'_{n-1}, c'_n, m\right) - D}{\sum\limits_{\substack{c' \in \mathbf{C}': \\ \mathrm{cnt}\left(c'_1, \ldots, c'_{n-1}, c', m\right) > L}} \mathrm{cnt}\left(c'_1, \ldots, c'_{n-1}, c', m\right)} & \\ & \mathrm{cnt}\left(c'_1, \ldots, c'_{n-1}, c'_n, m\right) > L \\ \gamma_{n-1}\left(c'_1, \ldots, c'_{n-1}, m\right) P\left(c'_n | c'_2, \ldots, c'_{n-1}, m\right) & \\ & \mathrm{cnt}\left(c'_1, \ldots, c'_{n-1}, c'_n, m\right) \leq L \end{cases} \tag{4.20}$$

with

$$\gamma_{n-1}\left(c'_1, \ldots, c'_{n-1}, m\right) = \frac{1 - \sum\limits_{\substack{c' \in \mathbf{C}': \\ \mathrm{cnt}\left(c'_1, \ldots, c'_{n-1}, c', m\right) > L}} P\left(c' | c'_1, \ldots, c'_{n-1}, m\right)}{1 - \sum\limits_{\substack{c' \in \mathbf{C}': \\ \mathrm{cnt}\left(c'_1, \ldots, c'_{n-1}, c', m\right) > L}} P\left(c' | c'_2, \ldots, c'_{n-1}, m\right)} \tag{4.21}$$

In order to use a model of order n, it suffices to store the probabilities of all orders up to n that are common enough in the data set and their corresponding back-off factors up to order $n - 1$. For natural language modelling, this generally results in a model that takes less memory compared to one that stores the probabilities of all combinations of order n. For chord sequence modelling however, this advantage is of rather limited significance, because typical chord vocabularies are rarely of such a size that memory becomes an issue.

The data sets we use to derive these models from will be described in section 4.5 and some ways to quantify the expressive power of these models in section 4.6.

4.3.2 Deducing models from Lerdahl's chord distance

The drawback of any probabilistic model derived from symbolic annotations, is that the model is only as good as the data set. Specifically, if the

data under test is significantly different from the data used to build the model, the statistics encoded in the model are no longer relevant. The effect is that the model is no longer helpful for automatic estimation and may even worsen the outcome by supporting outputs that conform to inappropriate probabilistic distributions. Therefore, a data set should preferably be large and cover a variety of musical genres. Due to the non-triviality of the annotation process, the data sets at our disposal do not fully comply with these requirements. This does not mean that the models derived from these sets are completely useless. It just means that the results attained with these models might be overly pessimistic or optimistic, depending on the data they are tested on.

As an alternative, we also constructed a model based on musical theory, because such a model does not require any data set at all. Unfortunately, there is no theory offering a numerical expression of the probability of occurrence of a chord sequence. On the other hand, expressions of perceptual musical distance between chords in a key do exist. We can then use these distances to approximate the sequence probabilities required for our model by assuming that chords that are perceptually close to each other are more likely to follow each other in sequence than perceptually more distant chords.

In analogy to the distances between notes and between chromas as mentioned in chapter 2, music theorists have come up with different numerical distances between chords in keys. A well known one, and also the one we will be using, is Lerdahl's chord distance. It has previously been used to construct the model incorporated in the system of Catteau et al. (2007).

4.3.2.1 Lerdahl's chord distance

Lerdahl's theory in its most general form expresses the distance between a chord in a key and another chord in a possibly different key. However, to stay in line with the models annotated on a symbolic set, we only need the distances between chords in the same key, a more specific version of his general theory. The Lerdahl distance between chords in a constant key is only dependent on the relative chord and the mode, just like we assumed in our derivation (thereby justifying this decision). To make this explicit, the distance will be expressed between relative chords in a mode. The calculation of the distance itself is based upon the definition of a hierarchy of chroma intervals in a chord, implying that some classes are more important than others in the perception of a chord. This is represented by the so-called *basic space*. Note that this is not a geometric space wherein the geometric distance is related to the perceptual distance, but an *algebraic space*, i.e. just a tool to visualise the calculations involved. This basic space organises the chroma intervals of a chord into five levels of decreasing stability with a top-down propagation, meaning that every chroma interval in

root space			2									
fifth space			2							9		
triad space			2			5				9		
diatonic space	0		2		4	5		7		9		11
chromatic space	0	1	2	3	4	5	6	7	8	9	10	11

Figure 4.4: Lerdahl's basic space for IImin in major

one level also appears in all lower levels. The topmost level in the hierarchy is called the *root space*, only the root is present there. It is followed by the *fifth space* and the *triad space* at levels two and three. As the name implies, respectively the fifth and the remaining chroma intervals of the chord are added at these levels. The fourth level is named the *diatonic space*, encompassing all chroma intervals of the mode. The basic space is completed by the *chromatic space* at level five, which includes all twelve chroma intervals in an octave.

We do not use the usual representation for relative chords, where the chord type is expressed as a set of distances to the root, because we require a representation that is independent of the root chroma interval. Instead, all components of the relative chord are expressed as distances to the tonic, just like the components of the mode. Because the difference between enharmonically equivalent chroma intervals is ignored, these distances to the tonic chroma intervals are noted as the number of semitones from the tonic. For example, a IImin chord, which is shorthand for $II\,\{P1, m3, P5\}$, can be noted in semitone distances as $2\,\{0, 3, 7\}$. Factoring out the dependence of the chord type on the root chroma interval is accomplished by simply adding the numeric value of the root to all chord components, giving $\{2, 5, 9\}$. Representing this chord (determining levels one to three) in the major mode (determining levels four to five) gives the basic space depicted in figure 4.4.

From the basic space representations of chords, Lerdahl's chord distance can be calculated as a sum of two parts. The first part is the number of *chroma interval differences*, which is defined as the combined number chromas over all levels of the hierarchy that are present in the destination chord, but not in the starting chord. For example, for the distance between IImin and Vmaj in major (notated as $\mathcal{D}_{LC}\,(II\text{min}, V\text{maj}, \text{major})$), the number of chroma interval differences is 4. This example is visualised in figure 4.5, where the basic space of Vmaj ($= V\,\{P1, M3, P5\} = 7\,\{0, 4, 7\} = \{7, 11, 2\}$) is given and the chroma intervals that are not present in IImin are marked in boldface. We can see that in each of the two highest levels (root space and fifth space), there is one chroma present in the destination chord that is not present in the starting chord (i.e. 7). In the third level (triad space), two new chromas are present (i.e. 7 and 11). There is no difference

root space								**7**				
fifth space			2					**7**				
triad space			2					**7**				**11**
diatonic space	0		2		4	5		7		9		11
chromatic space	0	1	2	3	4	5	6	7	8	9	10	11

Figure 4.5: Lerdahl's basic space for Vmaj in major. The numbers in boldface indicate the chroma interval differences with respect to the basic space of IImin in major (seen in figure 4.4)

between the two lowest levels of IImin in major and Vmaj in major[2]. The sum of the chroma interval differences over all levels is therefore equal to 4. Mathematically speaking, the construction of a multi-level basic space effectively serves as a kind of weighting curve such that changes in some chroma intervals are accounted for more than others.

The second part of the chord distance is the minimum number of steps along the diatonic circle (as seen in figure 2.6) between the two root chroma intervals. For root chroma intervals II and V, this is 1, such that in total $\mathcal{D}_{LC}\left(II\text{min}, V\text{maj}, \text{major}\right) = 4 + 1 = 5$.

4.3.2.2 Converting distances to probabilities

Because Lerdahl's theory gives a distance between two relative chords, there is no obvious way to derive a trigram model from it. In order to derive a trigram probability $P(c'_j | c'_{j-1}, c'_{j-2}, m_{j-1})$ that cannot be trivially decomposed into the combination of two bigrams $\left(c'_{j-2}, c'_{j-1}, m_{j-1}\right)$ and $\left(c'_{j-1}, c'_j, m_{j-1}\right)$, the relative chord c'_{j-2} needs to exert some influence on c'_j as well. The only solution, given that Lerdahl's distance works pairwise, would be to take the triangular distance $\mathcal{D}_{LC}\left(c'_{j-2}, c'_j, m_{j-1}\right)$ into account as well. However, this would lead to a disproportionate high probability for trigrams of the form $\left(c'_j, c'_{j-1}, c'_j, m_{j-1}\right)$, which is musically unwanted. Therefore we will only derive bigram probabilities from Lerdahl's chord distance.

As mentioned before, we convert distances to probabilities based on the assumption that chords that are close to each other are more likely to follow each other in succession. Small distances therefore need to be converted to large probabilities and this is done by applying a normalised exponential, or softmin function, of the form $e^{-v\mathcal{D}_{LC}}$, where v is inversely proportional to the mean chord distance. A consequence of the use of the

[2]As long as the two chords are considered in the same key, as is required for the relative chord change model, there will never be any chroma interval differences in the lowest two levels.

diatonic circle in the definition of Lerdahl's chord distance, is that it is undefined for distances involving one or two non-diatonic roots. Therefore, we use Lerdahl's chord distance only when both chords are diatonic. How the other probabilities are determined will be explained next.

The total probability mass is first divided into a part P_d which is available for transitions to a relative chord that is diatonic in m_{j-1} and a part $1 - P_d$ which is available for non-diatonic targets. As we already mentioned, transitions originating from a diatonic relative chord as well as ending in a diatonic relative chord (hereafter called diatonic transitions) are distributed according to Lerdahl's chord distance. The other transitions, from or to a non-diatonic chord or both, get assigned a uniform distribution. The final formulation of the theoretical relative chord model is then as follows

$$P(c'_j|c'_{j-1}, m_{j-1}) = \begin{cases} \nu_c P_d \exp\left(-\frac{\mathcal{D}_{LC}(c'_j, c'_{j-1})}{\overline{\mathcal{D}_c}}\right) & c'_j, c'_{j-1} \in \mathbf{D}(m_{j-1}) \\ \frac{P_d}{N_d(m_{j-1})} & c'_j \in \mathbf{D}(m_{j-1}) \wedge c'_{j-1} \notin \mathbf{D}(m_{j-1}) \\ \frac{1-P_d}{N_{nd}(m_{j-1})} & c'_j \notin \mathbf{D}(m_{j-1}) \end{cases} \tag{4.22}$$

where $\mathbf{D}(m)$ represents the set of relative chords that are diatonic in mode m. The quantities $N_d(m)$ and $N_{nd}(m)$ represent the number of diatonic, respectively non-diatonic chords for mode m. The factor $\overline{\mathcal{D}_c}$ is the mean Lerdahl distance between two diatonic chords in a key. The normalisation factor ν_c makes sure that:

$$\forall c'_{j-1} \in \mathbf{C}', m_{j-1} \in \mathbf{M} : \sum_{\substack{c'_j \in \mathbf{C}' \\ c'_j \neq c'_{j-1}}} P(c'_j|c'_{j-1}, m_{j-1}) = 1 \tag{4.23}$$

Even though the Lerdahl distance is symmetric[3] and therefore the diatonic transitions are too, the resulting model is asymmetric due to the non-diatonic transitions. The P_d factor controls how strong relative chords are steered towards diatonicity. The degenerate cases of setting it to 1 or 0 will only allow sequences with strictly diatonic, respectively non-diatonic, relative chords.

4.4 Constructing a key change model

Since the number of key changes in a corpus of music typically is an order of magnitude lower than the number of chord changes, we do not have

[3]To be precise, Lerdahl's chord distance is symmetric as long as the two involved chords consist of the same number of chromas, which is the case for our choice of chord vocabulary.

enough data available to reliably learn the probabilities of key changes. As such, we will only rely on a theoretical model for the key change probabilities. Luckily, Lerdahl's theory also defines a distance between two keys, which we can convert to a probability in a similar manner as the chord distance. This so-called regional distance has also been used for aligning a query key sequence with all sequences in a database in order to retrieve the most similar one (İzmirli, 2005b). The resulting theoretical key change model has been previously developed at our lab (Catteau et al., 2007).

4.4.1 Lerdahl's regional distance

Lerdahl's regional distance (region is Lerdahl's term for key) (Lerdahl, 2001, p.68) is a special form of his more general distance between two key-chord combinations. This means that the two keys are expanded to two key-chord pairs before the actual calculation begins. The diatonic triad on the first degree is considered as the most exemplary for a key, so the regional distance is actually defined as the distance between the diatonic triads on the first degree of the respective keys. This is Imaj for major keys and Imin for minor keys. The distance only depends on the chroma interval between the two tonics, not on the tonics themselves, as well as on both modes. Here too, we will make this dependency explicit by formulating the regional distance in terms of the two modes and the interval between the two tonics. It is symbolised by $\mathcal{D}_{LR}\left(i_j^{(tt)}, m_{j-1}, m_j\right)$.

Because the keys involved in the key distance are expanded to key-chord combinations, they also can be represented in the basic space. The only additional specification required is that the distance is now expressed in semitones from the *starting* tonic. Therefore the representation of the target key most likely does not have its tonic on 0, contrary to the starting key and to the key-chord combinations involved in the chord distance within one key as used for the relative chord change model. The starting key in $\mathcal{D}_{LR}(P5, \text{major}, \text{major})$ is therefore depicted in figure 4.6. By definition, choosing the major mode for the starting key means that its representation is equal to that of a "Imaj in major" key-chord combination. The basic space representation of the target key can be seen in figure 4.7. The target tonic is a perfect fifth higher than the starting tonic, which is equal to 7 semitones, so it is at position 7. Each component of the major mode $\{0, 2, 4, 5, 7, 9, 11\}$ therefore has to be increased by 7, applying modulo 12, giving $\{7, 9, 11, 0, 2, 4, 6\}$ at the diatonic space level. Similarly, we also need to add 7 to the formula of the Imaj chord $0\,\{0, 4, 7\}$ to get $\{7, 11, 2\}$. The topmost three levels are thus equal to the representation of "Vmaj in major", as can be seen by comparing figure 4.7 to figure 4.5. In both cases, the maj chord $\{0, 4, 7\}$ is placed 7 semitones higher than the reference tonic (which is 0). For Vmaj in major, this is due to the root being a perfect fifth higher, but in the same key $(7 + 0)$, whereas in the case of Imaj in $+P5$ ma-

root space	0											
fifth space	0							7				
triad space	0				4			7				
diatonic space	0		2		4	5		7		9		11
chromatic space	0	1	2	3	4	5	6	7	8	9	10	11

Figure 4.6: Lerdahl's basic space for a major mode starting key

root space								**7**				
fifth space			**2**					7				
triad space			**2**					7				**11**
diatonic space	0		2		4		**6**	7		9		11
chromatic space	0	1	2	3	4	5	6	7	8	9	10	11

Figure 4.7: Lerdahl's basic space for the major mode target key a perfect fifth higher than the starting key. The numbers in boldface indicate the chroma interval differences with respect to the basic space of a major mode starting key (seen in figure 4.6)

jor, it is due to the tonic of the target key being a fifth higher, but with the root of the chord equal to the tonic $(0 + 7)$. The result is two times the same chord, attained through different reasonings. On the other hand, the difference between the two is clearly visible on the diatonic space level. Again, the chroma interval differences on multiple hierarchical levels can be interpreted mathematically as a balancing factor in order to weigh changes in chroma differently according to their position in the key. For a full musicological justification of the different hierarchical levels, the reader is referred to Lerdahl (2001).

The numerical value of the key distance is further calculated in a familiar fashion. Again the number of new chroma intervals in the target basic space with respect to the starting basic space is counted, as indicated in figure 4.7. This is 5 for our example, and now also includes a contribution on the diatonic level (1 chroma interval difference in each of the root space, fifth space and diatonic space level, and 2 in the triad space level). We add this to the minimum number of steps between the roots (per definition equal to the tonics) along the diatonic circle of the target key, 1 in our example. This implies that the starting tonic needs to be diatonic in the target key for the regional distance to be defined. A third part is added to the calculation of the distance, in addition to these two parts that are the same as for the chord distance calculation. The minimum number of steps between the tonics along the circle-of-fifths, as seen in figure 2.3, is also added to the sum. This gives us a total key distance of $\mathcal{D}_{LR}(P5, \text{major}, \text{major}) = 5 + 1 + 1 = 7$. Because the definition uses once more the diatonic circle, on which not all chroma intervals are present,

	to major											
	0	1	2	3	4	5	6	7	8	9	10	11
from major	**0**	23	14	14	16	**7**	30	**7**	16	14	14	23
from minor	**7**	16	21	**7**	23	14	21	14	**9**	21	**10**	23
	to minor											
	0	1	2	3	4	5	6	7	8	9	10	11
from major	**7**	23	**10**	21	**9**	14	21	14	23	**7**	21	16
from minor	**0**	23	14	14	16	**7**	30	**7**	16	14	14	23

Table 4.2: Lerdahl's regional distance for all combinations of modes and chroma intervals between the tonics

only the distance between nearby keys can be calculated this way. The distance to keys that are further away is calculated by splitting it into a chain of multiple distances between keys that are pairwise nearby. The exact division into smaller distances is selected on the basis of a minimum distance rule, but the precise details are outside the scope of this text. A summary of the resulting distances can be found in table 4.2. The bold numbers indicate the 14 distances that can be directly calculated using the procedure described above, the others are subsequently derived from these 14 by adding the distances that form the shortest chain. For example, $\mathcal{D}_{LR}(M7, \text{major}, \text{minor})$ can be split into

$$\begin{aligned}\mathcal{D}_{LR}(P5, \text{major}, \text{major}) + \mathcal{D}_{LR}(M3, \text{major}, \text{minor}) = \\ \mathcal{D}_{LR}(7, \text{major}, \text{major}) + \mathcal{D}_{LR}(4, \text{major}, \text{minor}) = 7 + 9 = 16 \quad (4.24)\end{aligned}$$

4.4.2 From distance to probability

To convert Lerdahl's key distances to probabilities, we make the assumption again that keys close to each other are likely to appear in sequence and thus receive a high transition probability. One of the weaknesses in this assumption is that this gives inadequate probabilities for some key changes such as the so-called "gear change" or "one up" which are common in pop music, but not in music of the Common Practice period on which Lerdahl's theory is based. This illustrates that although theoretical models do not have a bias towards a particular data set, their applicability is still data-dependent to some extent. In this case, the dependency is determined by the degree to which the data set complies with the theory.

The change transitions are determined by Lerdahl's regional distance, following a procedure similar to the chord change model.

$$P(k_j|k_{j-1}, k_j \neq k_{j-1}) = \nu_k \exp\left(-\frac{\mathcal{D}_{LR}(k_{j-1}, k_j)}{\overline{\mathcal{D}_k}}\right) \tag{4.25}$$

with $\overline{\mathcal{D}_k}$ being the mean distance between two keys, and with ν_k representing a normalisation factor. The latter must ensure that,

$$\forall k_{j-1} \in \mathbf{K} : \sum_{\substack{k_j \in \mathbf{K} \\ k_j \neq k_{j-1}}} P(k_j|k_{j-1}) = 1 \tag{4.26}$$

4.5 Data sets

In order to build the data-driven models and to evaluate their quality, we need a set of annotated music files. There are two types of data sets. First there exists annotated audio with its keys and chords transcribed and synchronised to the audio. This kind of data is the most versatile in use because the annotations can be used on their own to build key and chord change models. Furthermore, it exposes the exact durations of the labels in the data. The annotated audio can also be used to evaluate the automatic estimation of keys and chords from audio in the following chapters. We have two such sets at our disposal. The first one is called *SEMA*. It comprises 142 song excerpts of 30 seconds in a variety of genres and tempi. The excerpts have been annotated at our lab by a professional musician with absolute pitch hearing. This set was originally assembled to provide a broad sample of rhythmic figures (Varewyck and Martens, 2007). Therefore it also contains some excerpts that are mostly rhythmically oriented. The original selection contained nineteen more files, but the human annotator judged that there was no harmony present in these files. As such, the remaining 142 songs range from straightforward harmony to borderline inharmonic, but each of these files was deemed by the human annotator to have at least one segment with a chord and a key. The complete list of artists and song titles can be found in appendix B.

The second data set is called *Isophonics*. It has been annotated at the Centre for Digital Music at Queen Mary, University of London (Mauch et al., 2009). More specifically, we take the subset that was used for the audio chord estimation task at MIREX 2009 and all editions since[4]. It contains 210 complete songs sourced from 12 studio albums by the Beatles (174 songs), 2 compilations by Queen (18 songs) and one cover-album by Zweieck (18 songs). Because of the limited number of artists in the set, it contains less sonic variety. All songs are also textbook examples of pop harmony, the few experimental album tracks by the Beatles have been excluded.

[4] `http://www.music-ir.org/mirex/wiki/2009:Audio_Chord_Detection`

Other types of data we have at our disposal consist of chord and key sequences in symbolic form, without associated audio files. Therefore they can only be used to construct symbolic models, and not to evaluate automatic keys and chord estimation. The symbolic sequences can come with some basic timing information, expressed in terms of a number of measures for instance, or without any timing information at all, meaning that the exact duration is not available either. One such data set comes from the University of Alicante and is called *9GDB*. It consists of 856 files in total, divided into three groups containing three genres each, leading to nine different genres in total. The "academic" group contains 235 pieces split over the genres "baroque" (56), "classical" (50) and "romanticism" (129). The "jazz" group has 338 files divided over "pre-bop" (178), "bop" (94) and "bossa nova" (66). Finally, the "popular" group consists of 283 files in the genres "blues" (84), "pop" (100) and "celtic" (99). The annotations can be downloaded from their website[5]. They were originally used to automatically classify music pieces into genres based on their harmony (Pérez-Sancho et al., 2009). Due to limitations of the Band-in-a-Box file format the files were initially encoded in, each file is only annotated with a single global key. We therefore expect the models to be more noisy, because possible key changes are not annotated. On the other hand, this data set is substantially bigger than the two audio sets, and is therefore expected to lead to higher and more reliable counts.

Our last source of data is just a readily available set of bigram change probabilities from the *MySong* program (Simon et al., 2008). This is an application that generates chord accompaniments for a given melody, which requires the same kind of probabilities we need in order to produce likely chord sequences. These sequences are obtained from 298 so-called "lead-sheets", a type of sheet music that serves as a blueprint of a song, containing just the melody, lyrics and chord symbols. The annotated songs reflect a broad spectrum of popular music, including pop, rock, R&B, jazz, and country. The drawback of using these probabilities, which are available on-line[6], is that we do not have any control over the exact method of calculation. We can learn from Simon et al. (2008) that one of the notable differences to our approach is the fact that the key modes were annotated automatically based on the chord sequences and the key signature. This suggests that the granularity of the key annotations might be too rough, as most short key modulations are not marked by a change of key signature and neither are modulations between relative keys, which share the same key signature. Obviously, because the raw data is not available, we cannot calculate probabilities other than bigrams. These drawbacks notwithstanding, the MySong data will provide an interesting point of comparison.

We created a relative chord model for each of the data sets using the

[5] `http://grfia.dlsi.ua.es/cm/projects/prosemus/database.php`
[6] `http://research.microsoft.com/en-us/um/people/dan/chords/`

data set	unigrams major	unigrams minor	bigrams major	bigrams minor	trigrams major	trigrams minor
SEMA	985	1269	731	983	461	685
Isophonics	12862	2192	12216	2024	11338	1776
9GDB	34011	6026	32403	5459	30051	4728

Table 4.3: Number of retained n-gram tokens in the back-off model per data set

data set	unique unigrams major	unique unigrams minor	unique bigrams major	unique bigrams minor	unique trigrams major	unique trigrams minor
SEMA	30	29	49	50	57	68
Isophonics	43	28	234	111	563	207
9GDB	47	41	594	303	1775	620
maximum	48	48	2256	2256	106032	106032

Table 4.4: Number of unique n-gram types per data set

procedure described in section 4.3.1. To give an idea of the size of the data, the number of n-gram tokens in each set is displayed in table 4.3. These are the retained n-grams when using our chord vocabulary of 48 chords. They cover 95% or more of the chords in the data sets. It can be seen that the proportion of major to minor mode strongly varies between sets. In the SEMA set the majority (56%) of chords is in the minor mode, whereas in the Isophonics and 9GDB set the major mode is much more prevalent (85%). Because we only consider relative chord changes in one mode and because we let key changes be handled by a theoretical model, the undesired effect of skewing the models towards one mode will be avoided however. In table 4.4, the number of unique n-gram types per mode and model order are shown together with their theoretical maxima. This gives us an idea of the breadth of the data sets. There is obviously a clear correlation between the number of types and the number of available tokens. The data suggests that the SEMA is too small to be generalise well. The major Isophonics model and both 9GDB models cover almost all unigrams, but this is not a necessarily indicative of the model's quality. From music theory we know that some relative chords will indeed be very rare. Remember that the 9GDB data set is only annotated with a global key, so this almost complete relative chord coverage can be overrated due to the erroneous interpretation of a relative chord in the global key instead of its correct local key. The number of n-gram types increases at a much slower pace than the theoretical number of n-grams, indicating that sequences are not just chained randomly together. Finally, the most common bigrams and trigrams per mode are displayed in table 4.5 and in table 4.6, together with

SEMA	major mode Isophonics	9GDB
Vmaj-Imaj (9.01%)	Vmaj-Imaj (10.93%)	Vmaj-Imaj (13.65%)
IVmaj-Imaj (7.97%)	Imaj-IVmaj (10.55%)	Imaj-Vmaj (7.10%)
Imaj-IVmaj (6.35%)	IVmaj-Imaj (10.54%)	Imaj-IVmaj (6.01%)
Imaj-Vmaj (4.73%)	Imaj-Vmaj (5.38%)	IImin-Vmaj (5.62%)
Vmaj-IVmaj (4.04%)	IVmaj-Vmaj (4.44%)	IVmaj-Imaj (3.66%)
SEMA	**minor mode Isophonics**	**9GDB**
Vmaj-Imin (8.78%)	Vmaj-Imin (8.86%)	Vmaj-Imin (10.05%)
VIImaj-Imin (6.74%)	Imin-IVmin (6.60%)	Imin-Vmaj (5.34%)
Imin-VIImaj (6.48%)	IVmin-Imin (4.05%)	VIImaj-IIImaj (5.14%)
Imin-IVmin (4.79%)	Imin-VImaj (3.76%)	Imin-VIImaj (3.30%)
Imin-IIImaj (4.61%)	VImaj-Imin (3.23%)	VIImaj-Imin (3.30%)

Table 4.5: Common bigrams in three data sets

SEMA	major mode Isophonics	9GDB
Imaj-IVmaj-Imaj (4.30%)	Imaj-IVmaj-Imaj (6.42%)	Imaj-Vmaj-Imaj (5.61%)
IImin-Vmaj-Imaj (3.39%)	IVmaj-Imaj-IVmaj (4.66%)	Vmaj-Imaj-Vmaj (4.54%)
IVmaj-Imaj-IVmaj (2.73%)	Vmaj-Imaj-IVmaj (3.89%)	IImin-Vmaj-Imaj (4.38%)
Vmaj-IVmaj-Imaj (2.47%)	IVmaj-Vmaj-Imaj (3.42%)	Vmaj-Imaj-IVmaj (2.89%)
IVmaj-Vmaj-Imaj (2.47%)	Imaj-Vmaj-Imaj (2.84%)	Imaj-IVmaj-Imaj (2.66%)
SEMA	minor mode Isophonics	9GDB
VIImaj-Imin-VIImaj (4.03%)	Vmaj-Imin-IVmin (3.83%)	Imin-Vmaj-Imin (4.74%)
Imin-VIImaj-Imin (3.83%)	Imin-Vmaj-Imin (2.60%)	Vmaj-Imin-Vmaj (3.12%)
IVmin-Vmaj-Imin (3.24%)	Imin-IVmin-Imin (2.40%)	Imin-VIImaj-Imin (2.53%)
Imin-Vmaj-Imin (3.15%)	Imin-IVmin-Vmaj (2.20%)	VIImaj-Imin-VIImaj (2.36%)
Vmaj-Imin-Vmaj (2.36%)	Imin-VImaj-Imin (2.04%)	IIImaj-VIImaj-IIImaj (1.79%)

Table 4.6: Common trigrams in three data sets

the percentage of tokens they represent. There is a fair amount of consistency between the models, and the obtained n-grams are all very plausible according to music theory.

4.6 Quantifying model fitness: perplexity

4.6.1 Quantifying the benefits of context size

Later on, we will use chord change models to support automatic key and chord estimation and we will see what the influence is of the different models. However, there are already ways to compare these models on their own. The principal ideas behind the definition of sequence models, is that some sequences are more likely than others. In other words, the context reduces the uncertainty about a symbol after having seen the preceding symbols. This remaining uncertainty is conveniently expressed by the entropy H (in nats):

$$\begin{aligned} H(n,m) &= -\sum_{c'_1,\ldots,c'_{n-1},c'_n \in \mathbf{C}'} P\left(c'_1,\ldots,c'_{n-1},c'_n|m\right) \log P\left(c'_n|c'_1,\ldots,c'_{n-1},m\right) \\ &= -\sum_{c'_1} P\left(c'_1|m\right) \sum_{c'_2} P\left(c'_2|c'_1,m\right) \\ &\quad \ldots \sum_{c'_n} P\left(c'_n|c'_1,\ldots,c'_{n-1},m\right) \log P\left(c'_n|c'_1,\ldots,c'_{n-1},m\right) \end{aligned} \tag{4.27}$$

From the entropy we can derive the *perplexity of a model* as

$$PP_{\text{model}}(n,m) = e^{H(n,m)} \tag{4.28}$$

This measure represents the number of symbols that can follow the context under the assumption that these symbols were all equally likely. For example, a model with $N_c = 48$ and $PP_{model} = 9$ contains the same amount of information as a uniformly distributed model with $N_c = 9$. The case of a uniformly distributed model is also known as the 0-gram model, which has a perplexity that is equal to the number of possible chords N_c.

The reasoning behind the use of larger contexts to build our models, is that larger contexts will hopefully lead to less uncertainty and thus decrease the perplexity, but there is no guarantee that raising the order will give rise to a significant drop in perplexity. We calculated the perplexity of each of the chord change models derived from the data sets presented in section 4.5 and plotted them in figure 4.8. As can be seen in the picture, the perplexity decreases steadily with an increase in context size. This indicates that it is worth investigating if the use of higher order models also

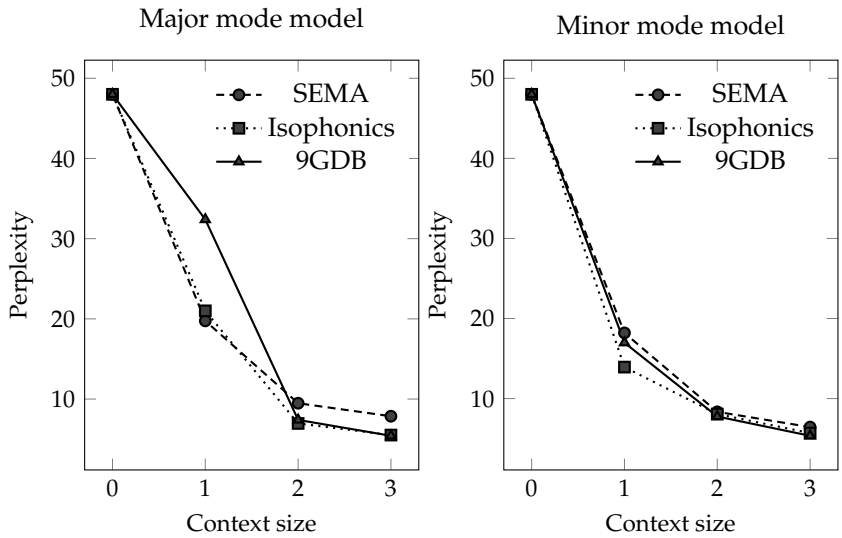


Figure 4.8: Model perplexities for the major and minor chord change models derived from three different data sets

improves the key and chord estimation from audio. The major models exhibit larger differences between each other than the minor models, which is likely due to the fact that the number of tokens available for the construction of the major models can differ up to an order of magnitude, whereas the number of tokens involved in the construction of the minor models is of the same order. We also notice that the rate of decrease in perplexity slows down quickly, which shows that the expected improvement will converge to a point after which an increase in context size will no longer change the results.

4.6.2 Quantifying model specificity

Another characteristic of our relative chord models, is that a model built on one set may not well represent the data from another set, even though the Kneser-Ney smoothing tries to mitigate this lack of generalisation somewhat. This gives rise to the eternal adage of "generality versus specificity": the less internal variation there is within a data set, the better it can be modelled (leading to a more useful model) at the cost of being less applicable to significantly different new data. It is therefore interesting to measure whether a model is more generally applicable, but of limited power, or more specific and of greater power. For this, we can use the *test set perplexity*. Suppose that we have a we have a data set $\underline{\mathbf{S}} = \left\{\underline{\mathbf{s}}_1, \underline{\mathbf{s}}_2, \ldots, \underline{\mathbf{s}}_{\mathbf{N_s}}\right\}$

	n					
	1	2	3	P_u	L	D
SEMA (leave-one-out)	11.99	11.00	10.62	0.09	3	3
Isophonics (20-fold)	11.70	7.69	7.83	0.03	2	2
9GDB (5-fold)	16.34	7.91	7.11	0.01	1	1

Table 4.7: Mean validation set perplexities per data set and model order and corresponding free parameter values

of N_s music pieces and that $\underline{\mathbf{s}}_\mathbf{y}$ is characterised by the key-chord sequence $(k_1, c_1), (k_2, c_2), \ldots, \left(k_{|\underline{\mathbf{s}}_\mathbf{y}|}, c_{|\underline{\mathbf{s}}_\mathbf{y}|}\right)$ of length $\left|\underline{\mathbf{s}}_\mathbf{y}\right|$. The likelihood of a piece $\underline{\mathbf{s}}_\mathbf{y}$ as generated by the n-gram model defined by the probabilities

$$P\left(c'_j | c'_{j-1}, \ldots, c'_{j-n+1}, m_{j-1}\right)$$

is then

$$\begin{aligned} P\left(\underline{\mathbf{s}}_\mathbf{y}\right) &= \prod_{j=1}^{|\underline{\mathbf{s}}_\mathbf{y}|} P\left(c'_j | c'_{j-1}, \ldots, c'_{j-n+1}, m_{j-1}\right) \\ &= \exp\left(\sum_{j=1}^{|\underline{\mathbf{s}}_\mathbf{y}|} \log P\left(c'_j | c'_{j-1}, \ldots, c'_{j-n+1}, m_{j-1}\right)\right) \end{aligned} \tag{4.29}$$

From the probabilities $P\left(\underline{\mathbf{s}}_\mathbf{y}\right)$, the test set perplexity of the entire data set $PP_{test}\left(\underline{\mathbf{S}}; n\right)$ can be calculated as as the reciprocal of the geometric mean of the likelihoods of all music pieces in the data set:

$$\begin{aligned} PP_{test}\left(\underline{\mathbf{S}}; n\right) &= \left(\prod_{y=1}^{N_s} P\left(\underline{\mathbf{s}}_\mathbf{y}\right)\right)^{-\frac{1}{\sum_{y=1}^{N_s} |\underline{\mathbf{s}}_\mathbf{y}|}} \\ &= \exp\left(-\frac{\sum_{y=1}^{N_s} \sum_{j=1}^{|\underline{\mathbf{s}}_\mathbf{y}|} \log P\left(c'_j | c'_{j-1}, \ldots, c'_{j-n+1}, m_{j-1}\right)}{\sum_{y=1}^{N_s} \left|\underline{\mathbf{s}}_\mathbf{y}\right|}\right) \end{aligned} \tag{4.30}$$

This measure[7] has also been used by Scholz et al. (2009) to demonstrate that sequences of relative chords in their keys are more informative than sequences of absolute chords and by Rohrmeier and Graepel (2012) to compare the predictive power of n-grams to alternative types of harmonic modelling. We first used the test set perplexity to optimise the free parameters

[7] although expressed in bits instead of nats

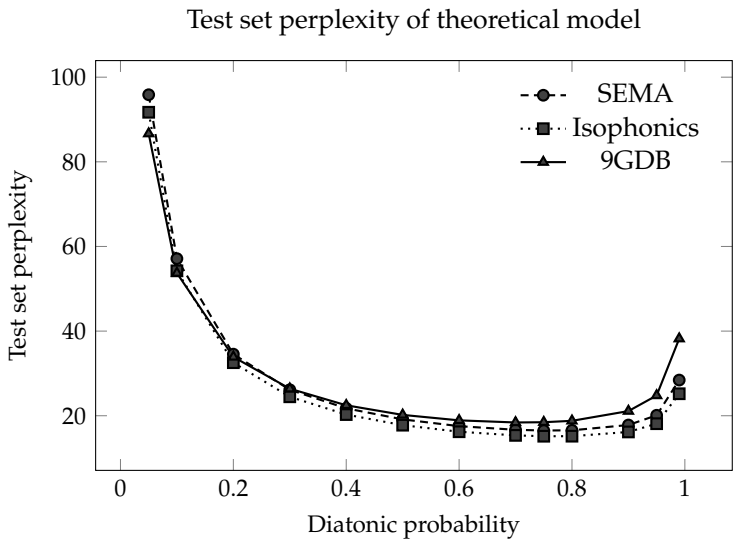


Figure 4.9: Test set perplexity of the theoretical model in function of the probability of a diatonic chord

of the smoothing procedure (pruning limit L, discount D and the probability of unseen unigrams P_u). We performed a multi-fold cross-validation per data set with a grid search over a number of parameter combinations in order to obtain the combination with the lowest mean trigram perplexity. The mean validation set perplexities per data set and model order are displayed in table 4.7 together with the corresponding smoothing parameters. As expected, the optimal value for P_u is proportional to the number of unseen unigrams, but in general, the test set perplexities are not very sensitive to changes in these parameters. For the theoretical model, we have examined how the test set perplexity is affected by the probability P_d of going to a diatonic chord. A plot of the test set perplexities for all three data sets in function of P_d can be seen in figure 4.9. Its optimal value consistently lies around 0.75, which is about the proportion of diatonic chords measured in the data sets.

We then calculated the test set perplexity results for all combinations of models and test sets, where the models are again constructued using the entire data set. The results can be found in table 4.8. A first conclusion that can be drawn from the data is that the lowest perplexities arise when the test set matches the model learning set, despite the usage of Kneser-Ney smoothing. When we compare bigram to trigram test set perplexities, we see that the decline present in the model perplexity (calculated with the same smoothing parameters) is only reflected in the cases where the test set matches the model. Based on the validation set perplexities however,

	test set		
bigram model	SEMA	Isophonics	9GDB
SEMA	7.45	10.14	13.21
Isophonics	10.65	6.25	11.16
9GDB	10.81	8.56	7.20
MySong	12.31	8.93	10.94
Theoretical	16.51	15.16	18.47

(a) bigram models

	test set		
trigram model	SEMA	Isophonics	9GDB
SEMA	6.52	10.43	13.45
Isophonics	11.74	4.90	11.43
9GDB	10.59	8.71	5.20

(b) trigram models

Table 4.8: Test set perplexities for all combinations of models and test sets

which show little to no decline between bigrams and trigrams, we can deduce that the decline is due to overfitting. When the models are applied to different test sets, the perplexities for bigram and trigram models are close to each other. We conclude that the larger context adds information that is mostly specific to that data set. The theoretical model clearly lags behind in performance compared to the models based on data. A possible explanation could be that our assumption that chord sequences are composed of perceptually close chord pairs does not always lead to realistic sequences. In addition, the probabilities for non-diatonic transitions might not be distinctive enough, because they are not covered by Lerdahl's theory. This causes roughly 25% of the relative chord changes to be uniformly distributed . An indication that this large unaccounted fraction effectively proves problematic, is the fact that the theoretical model has its lowest perplexity in combination with the Isophonics set, which we know is the most compliant with classical harmony theory and therefore contains the most diatonic transitions. Of the corpus-based models, the SEMA model performs consistently worse when applied to a different test set, confirming our hypothesis that the set is too small to generalise well.

4.7 Conclusion

In this chapter, the modelling of musicological knowledge in the form of key and chord sequence probabilities has been discussed. We first presented the general ideas and motivation behind our proposed models. Then we gave detailed procedures to create these models based on annotated data sets or music theory. Finally, we quantified the resulting models in terms of size and specificity. This allowed us to compare multiple models coming from different sources of knowledge and to draw some conclusions concerning their applicability. In the next chapters, we will embed the models in a system for the automatic estimation of keys and chords from audio. We will then see if the mutual relations between the models of different origin are also reflected in the results of the estimation procedure.

5 A probabilistic system for estimating chords and local keys

In this chapter, a system for the automatic estimation of keys and chords from audio will be presented. We will use the bigram context model presented in the previous chapter as one component, but we will also introduce the other components of the system. The resulting system will be subjected to a series of experiments in order to examine its accuracy and to assess the effects of different context models on this accuracy. The final system will then serve as a base-line that will be further enhanced in the next chapters.

5.1 Architecture of the system

The proposed approach can be classified as a knowledge-based simultaneous local key and chord estimation system. It is based on the framework of a hidden Markov model (HMM), where the different aspects of the system – acoustic, duration and context – are controlled by musicological knowledge, as opposed to optimising them on a data set. This implies that only a few free parameters are needed to balance the different aspects and that the majority of the parameters have a clear and directly interpretable musicological meaning. As outlined in chapter 3, the system can be divided into separate feature extraction and processing phases.

5.1.1 Feature extraction

An input audio file is converted to a chroma-based feature stream in three steps. In the first stage, the waveform is resampled to 8 kHz and converted to mono. The resulting waveform is then subjected to a spectral analysis with the following characteristics: the analysis is performed on 150 ms long fragments (called *frames*), the frames are weighted with a Hamming window and for each windowed frame, a chroma vector is calculated. The *analysis shift* T_a, defined as the time between subsequent frames, is 20 ms. Finally, the local spectral analysis is followed by a smoothing of the chromas.

CHROMA VECTOR ANALYSIS As in most other systems, our acoustic observation vectors are chroma vectors. In its simplest form, a chroma vector is just a log-frequency representation of the spectral content, folded into a single octave. The problem with such a representation is that e.g. the third harmonic of a pitch folds into a chroma that is located at $+7$ or -5 semitones with respect to the fundamental. Consequently, it will add evidence to a second pitch class that is not necessarily present as a played note in the signal. In chapter 3, we listed a number of approaches to mitigate this problem. To recapitulate, the higher harmonics can be cleared from the spectrum by assigning them to a fundamental frequency, or they can be left intact and be accounted for in the later feature processing phase. We chose to deal with the higher harmonics in the feature extraction phase and to use the multi-pitch tracking approach of Varewyck et al. (2008), developed at our lab. It is publicly available through the MARSYAS audio software framework (Tzanetakis and Cook, 2000). We configure it such that a fundamental frequency should be supported by at least one harmonic. Given that the sampling frequency is reduced to 8 kHz, the highest detectable fundamental frequency is therefore equal to 2000 Hz. To reduce the sensitivity to the sound volume, the elements of non-zero chroma vectors are rescaled by dividing them by the L_1 norm, defined as $\|\mathbf{x}\|_1 = \Sigma_i |x_i|$, such that they add up to one.

SMOOTHING CHROMA VECTORS A smoothing of the observations is achieved by calculating the mean chroma vectors over a number of successive frames. This can in principle be performed in two ways: either by taking the mean over fixed length frame sequences, or by taking the mean over variable length frame sequences that are presumed to reveal an interesting segmental context (e.g. intervals between subsequent beats). We only consider the fixed length smoothing approach because it is not burdened by errors made by the beat-tracking algorithm.

The smoothed chroma vectors are calculated by letting a *smoothing window* slide over the observations and by taking the mean over all frames whose centre falls inside the smoothing window. We call the number of

frames in the window the *smoothing size* (S). The distance between successive smoothing windows gives rise to the final rate of the observations that will be used in the feature processing phase later on. This time shift between subsequent smoothed frames is hereafter called the *observation shift*, and denoted as T_x. It is either equal to the analysis shift, or to a multiple s thereof: $T_x = sT_a$. The observation shift is limited by the smoothing size S: $s > S$ would imply that some frames are not used in the smoothing, resulting in an unnecessary loss of information, so $s = \{1, 2, \ldots, S\}$. By varying the observation shift T_x, we can trade off computational load for time resolution, and possibly accuracy. If the observation shift is smaller than the smoothing size, the resulting smoothed chroma vectors are overlapping in time. In that case, we turn them into disjoint segments by keeping only sections of observation shift length, centred around the middle. The outcome forms the acoustic observation vector sequence $\mathbf{X} = \{\mathbf{x}_1, .., \mathbf{x}_N\}$ that will be fed into the feature processing phase. A diagram depicting the smoothing for the two extreme cases of the observation shift ($s = 1$ and $s = S$) can be seen in figure 5.1.

Smoothed chroma vectors whose power falls below a certain threshold are not fed into the probabilistic framework, but immediately get the no-chord and no-key assigned. In practice, this means that the silence at the beginning and end of a music piece is not processed further.

5.1.2 Probabilistic framework

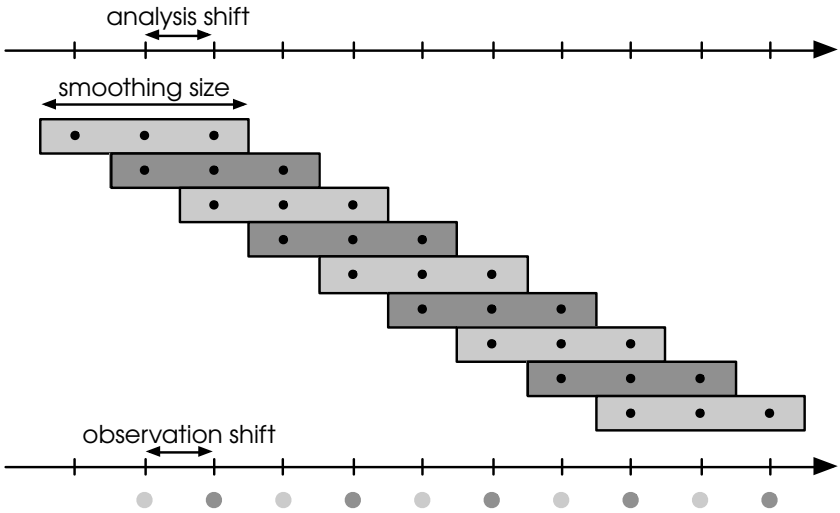
The feature processing phase implements a unified probabilistic framework for the simultaneous recognition of chords and keys. Its objective is to retrieve the most likely sequence of states $\hat{Q} = \{\hat{q}_1, \ldots, \hat{q}_{N_x}\}$ for the acoustic observation sequence $\mathbf{X}$ of length N_x. Each state q consists of the combination of a key k and a chord c, and the state sequence is equivalent to a sequence of (key, chord) pairs: in what follows, q_n is equivalent to (k_n, c_n)[1]. Using Bayes' rule, it follows that

$$\begin{aligned} \hat{Q} &= \arg\max_{Q} P(Q|\mathbf{X}) \\ &= \arg\max_{Q} P(Q)P(\mathbf{X}|Q) \end{aligned} \tag{5.1}$$

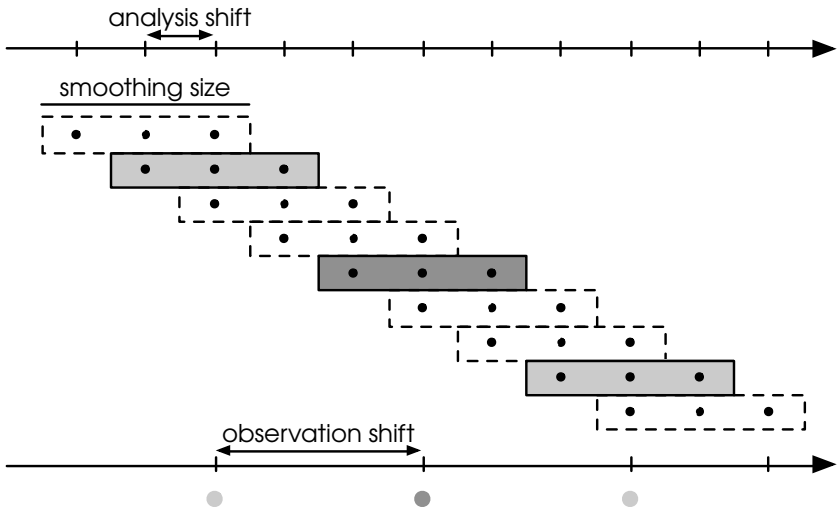
The acoustic likelihood $P(\mathbf{X}|Q)$ gets further factorised by making the standard assumption that acoustic observations emitted from a state are independent of each other:

$$P(\mathbf{X}|Q) = \prod_{n=1}^{N_x} P(\mathbf{x}_n|k_n, c_n) \tag{5.2}$$

[1]Note that the letter n is used in this and subsequent chapters as an index of time steps. There is no relation with the use of n for generalised n-grams of the last chapter. We only use bigram models here.



(a) sliding smoothing



(b) disjoint smoothing

Figure 5.1: Schematic of the smoothing procedure. The black circles represents the chroma vectors calculated on all frames, and alternating grey circles the resulting smoothed chroma vectors that are used as observations in the probabilistic framework. The smoothing windows used for the computation of the latter are indicated by boxes, here for $S = 3$, where only the shaded boxes need to be computed. The upper pane shows the situation when the observation shift is equal to the analysis shift ($T_x = T_a$), the lower when the observation shift is equal to S times the analysis shift ($T_x = ST_a$).

Because of the temporal dependencies in music, which get even amplified by the smoothing in the feature extraction, this assumption is clearly an oversimplification, although common. In speech processing, this simplification is often compensated by the introduction of delta-features, which take the differences between subsequent observations into account. However, while it is relatively straightforward to derive acoustic models for chroma features from music theory, there is no obvious link between delta-features and music theory. Therefore we do not use them in our knowledge-based system. Accounting for the temporal dependency between observations will be taken care of by the prior musicological model, which will be discussed later.

We further simplify the model by assuming that keys and chords can be independently tested for an accordance between observation and label:

$$P\left(\mathbf{X}|Q\right) = \prod_{n=1}^{N_x} \underbrace{P\left(\mathbf{x_n}|k_n\right)^{\alpha_k}}_{\text{key acoustic model}} \underbrace{P\left(\mathbf{x_n}|c_n\right)^{\alpha_c}}_{\text{chord acoustic model}} \tag{5.3}$$

These two terms will respectively be calculated by a key and a chord acoustic model. We also added balance parameters α_k and α_c so that we can later control the contribution ratio of the two acoustic models in the final system.

On the other hand, the prior probability $P(Q)$ is computed by making the first order Markov assumption. This means that

$$P(Q) = \prod_{n=1}^{N_x} P\left(q_n|q_{n-1}\right) \tag{5.4}$$

It is here that we would like to use the bigram musicological knowledge models proposed in chapter 4. The major difference between the construction of the models there and in the current case, is that we no longer dispose of a chord-wise segmentation. We have to follow the fixed, blind observation rate determined by the feature extraction phase instead. This means that a change of state q now no longer implies a chord change. We keep the requirement that key changes can only occur together with chord changes, such that any movement between consecutive observations falls into one of three cases:

- the chord and key stay the same
- the chord changes, but the key stays the same
- both chord and key change

The transition probability related to each of these cases gets calculated by the combination of four submodels: a chord change model, a key change model, a chord duration model and a key duration model, where the change

models are those of the previous chapter. Not all submodels are used in all cases though, the change models are obviously only active when their corresponding change takes place. The main rationale for this factorisation into smaller models is that in this way, the different submodels can be manipulated on their own and the prior knowledge they encode can come from different sources. Furthermore, because fewer variables are involved in the combination of these models compared to the original monolithic model, there are considerably fewer parameters in the model.

Formally, the decomposition of the prior probability into the different cases goes as follows. We distinguish two types of state transitions: self-transitions ($q_n = q_{n-1}$) and transitions to another state ($q_n \neq q_{n-1}$). By applying the law of total probability, we get the following expression:

$$P\left(q_n | q_{n-1}\right) = \begin{cases} P\left(q_n = q_{n-1} | q_{n-1}\right) & \text{if } q_n = q_{n-1} \\ P\left(q_n \neq q_{n-1} | q_{n-1}\right) P\left(q_n | q_{n-1}, q_n \neq q_{n-1}\right) & \text{if } q_n \neq q_{n-1} \end{cases} \tag{5.5}$$

Because we jointly estimate keys and chords, these conditions can also be written as $q_n = q_{n-1} \iff k_n = k_{n-1} \wedge c_n = c_{n-1}$ and $q_n \neq q_{n-1} \iff k_n \neq k_{n-1} \vee c_n \neq c_{n-1}$. To simplify the model even more, we impose that key changes are only allowed to co-occur with chord changes. Consequently, the condition $q_n \neq q_{n-1}$ can actually be narrowed down to $c_n \neq c_{n-1}$. Formulating the probabilities in terms of keys and chords instead of state variables q, then gives us

$$P\left(k_n, c_n | k_{n-1}, c_{n-1}\right) = \begin{cases} \underbrace{P\left(k_n = k_{n-1}, c_n = c_{n-1} | k_{n-1}, c_{n-1}\right)}_{\text{self-transition probabilities}} \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{if } k_n = k_{n-1} \wedge c_n = c_{n-1} \\ \underbrace{P\left(c_n \neq c_{n-1} | k_{n-1}, c_{n-1}\right) P\left(k_n, c_n | k_{n-1}, c_{n-1}, c_n \neq c_{n-1}\right)}_{\text{state-changing probabilities}} \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{if } c_n \neq c_{n-1} \end{cases} \tag{5.6}$$

The self-transition probabilities follow from a state duration model. It accounts for the fact that keys and chords are likely to last longer than one observation shift. We decompose it further, such that

$$P\left(k_n = k_{n-1}, c_n = c_{n-1} | k_{n-1}, c_{n-1}\right) = \underbrace{P\left(c_n = c_{n-1} | k_{n-1}, c_{n-1}\right)}_{\text{chord duration probabilities}} \underbrace{P\left(k_n = k_{n-1} | k_{n-1}, c_{n-1}, c_n = c_{n-1}\right)}_{=1} \tag{5.7}$$

Because keys can only change together with chords, the chord equality in the conditional part of $P\left(k_n = k_{n-1} | k_{n-1}, c_{n-1}, c_n = c_{n-1}\right)$ also implies a

key equality, so the whole term reduces to 1. The co-occurrence requirement therefore also implies that the state duration model that calculates $P(q_n = q_{n-1}|q_{n-1})$ is actually a chord duration model that calculates the *chord duration probabilities* $P(c_n = c_{n-1}|k_{n-1}, c_{n-1})$.

To model the transitions that change state, we start with the same decomposition as for the self-transitions:

$$\begin{aligned} &P(c_n \neq c_{n-1}|k_{n-1}, c_{n-1})\, P(k_n, c_n|k_{n-1}, c_{n-1}, c_n \neq c_{n-1}) = \\ &\qquad \left(1 - \underbrace{P(c_n = c_{n-1}|k_{n-1}, c_{n-1})}_{\text{chord duration probabilities}}\right) \\ &\qquad \underbrace{P(c_n|k_{n-1}, c_{n-1}, c_n \neq c_{n-1})}_{\text{chord change probabilities}} \underbrace{P(k_n|k_{n-1}, c_n, c_{n-1}, c_n \neq c_{n-1})}_{\text{key transition probabilities}} \end{aligned} \tag{5.8}$$

In this formulation, we can recognise the bigram *chord change probabilities* of chapter 4 in the term $P(c_n|k_{n-1}, c_{n-1}, c_n \neq c_{n-1})$. Only those indices n that follow a chord change take part in this term, which makes it equivalent to the term $P(c_j|k_{j-1}, c_{j-1})$ of the previous chapter where the successive indices j are by definition increasing with chord changes.

We have also encountered the key transition probabilities in the previous chapter. Just like there, we use the law of total probability to decompose it into separate *key duration probabilities* and *key change probabilities*, giving us:

$$\begin{aligned} &P(k_n|k_{n-1}, c_n, c_{n-1}, c_n \neq c_{n-1}) = \\ &\qquad \begin{cases} \underbrace{P(k_n = k_{n-1}|k_{n-1}, c_n, c_{n-1}, c_n \neq c_{n-1})}_{\text{key duration probabilities}} \\ \qquad\qquad\qquad\qquad \text{if } k_n = k_{n-1} \wedge c_n \neq c_{n-1} \\ \left(1 - \underbrace{P(k_n = k_{n-1}|k_{n-1}, c_n, c_{n-1}, c_n \neq c_{n-1})}_{\text{key duration probabilities}}\right) \\ \quad \underbrace{P(k_n|k_{n-1}, c_n, c_{n-1}, c_n \neq c_{n-1}, k_n \neq k_{n-1})}_{\text{key change probabilities}} \\ \qquad\qquad\qquad\qquad \text{if } k_n \neq k_{n-1} \wedge c_n \neq c_{n-1} \end{cases} \end{aligned} \tag{5.9}$$

Because the key duration probabilities contain the requirement $c_n \neq c_{n-1}$ in the conditional part, key duration is not expressed as a number of time steps n, unlike chord duration. Instead it is expressed as a number of chord changes, which is musicologically more intuitive.

To summarise, we have factorised the prior probability into four terms, each with a distinct, musicologically interpretable role. We introduce the

following short notation for them:

$$P_{cd} = P\left(c_n = c_{n-1} | k_{n-1}, c_{n-1}\right) \tag{5.10}$$
$$P_{cc2} = P\left(c_n | k_{n-1}, c_{n-1}, c_n \neq c_{n-1}\right) \tag{5.11}$$
$$P_{kd} = P\left(k_n = k_{n-1} | k_{n-1}, c_n, c_{n-1}, c_n \neq c_{n-1}\right) \tag{5.12}$$
$$P_{kc2} = P\left(k_n | k_{n-1}, c_n, c_{n-1}, c_n \neq c_{n-1}, k_n \neq k_{n-1}\right) \tag{5.13}$$

We will use multiple alternatives for the models that calculate these probabilities. Our proposals for the bigram key and chord change models have already been presented, so we only need to elaborate the key duration and chord duration models. In practice, the models work together to calculate the prior model probability in a way that can be broken down into the previously introduced three cases (plus the disallowed combination of a key change without chord change):

$$P\left(k_n, c_n | k_{n-1}, c_{n-1}\right) = \begin{cases} 0 & \text{if } k_n \neq k_{n-1} \wedge c_n = c_{n-1} \\ P_{cd} & \text{if } k_n = k_{n-1} \wedge c_n = c_{n-1} \\ \left(1 - P_{cd}\right) P_{cc2}^{\beta_c} P_{kd} & \text{if } k_n = k_{n-1} \wedge c_n \neq c_{n-1} \\ \left(1 - P_{cd}\right) P_{cc2}^{\beta_c} \left(1 - P_{kd}\right) P_{kc2}^{\beta_k} & \text{if } k_n \neq k_{n-1} \wedge c_n \neq c_{n-1} \end{cases} \tag{5.14}$$

We have introduced two additional parameters β_c and β_k to have more control over the relative importances of the key and chord change models. If set to zero, all change probabilities will be the same, which comes down to turning the change models off completely. Increasing the value of the exponents therefore allows us to gradually raise the influence of the change models. However, the normalisations of equation (4.23) and equation (4.26) need to be reapplied after this exponential weighing in order to conserve the total probability.

The final decoding of the state sequence that optimally explains the observation sequence under the musicological model constraints is performed according to the Viterbi (1967) algorithm. In order to prevent numerical underflow and to reduce the computational requirements, we work in the logarithmic domain. For our key-chord states, this means that the main Viterbi iteration to calculate the intermediary forward variable δ_n, defined as the probability of the best path that ends in (K_2, C_2) at time n, takes the following form:

	$P1$	$m2$	$M2$	$m3$	$M3$	$P4$	$d5$	$P5$	$A5$	$M6$	$m7$	$M7$
maj	1	0	0	0	1	0	0	1	0	0	0	0
min	1	0	0	1	0	0	0	1	0	0	0	0
dim	1	0	0	1	0	0	1	0	0	0	0	0
aug	1	0	0	0	1	0	0	0	1	0	0	0

Table 5.1: Binary templates for four chord types

$$\begin{aligned}
&\forall K_2 \in \mathbf{K}, \forall C_2 \in \mathbf{C}: \log_{10} \delta_n \left(k_n = K_2, c_n = C_2\right) = \\
&\quad \log_{10} P\left(\mathbf{x_n} | k_n = K_2, c_n = C_2\right) \\
&\quad + \max_{\substack{K_1 \in \mathbf{K} \\ C_1 \in \mathbf{C}}} \Big\{ \log_{10} \delta_{n-1} \left(k_{n-1} = K_1, c_n = C_1\right) \\
&\qquad + \log_{10} P\left(k_n = K_2, c_n = C_2 | k_{n-1} = K_1, c_{n-1} = C_1\right) \Big\}
\end{aligned} \tag{5.15}$$

The optimal key and chord sequences $\hat{K}$ and $\hat{C}$ for the observation sequence $\mathbf{X}$ can then be retrieved as $\hat{K}, \hat{C} = \arg\max \delta_{N_x}(k_{N_x}, c_{N_x})$.

Because of the requirement that key changes should co-occur with chord changes, the resulting system is reminiscent of a hierarchical hidden Markov model, as proposed by Fine et al. (1998). Keys would then be a first layer in the hierarchy that can emit chord sequences in a second layer. However, all hierarchical HMMs can be written as regular HMMs and the advantage of the former, namely that the hierarchy can be taken into account when training the system with a Baum-Welch procedure, is irrelevant here because we only decode a manually fixed HMM. Therefore we chose to formulate our system in the simpler framework of a regular HMM. In the following sections, we will have a more detailed look at the acoustic and duration models and the different options for calculating them.

5.1.2.1 Acoustic models

The chord acoustic model calculates the likelihood of an acoustic observation vector given a proposed chord. We consider two simple approaches which both rely on the definition of a 12-dimensional binary template $\mathbf{t}(c)$ to represent chord c. The v-th element $t_v(c)$ of $\mathbf{t}(c)$ is 1 if chord c implies a component of chroma v, and 0 in the other case. The 48 chord templates needed for our chord vocabulary arise as all the possible rotations of the four chord type templates depicted in table 5.1. Since we limit the estimation to triads, all considered chords will have a template with three non-zero values.

The first approach is to consider the cosine similarity (or scaled inner product) between the observation vector and the chord template as an acoustic posterior probability.

$$P\left(c_n|\mathbf{x_n}\right) = \frac{\langle \mathbf{t}\left(c_n\right), \mathbf{x_n}\rangle}{\|\mathbf{t}\left(c_n\right)\|_2 \, \|\mathbf{x_n}\|_2} \tag{5.16}$$

Since the elements of the observation vectors and the chord templates are positive numbers, the cosine similarity ranges from 0 (no evidence found for any of the chromas present in the chord) to 1 (the observation vector only has contributions on the chromas present in the chord). In order to convert this posterior probability to a likelihood, it technically needs to be scaled by the mean posterior probability in the data set, $P_{mpp} = \overline{P\left(c_n|\mathbf{x_n}\right)}$:

$$P\left(\mathbf{x_n}|c_n\right) \propto \frac{P\left(c_n|\mathbf{x_n}\right)}{P_{mpp}} \tag{5.17}$$

This introduces a dependency on the data set, however, which we want to avoid in our acoustic models. Since measuring the P_{mpp}s revealed that their values are relatively close to each other anyway, we opted to fix P_{mpp} to 1.

The second approach is the one adopted by Catteau et al. (2007). It assumes that the elements of the observation vector can be considered independent of each other, and that the acoustic likelihood can thus be factorised as

$$P(\mathbf{x_n}|c_n) = \prod_{v=1}^{12} P(x_{nv}|t_v(c_n)) \tag{5.18}$$

Since $t_v(c)$ is either 1 or 0, there are only two probability distributions to model. A simple model is the following

$$P(x|0) = \nu_0 \left(P_0 + (1 - P_0)\exp\left(-\frac{x^2}{2\sigma^2}\right)\right) \tag{5.19}$$

$$P(x|1) = \begin{cases} \nu_1 \left(P_0 + (1 - P_0)\exp\left(-\frac{(x-\overline{x})^2}{2\sigma^2}\right)\right) & x \in (0, \overline{x}) \\ \nu_1 & x \in (\overline{x}, 1) \end{cases} \tag{5.20}$$

The quantities ν_0 and ν_1 are normalisation factors whereas P_0 is an offset which preserves some probability density at $x = 0$ or $x = 1$ for a template value of 1 and 0 respectively. Since we assume triads, and since the observation vectors are normalised, it follows that $\overline{x} = 0.33$ is an appropriate choice. In order to obtain sufficiently discriminating distributions, we chose $\sigma = 0.13$ (sufficiently smaller than $\overline{x}$). The parameter P_0 was set to 0.05. We will refer to this model as the (half-)Gaussian model.

The key acoustic model $P\left(\mathbf{x_n}|k_n\right)$ is calculated very similarly to the chord acoustic model. Here we use the non-binary templates defined by

	$P1$	$m2$	$M2$	$m3$	$M3$	$P4$	T	$P5$	$m6$	$M6$	$m7$	$M7$
major	5	2	3.5	2	4.5	4	2	4.5	2	3.5	1.5	4
minor	5	2	3.5	4.5	2	4	2	4.5	3.5	2	1.5	4

Table 5.2: Temperley templates for four modes

Temperley. These are vectors representing the stability of the 12 pitch classes relative to a given key. They are based on the Krumhansl profiles, but specifically adjusted for computational key-finding (Temperley, 1999). The templates for the two modes that are rotated to form the 24 key templates we need are shown in table 5.2. We use them in combination with the cosine similarity in the same way as we did for the chord acoustic model.

5.1.2.2 Duration models

As a first option for the chord duration model, we consider a general geometric model, obtained by using a single value for all chords:

$$P\left(c_n = c_{n-1} | k_{n-1}, c_{n-1}\right) = P_c$$

We can relate this chord self-transition probability P_c to the mean chord duration $\overline{d_c}$, which is expressed in seconds. By setting the mean chord duration expressed in observation time steps (obtained by dividing the mean chord duration by the observation shift T_x) equal to the mean of the geometric distribution, one obtains

$$\frac{\overline{d_c}}{T_x} = \frac{1}{1 - P_c} \iff P_c = 1 - \frac{T_x}{\overline{d_c}} \tag{5.21}$$

This mean duration can for example be measured in an annotated data set. In the SEMA set, $\overline{d_c} = 1.71$. We round this up to 1.76, such that $P_c = 0.875$ for disjoint smoothing or $P_c = 0.989$ for sliding smoothing.

The key duration model is calculated similarly, also by setting a single value for all keys:

$$P\left(k_n = k_{n-1} | k_{n-1}, c_n, c_{n-1}, c_n \neq c_{n-1}\right) = P_k$$

The difference with the chord duration model is that the key duration model is only active when a chord changes, so the time scale we are working on is defined by the chord changes. Therefore the mean of the geometric distribution can be set to the average number of chords per key $\overline{d_k}$.

$$\overline{d_k} = \frac{1}{1 - P_k} \iff P_k = 1 - \frac{1}{\overline{d_k}} \tag{5.22}$$

Because the number of key changes per song is generally a lot less than the number of chord changes, we do not think it is advisable to measure a mean key duration $\overline{d_k}$ on a data set. This might not be representative for unseen data, given the current amount of annotated data available. Therefore we fix this value to 8, leading to $P_k = 0.875$, such that it is not very difficult to change key.

5.2 Experimental evaluation

The experimental evaluation of our system first of all aims to determine optimal values for the free balance parameters $(\alpha_k, \alpha_c, \beta_k, \beta_c)$. Furthermore, we will evaluate different configurations of our system in order to get a better insight into the impact of changes to the individual components. Therefore, we will use the label-synchronised audio of the SEMA and Isophonics data sets presented in section 4.5 together with some evaluation measures. In general, we will use the SEMA set to search for good balance parameters and then apply the resulting system configurations to the Isophonics set to test the generalisation capabilities of the system.

5.2.1 Evaluation measures

The performance of our system is measured as the percentage of time the extracted key or chord equals the annotated key or chord. What we mean by being "equal to the annotation", is that the two should have the same tonic or root and the same mode or chord type. Their pitch spelling may differ however. Estimating related keys or chords will therefore not add to the score. We restrict the chord evaluation to segments where one of the basic triads (maj–min–dim–aug) has been annotated. This is so in 62.55% of the data for the SEMA set and 77.44% of the data for the MIREX set. By doing so instead of mapping the complex chord annotations to triads as we did in the chord change model derivation, we will get results that are clearer to analyse. Besides the triad starting on the root, other triads can be found in most complex chords. So if a complex chord annotation would get reduced to the single triad on its root, it is impossible to distinguish between an incorrect estimation that is somewhat understandable because all of its chromas are present in the audio signal (e.g. an *E*min estimation for a *C*maj7), and one that is plainly wrong because the signal contains just the triad. We therefore prefer an increase in meaningfulness of the evaluation over an increase in amount of evaluation data. Key estimation performance on the other hand, is measured over the whole data set.

In practice, the evaluation score is calculated by taking the union of both annotated and estimated label boundaries, and by then comparing

the annotated with the estimated label for all of the resulting segment pairs (limited to those annotated with a basic triad for chord evaluation). The binary score (1 for equal labels, 0 otherwise) for a segment is then multiplied by its duration and these weighted scores are summed together. Finally, this sum is divided by the total duration of all segments withheld for evaluation.

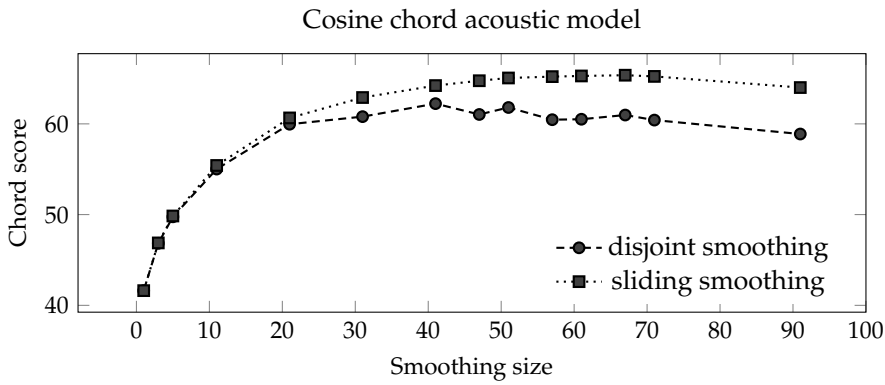
When comparing different configurations of the same system, we also check whether the differences in the evaluation results are statistically significant, or if these differences can just be attributed to statistical noise. Because the experiments are carried out over the same data sets for all configurations, we can use paired difference tests to this end. For chord estimation results, we employ Wilcoxon (1945)'s signed rank test, which operates under the assumption that both sets of results are distributed symmetrically around their median. This is the case for most chord outputs: the test results display some variance due to the range of inherent complexity of the different songs, but there is no underlying process that causes a significant skew towards either direction. This is not the case for key estimation results however. Because key labels extend over a long period of time, especially when no key changes happen in a song, we expect our system in the majority of cases to estimate the key either correctly or incorrectly over the whole duration of the music piece. This leads to two clusters of results around 0% and 100%, and a significant asymmetry if the median result deviates from 50%. Therefore we use the sign test when the results are asymmetrically distributed, because it does not make the symmetry assumption, though at the expense of a lower statistical power.

5.2.2 Impact of the feature extraction

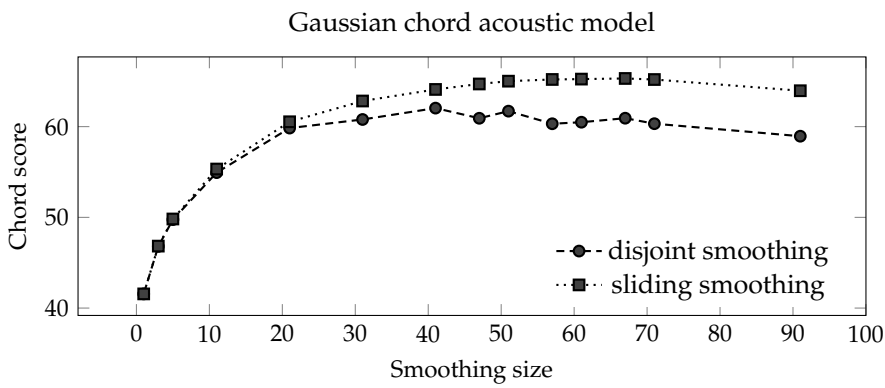
In a first experiment, we evaluate the chord acoustic model on its own and illustrate the influence of the feature extraction phase in case no explicit duration model is present. Therefore we use a simple frame-wise system that returns the most likely chord label (calculated independently from the key) for each feature vector, without any duration or change models. This means that we optimise the probability

$$P\left(K, C|\mathbf{X}\right) = \prod_{n=1}^{N_x} P\left(\mathbf{x_n}|k_n\right) P\left(\mathbf{x_n}|c_n\right) \tag{5.23}$$

We investigate the Gaussian and the cosine similarity chord acoustic models outlined in section 5.1.2.1 and alter only the smoothing step of the feature extraction. We combine a number of smoothing sizes S with the two extreme cases of the observation shift. In the first case, we employ a shift equal to the analysis shift T_a (maximal temporal precision) and we refer to this case as *sliding smoothing*. In the second case, we employ an observation shift that is equal to ST_a (maximal computational speed-up). This



(a)



(b)

Figure 5.2: Influence of smoothing size on the SEMA chord results for two chord acoustic models and two smoothing approaches

approach is therefore called *disjoint smoothing*. The chord estimation results on the SEMA set for the four configurations as a function of the smoothing size are depicted in figure 5.2.

Averaging the observations ($S > 1$) clearly has a positive effect on the chord score. For the sliding window approach, the score increases until $S = 67$. For longer smoothing sizes, the drawback of possibly averaging over multiple chords, and thus smearing information, outweighs the advantage of creating a more stable input. In the case of disjoint smoothing, the optimal point is significantly lower and located at $S = 41$. As soon as $S > 11$, a loss of time resolution ($T_o > T_a$) causes a loss in accuracy. However, this loss is restricted to 5%. The difference between the two choices of the chord acoustic model is negligible.

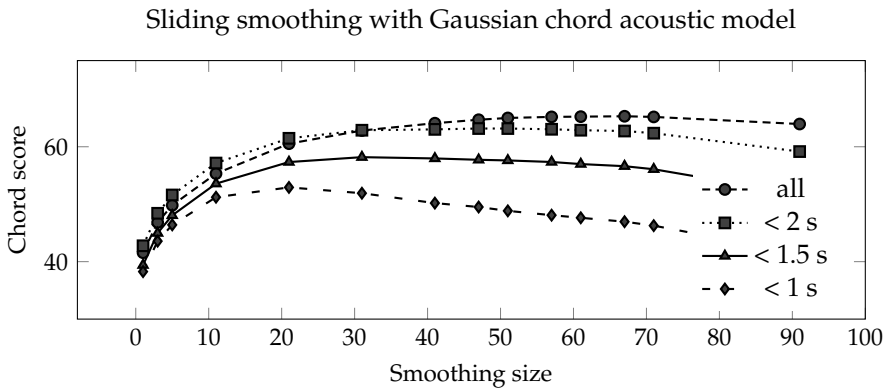


Figure 5.3: Influence of smoothing sizes on short chords

Although the results point to an optimal smoothing size that is relatively long, we argue that such a long smoothing is unacceptable because it leads to a large number of undetected short chords. Because of the nature of our evaluation measure, wrongly recognised short chords have a smaller impact on the total score than they would have in an evaluation where the performance of chords is not weighted by their length. In order to visualise this phenomenon, we have compared the results for multiple smoothing sizes on the entire evaluation data set with the results on subsets that only consist of chords below a certain duration (as determined from the reference annotations). The results for the Gaussian chord acoustic model are shown in figure 5.3, where we use sliding smoothing to exclude the effects of a diminishing temporal resolution associated with disjoint smoothing. We considered subsets of chords that were shorter than 2 s, 1.5 s and 1 s. They respectively account for 56%, 39% and 19% of the evaluation data. The fact that the curves diverge more and more as the smoothing size increases, illustrates that short chords are increasingly discarded. Consequently, the optimal smoothing size depends on the chord length, where shorter chords obviously require shorter smoothing sizes.

As we believe that the detection of short chords is important, we choose to fix the smoothing size to 11, which we support by the following arguments. First of all, this point is a good compromise between keeping the largest part of the initial surge in the score (the curves of figure 5.2 are not increasing as fast any more) while still making it possible to recognise short chords (the curves in figure 5.3 are still close to each other and none has passed its maximum). Second, for this smoothing size, there are only small (though significant) differences between disjoint and sliding smoothing, so that we can continue to examine both types of smoothing. Finally, we argue that by including a duration model, we will be able to restore the performance to the level of the optimal smoothing by preventing the

generation of spurious short chords, even when limited smoothing is used.

Note that the key acoustic model works on the same smoothed observations with a smoothing size of 11. This means that the extrema of the frames involved in the smoothing are only 350 ms ($= 150 + 10 \times 20$) apart from each other. Usually, the features for stand-alone key estimation algorithms (Pauws, 2004) and combined key-chord estimation algorithms (Rocher et al., 2010) are calculated over much longer intervals than this, since a key is by definition established over a longer time. However, we feel that the previously made reasoning applies here as well. Although the benefits of a long smoothing window may be even more substantial, such windows may lead to imprecise key boundary locations. The fact that we employ a short smoothing window means that the results of the key acoustic model on its own are not very informative though. Therefore they are not reported in detail here. As could be expected, the results are mediocre and keep on increasing with a larger smoothing size beyond the smoothing sizes we examined. In fact, the key acoustic model output tends to track chords rather than keys. It hence needs to be stabilised in the full system by imposing keys to change more slowly than chords.

5.2.3 Impact of the duration models

To assess the impact of the duration models, we use the full system, but with the change models switched off ($\beta_k = 0, \beta_c = 0$). The probability to optimise then is

$$P\left(K, C|\mathbf{X}\right) = \prod_{n=1}^{N_x} P\left(\mathbf{x_n}|k_n\right)^{\alpha_k} P\left(\mathbf{x_n}|c_n\right)^{\alpha_c} \begin{cases} P_{cd} & \text{if } k_n = k_{n-1} \wedge c_n = c_{n-1} \\ \left(1 - P_{cd}\right) P_{kd} & \\ & \text{if } k_n = k_{n-1} \wedge c_n \neq c_{n-1} \\ \left(1 - P_{cd}\right)\left(1 - P_{kd}\right) & \\ & \text{if } k_n \neq k_{n-1} \wedge c_n \neq c_{n-1} \end{cases} \tag{5.24}$$

The acoustic balance factors α_k and α_c are then altered in a grid search on the SEMA set to arrive at the optimal combination of acoustic and duration models. Because both key and chord acoustic models and key and chord duration models depend on either the key or the chord and because the change models that account for the key-chord interaction are switched off (i.e. uniformly distributed), the key and chord sequences can be optimised almost independently from each other. Changing the balance of the key acoustic model influences the chord sequence only minimally and vice versa. This configuration is therefore almost equal to fusing two separate HMMs, one for chord estimation and one for key estimation, except for the requirement that key changes should co-occur with chord changes.

	chord acoustic only		with duration models	
chord acoustic model	disjoint	sliding	disjoint	sliding
Gaussian	54.93	55.33	71.17	71.83
cosine similarity	55.02	55.42	69.34	70.27

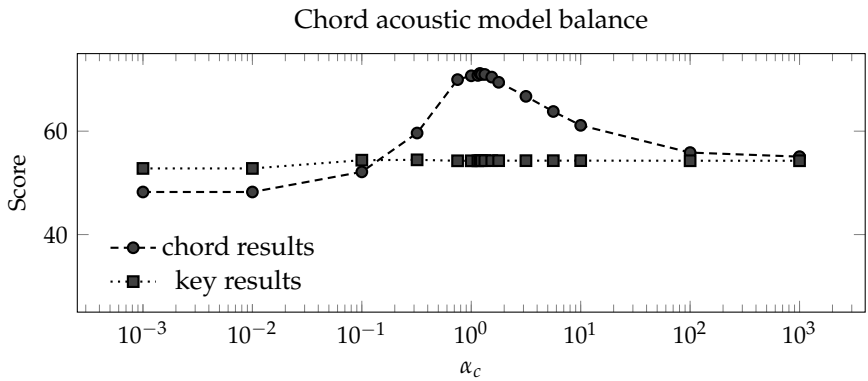
Table 5.3: Chord results on the SEMA set

	key acoustic only		with duration models	
chord acoustic model	disjoint	sliding	disjoint	sliding
Gaussian	25.89	25.75	54.33	53.63
cosine similarity	25.89	25.75	53.10	53.65

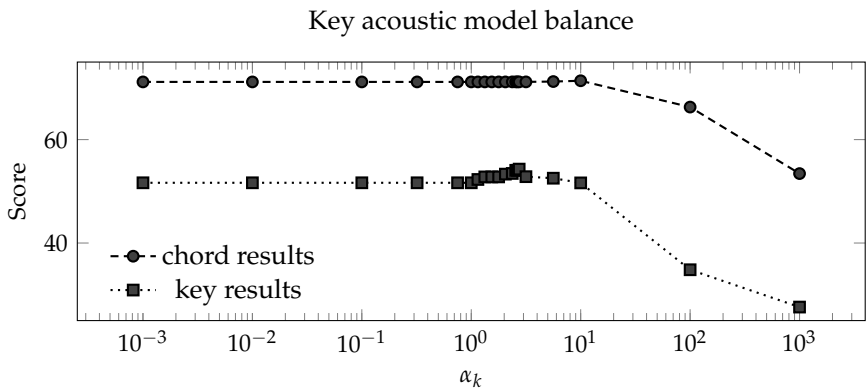
Table 5.4: Key results on the SEMA set

The chord and key results on the SEMA set presented in table 5.3 and table 5.4 show that the addition of the duration models leads to a substantial improvement. These improvements are all highly significant with $p < 10^{-8}$ or less. In contrast to the experiments with acoustic-only systems, the chord results show a consistent difference between the two chord acoustic models. The Gaussian model outperforms the cosine similarity model by more than one percentage point with significance of $p < 0.01$. The differences in key results are not significant, however. In fact, the number of files with a different key output between the two is so small that significance tests are rejected by default. The difference between sliding and disjoint smoothing remains small as well (<1), nevertheless it is statistically significant ($p < 0.02$ or less). The key estimation performance is notably lower than the chord estimation performance. The explanation is that the key acoustic model works on the same relatively short feature segments as used for the chord estimation, which we conclude to be too short to reliably track the correct key. Nonetheless, we previously stated our reasons not to increase the observation length. So far, the hypothesised chords only influence the duration of the key candidates, so the key-chord interaction is barely exploited. Even though this already leads to an absolute improvement of almost 30 %, we expect another large improvement in the key output when the actual chord labels can suggest the implied key through a non-uniform chord change model.

We now take a closer look at the results of the system with disjoint smoothing and a Gaussian chord acoustic model. In figure 5.4, the acoustic model balance factors α_k and α_c are altered around their optimal point ($\alpha_c = 1.2, \alpha_k = 2.74$), while keeping the other constant. We can see that changing the chord acoustic balance factor α_c strongly influences the chord results and has little effect on the key results, as was expected from this



(a)



(b)

Figure 5.4: Influence of acoustic model balance factors on the SEMA set

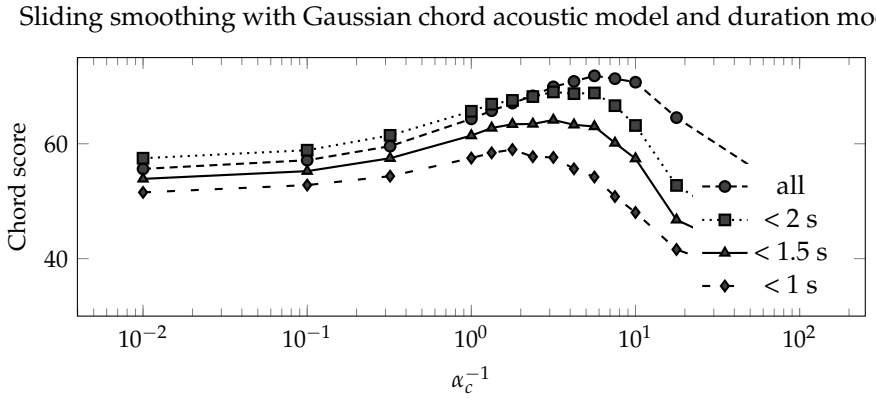


Figure 5.5: Influence of the chord acoustic model balance factor on short chords

configuration without change models. Changing α_k similarly does not influence the chord results. The fact that changes in one acoustic model balance exert such little influence on the other, show that imposing the requirement that key changes should co-occur with chord changes is reasonable. Only when α_k gets very large will this assumption force the chord changes to follow the superfluous key changes that arise when the key acoustic model strongly dominates the duration models. The chord results are then dragged down with the key results as can be seen in the right part of figure 5.4b, but since this only happens for such degenerate configurations with no practical use anyway, it is not really a problem. In addition, the sensitivity of the key output to α_k is also minimal. When α_k takes small values, the duration model becomes dominant and the output contains no key changes. Because of the relative rarity of key changes, this does not fundamentally alter the results. However, since the key acoustic model is the only component responsible for labelling key segments, setting $\alpha_k = 0$ would lead to random key output (with the chord output unaffected).

To end this section, we quickly look at the performance on short chords again to see if the addition of the duration models has changed the system's behaviour. In figure 5.5, the chord results of the sliding smoothing/Gaussian chord acoustic model configuration are displayed in relation to α_c^{-1}, split out according to maximal chord length. On the x-axis we put α^{-1}, such that going to the right equals an increase in dominance of the duration model, comparable with the increase in smoothing size of figure 5.3. Short chords still prove to be harder to estimate correctly, which is expected because less observations are available to make this estimation, and the globally optimum amount of duration modelling is still too aggressive for the shortest chords. On the other hand, the difference in performance between chords shorter than 1 s and all chords, measured

at the globally optimal point, has decreased slightly. Moreover, although aligning the optimal duration model setting such that all chord lengths achieve their maximal improvement at the same time has proven to be impossible, the results for each of the chord-length categories at the globally optimal setting are still better than best results achieved through adapting the feature smoothing size to each of the chord lengths separately in an acoustic-only system (which cannot be done in practice because it requires knowledge of the reference chord segmentation). So all-in-all, the introduction of the duration models does allow us to improve the performance compared to a system with more aggressive smoothing and no duration models.

5.2.4 Impact of chord change models

5.2.4.1 Adding a chord change model

Next, we evaluate the contribution of the chord change model by introducing a positive chord change balance factor β_c. The probability to optimise then takes the following form:

$$P\left(K, C|\mathbf{X}\right) = \prod_{n=1}^{N_x} P\left(\mathbf{x_n}|k_n\right)^{\alpha_k} P\left(\mathbf{x_n}|c_n\right)^{\alpha_c} \begin{cases} P_{cd} & \text{if } k_n = k_{n-1} \wedge c_n = c_{n-1} \\ \left(1 - P_{cd}\right) P_{cc2}^{\beta_c} P_{kd} & \\ & \text{if } k_n = k_{n-1} \wedge c_n \neq c_{n-1} \\ \left(1 - P_{cd}\right) P_{cc2}^{\beta_c} \left(1 - P_{kd}\right) & \\ & \text{if } k_n \neq k_{n-1} \wedge c_n \neq c_{n-1} \end{cases} \tag{5.25}$$

We use a system with a Gaussian chord acoustic model and disjoint smoothing to calculate the results, as this turned out to be the optimal trade-off between accuracy and speed in the previous experiments. All chord change models described in section 4.5 are put to test and the SEMA set is used as a development set for seeking the optimal combination of $(\alpha_k, \alpha_c, \beta_c)$ for each of these chord change models.

With the introduction of the chord change models, a new complication arises. Previously, the key and the chord output exerted so little influence on each other that it was feasible to attain the best chord and the best key performance with a single parameter configuration. Now it is possible that the two transcription tasks compete with each other instead of reinforcing each other, especially when the chord change model is not matched to the data set. It might be better to insert more chord changes or to choose slightly suboptimal chords that fit the mismatched chord change model better, such that the correct key is found over a longer time span. The reverse case is also true, optimising the detection of short-term chords

might cause the key estimation to lose track. This makes it generally impossible the get the best key and chord sequence with the same configuration. Therefore we first determined for each chord change model the parameter combination that maximises the chord score (displayed in the left column of table 5.5a), and then the combination that maximises the key score (right column of table 5.5a). In order to understand better how the optimal key and optimal chord configuration relate to each other, we show the average over all files of the ratio of the estimated number of chord segments CS_{est} to the number of chord segments in the reference annotations CS_{ref}. A value higher than one signifies over-segmentation, below one means under-segmentation.

The eleven optimal balance parameter configurations (and not twelve because the optimal chord and optimal key configuration are the same for the uniform chord change model) that follow from this parameter sweep are subsequently used to calculate the output on the Isophonics data set. However, the balance parameters $(\alpha_c, \alpha_k, \beta_c)$ for the SEMA and the Isophonics chord change model are interchanged for this experiment in order to follow the match between data set and chord change model. The resulting key and chord scores can be found in table 5.5b.

CHORD ESTIMATION RESULTS The results show that all of the chord change models improve the chord estimation results, and all improvements on both sets are statistically significant with a minimum of $p < 0.02$ with respect to the uniform chord change model. However, in absolute terms the improvement is rather limited, with an increase of about 2 %-points.

KEY ESTIMATION RESULTS The impact on the key estimation can be higher, with an increase up to 17.5 %-points on the SEMA set and 8 %-points on the Isophonics set. But here we see that results vary greatly across chord change models. On the SEMA set, the MySong and theoretical models are only able to achieve small improvements, even non-significant in the case of the theoretical model (the other improvements are statistically significant). These key results mirror the trends of the model specificity tests of table 4.8 very well. On the Isophonics set, only the matched Isophonics chord change model is able to significantly improve the key output $(p < 10^{-3})$.

DATA GENERALISATION AND PARAMETER SENSITIVITY Despite the use of parameters that are optimised for the SEMA data set, the figures reveal that chord and key estimation is inherently easier on the Isophonics set. This does not come as a surprise, since the Beatles songs that form the majority of the Isophonics set are known for their harmony based compositions, while the SEMA set was not assembled particularly for chord and key estimation and thus contains a number of more rhythmically oriented songs. The fact that especially key estimation is harder for the SEMA set

	optimal chord configuration				optimal key configuration			
	chord	key	$(\alpha_c, \alpha_k, \beta_c)$	$\overline{\left(\frac{CS_{est}}{CS_{ref}}\right)}$	chord	key	$(\alpha_c, \alpha_k, \beta_c)$	$\overline{\left(\frac{CS_{est}}{CS_{ref}}\right)}$
uniform	**71.17**	54.33	$(1.2, 2.74, 0)$	1.72	71.17	**54.33**	$(1.2, 2.74, 0)$	1.72
SEMA	**73.46**	71.15	$(0.75, 0.27, 0.65)$	1.46	73.28	**71.85**	$(0.75, 0.27, 0.75)$	1.48
Isophonics	**72.92**	62.43	$(0.75, 1, 1)$	1.53	71.64	**67.90**	$(1.33, 1, 1.78)$	2.51
9GDB	**73.15**	62.65	$(0.75, 0.56, 1)$	1.55	72.47	**67.47**	$(1, 0.56, 1.78)$	1.95
MySong	**73.43**	47.31	$(0.87, 0.1, 2.05)$	1.80	69.75	**58.53**	$(2.05, 2.37, 1.33)$	3.51
theoretical	**73.08**	51.19	$(0.75, 7.5, 1.33)$	1.51	66.30	**54.96**	$(0.42, 2.37, 0.32)$	0.75

(a) SEMA data set

	optimal chord configuration				optimal key configuration			
	chord	key	$(\alpha_c, \alpha_k, \beta_c)$	$\overline{\left(\frac{CS_{est}}{CS_{ref}}\right)}$	chord	key	$(\alpha_c, \alpha_k, \beta_c)$	$\overline{\left(\frac{CS_{est}}{CS_{ref}}\right)}$
uniform	**76.45**	68.54	$(1.2, 2.74, 0)$	0.90	76.45	**68.54**	$(1.2, 2.74, 0)$	0.90
SEMA	78.16	**71.33**	$(0.75, 1, 1)$	0.84	**78.18**	66.86	$(1.33, 1, 1.78)$	1.19
Isophonics	78.56	76.00	$(0.75, 0.27, 0.65)$	0.81	**78.83**	**76.66**	$(0.75, 0.27, 0.75)$	0.82
9GDB	**78.41**	66.65	$(0.75, 0.56, 1)$	0.85	78.37	**66.78**	$(1, 0.56, 1.78)$	1.04
MySong	**78.50**	57.64	$(0.87, 0.1, 2.05)$	0.94	75.84	**65.23**	$(2.05, 2.37, 1.33)$	1.55
theoretical	**77.26**	61.51	$(0.75, 7.5, 1.33)$	0.82	69.93	**69.08**	$(0.42, 2.37, 0.32)$	0.47

(b) Isophonics data set

Table 5.5: Key and chord score of the configurations that maximise chord (left column) or key score (right column) for different chord change models. The best results per chord change model are set in boldface.

than for the Isophonics set possibly explains why only the matched Isophonics chord change model can improve the key results: there is simply less room for improvement. Another reason could be due to the difference in parameter sensitivity between chord and key estimation. The chord score hypersurface formed by varying the balance parameters for the SEMA set is smooth and monotonous, such that it is easy to find the optimal configuration. The resulting configurations carry over well to the unseen Isophonics data set. The key score hypersurface on the other hand is rough, with irregular ripples that can change the score by multiple percentage points between nearby parameter combinations. This in turn can be explained by the fact that the longer temporal nature of a key makes a single change in key label affect a longer time span than a single change in chord label. The effect is that finding the optimal key configurations is harder and that they generalise worse to the Isophonics data than the optimal chord configurations. The latter is apparent when the SEMA chord change model is used with the Isophonics data set, for instance. The supposedly optimal chord configuration here has a better key estimation result than the supposedly optimal key configuration, and the other way around. Finally, it can be argued that the degree to which the test data complies with a chord change model is inherently data-dependent and that the balances need to reflect this. This would prove problematic for unseen data, as there is no way to find the optimal balance beforehand then. The model specificity tests of table 4.8 particularly suggest that the 9GDB model should be capable of an increase in key estimation results on the Isophonics set too. Where the configuration lies that achieves this result and if this configuration is also satisfactory for the SEMA data set can only be answered through a new parameter sweep on the Isophonics data. Future research with additional data sets will hopefully resolve this question of data generalisation.

CHORD SEGMENTATION ANALYSIS The chord segmentation ratio brings another data-dependency to light. The system without a chord change model produces over-segmented chord sequences on the SEMA data, but under-segmented sequences on the Isophonics set. For both sets, adding a chord change model and configuring it for optimal chord estimation generally results in a decrease of the estimated number of chord segments, irrespective of whether the base uniform system was already under-segmenting or not. However, improving the chord rate and a better chord score do not necessarily go hand in hand. For instance, using the MySong chord model instead of the uniform model on the SEMA set leads to a significant improvement in chord score, but also increases the over-segmentation. Even when using the same chord change model, i.e. the 9GDB model on the Isophonics set, we can see that the chord scores of both configurations are the same, although the optimal key configuration has a better segmentation.

KEY-CHORD REINFORCEMENT We notice that when using the worst chord change models (MySong and theoretical), the improvements with respect to the uniform model in both key and chord estimation are only possible at the expense of the complimentary task, whose result drops below the baseline set by the uniform model. This further confirms their unsuitability for the current data sets. The matched chord change model is of course most capable of combining both tasks in a single configuration. Surprisingly, the correlation between key results and chord results is rather low. It is possible to improve chord output while the key output deteriorates compared to the uniform model case. We will investigate how this is possible in the next section.

SUMMARY In summary, the choice of chord change model does not make a huge difference as far as chord estimation is concerned, whereas it is clearly a major factor for key estimation. Consequently, the effects of a mismatch between chord change model and data set or between model balance and data set are more severe for the key estimation results, to the extent that a mismatch can easily degrade key estimation performance until it is worse than when a base-line uniform model is used. For chord estimation, the effect of adding a chord change model is much more consistent, such that a performance increase over the uniform model is almost guaranteed, although unseen data sets might not get all the potential out of them. As can be expected, the relative chord change models that match the test set perform the best, both for key and chord estimation. The test set perplexity described in section 4.6 gives a good indication of this match. More in general, the bigger the mismatch between chord change model and data set, the harder it is to achieve a synergetic effect between the two tasks of key and chord estimation.

5.2.4.2 Error analysis

The fact that the key and chord results are not strongly correlated shows that the key and chord estimation processes do not use the information contained in the chord change models in the same way. In order to get a better understanding of the effect of the chord change models, we make an in-depth analysis of the type of errors our system makes. We use the experiments on the Isophonics data, as this has not been used to tune the parameters and therefore its results are most representative.

CHORD ERROR CATEGORISATION We divide the chord estimation errors into four categories. A first class comprises all wrongly estimated chords that *differ only one chroma* from the correct chord (or even have all chromas correct but got the root wrong for augmented chords). A second category consists of the chords that are *shifted a perfect fifth* up or down with respect to the reference chord. The chord type is therefore correctly estimated, but

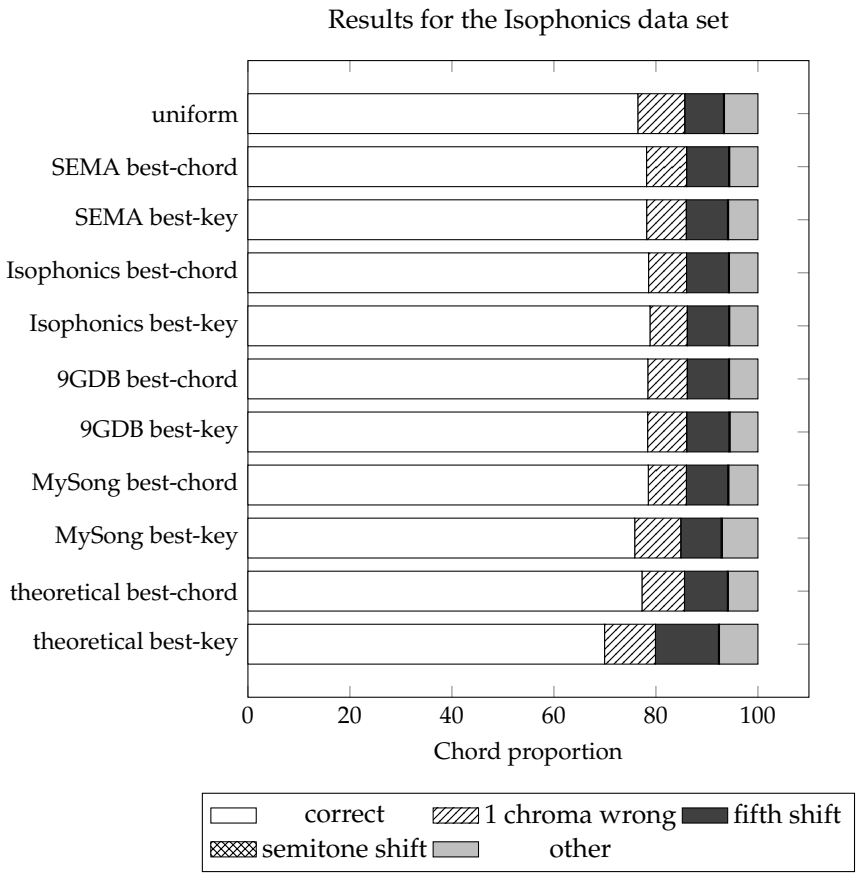


Figure 5.6: Chord estimation output on the Isophonics data broken down into categories

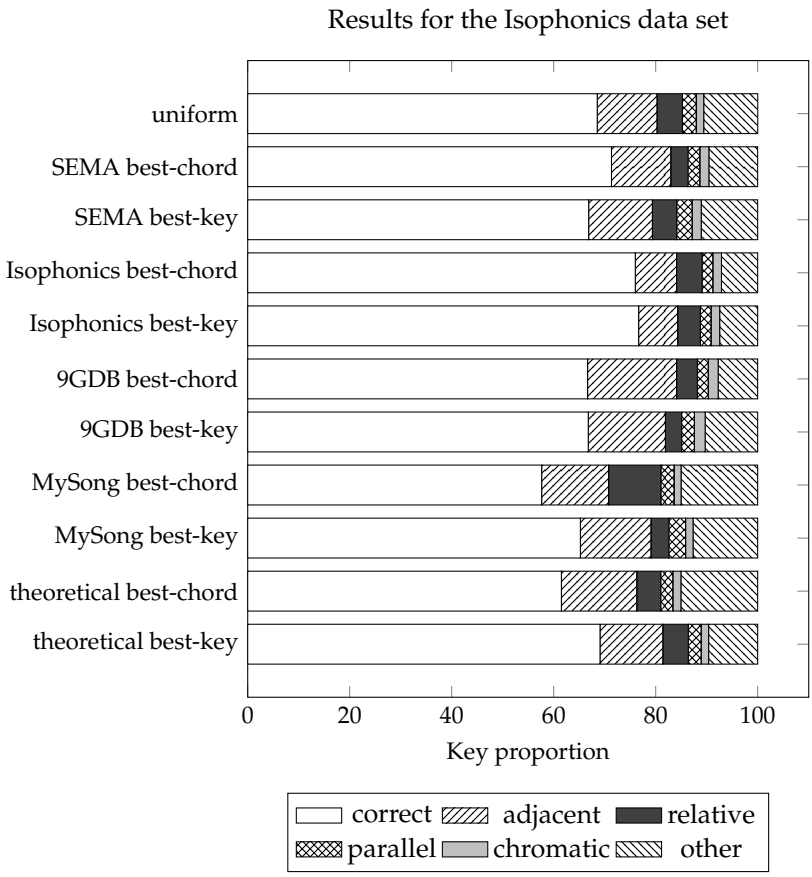


Figure 5.7: Key estimation output on the Isophonics data broken down into categories

the estimated root is adjacent to the reference root on the circle of fifths (e.g. *F*maj and *G*maj for a *C*maj annotation). Note that these chords have at most one chroma in common with the annotated chord (1 for maj and min chords, 0 for dim and aug chords). The third class consists of the chords that are *shifted a semitone*, such that the chord type is again correct but the root is a semitone higher or lower than the annotation (e.g. *C*♯maj and *B*maj for a *C*maj annotation). These chords have no chromas in common with the annotated chord. The remaining chords share maximum one chroma with the annotation and are grouped together in a fourth category.

KEY ERROR CATEGORISATION For the key estimation results, we identify five categories of errors. The first one consists of the *adjacent keys*, whose modes are correct but whose tonics are adjacent to the annotated tonic on the circle of fifths. The second category contains the *relative keys*, that share the same key signature, but where the mode is flipped (e.g. *A* minor for a *C* major annotation and *E*♭ major for a *C* minor annotation). Third are the *parallel keys* where the tonic is correct, but the mode is flipped (e.g. *C* minor for a *C* major annotation). Fourth are the *chromatic keys,* whose mode is correct, but whose tonic is a semitone off the correct key. Finally, all remaining key errors form the last category.

CHORD ERROR ANALYSIS The breakdown per chord category in figure 5.6 tells us that the addition of a chord change model is most effective in reducing chords that differ in one chroma from the correct chord, the other categories decrease less or not at all. Missing just one of the chromas is acoustically quite likely, because of various causes of noise in the short-time signal. On the other hand, harmonically this can lead to chords that are very different in function. It is therefore understandable that the chord change model is best able to correct these errors that are very conspicuous in the harmonic sense. Another notable observation from the chord errors is the large proportion of fifth shift errors. This is somewhat surprising, as they are acoustically not very similar to the chords they replace (maximum one shared chroma). It is even more remarkable when one realises that they form only 4% of the possible confusion pairs, while the chords with one wrong chroma represent a 16% share. The fact that these errors are also prevalent with the uniform chord change model indicates that these errors originate before the harmony modelling. It seems plausible to assume that these errors stem from mistakes in the chroma analysis, as fifth confusions between chromas (due to erroneous tracking of the first partial) are acoustically a lot more common than confusing two chords a fifth apart. More intelligent chord change models do not help to decrease these fifth shift errors, on the contrary. This might seem counterintuitive, but chords that are shifted a fifth still fit in the original key in most cases, so transitioning to them instead of to the annotated chord will not be penalised by the chord change model. For that reason, we hypothesise that using the

more restrictive chord change models also has the unwelcome side-effect that once an error has been made (e.g. because of the tracking of a partial instead of a fundamental frequency in the chroma analysis), the stronger harmonic constraints make it more difficult to quickly recover from the errors in the acoustic model. Luckily, the benefits of a better adapted chord change model still outweigh this unwanted drawback. Chromatically shifted chords are both acoustically and harmonically unrelated, so the only cause of these errors would be tuning problems in the chroma extraction step. The unvaryingly low numbers show that this is not the case. Finally, the use of chord change models leads to a decrease of the uncategorised errors, which can be expected because the chord change model makes it more difficult to generate chords that are completely unrelated to the surrounding chords.

KEY ERROR ANALYSIS The counter-productive effect of using better chord change models is less visible in the key errors, shown in figure 5.7: an increase in correct key estimations does not lead to a systematic increase in a specific error category. Instead, all error percentages decrease or at least stay constant with respect to the uniform chord change model[2]. This is undoubtedly because a single wrongly estimated chord does not disrupt the key estimation that much. The gain of the best performing configurations seems to lie in a reduction of adjacent key errors and of the unrelated keys. The former are strongly harmonically related keys of the same mode, so here the chord change model is able to clear up the confusion between nearby tonics. The relative and parallel keys are also closely related harmonically, but differ in mode. The fact that these two error categories are much more stable shows that the chord change models are less successful in correcting wrong modes. In general, the harmonically related key errors (adjacent, relative and parallel) account for the majority of the errors, despite that they only represent 17.39% of the *possible* confusion pairs (8.70% adjacent, 4.35% relative and parallel each). The chords belonging to these harmonically related keys are greatly overlapping, which explains why roughly speaking the same chord scores can be achieved by systems that differ almost 20% in key score. Chromatic errors are indicative of tuning problems and therefore are indifferent with respect to chord change model, as expected. Finally, because a chord change model forces the system to take the chord sequence into account when estimating the key, we see that in the best performing configurations, where there is a synergetic effect between the two tasks, the unrelated key errors decrease significantly as well.

SUMMARY We conclude that the information contained in the chord change models is mostly used to reject chords that are acoustically close, but

[2]We ignore the degenerate configurations where the percentage of correctly estimated keys decreases. At least some error categories obviously need to increase in these cases.

harmonically far off, not only in the correct key, but also in the related keys. Finding the exact key is therefore not an absolute requirement for the chord estimation to benefit from a chord change model, as the key-chord relation is quite loose and only the extrema of the probabilities are used. It is consequently also less important that the data set matches the chord change model perfectly. The key estimation on the other hand uses the full spectrum of information present in the chord change models. Subtle differences in the probability of one and the same chord sequence between models can thus easily steer the key estimation towards a related key.

5.2.4.3 Without key acoustic model

Since the role of the chord change model is so decisive for the key results, it would be interesting to know to what extent the key acoustic model plays a role in the estimation. Therefore we look at the best performance we can get from the system without using the key acoustic model ($\alpha_k = 0$), which means that the probability to optimise is

$$P\left(K, C|\mathbf{X}\right) = \prod_{n=1}^{N_x} P\left(\mathbf{x_n}|c_n\right)^{\alpha_c} \begin{cases} P_{cd} & \text{if } k_n = k_{n-1} \wedge c_n = c_{n-1} \\ \left(1 - P_{cd}\right) P_{cc2}^{\beta_c} P_{kd} & \text{if } k_n = k_{n-1} \wedge c_n \neq c_{n-1} \\ \left(1 - P_{cd}\right) P_{cc2}^{\beta_c} \left(1 - P_{kd}\right) & \text{if } k_n \neq k_{n-1} \wedge c_n \neq c_{n-1} \end{cases} \tag{5.26}$$

The estimated key is then completely determined by how well the estimated chords fit the chord change model. This sort of configuration has been examined before by Burgoyne and Saul (2005); Catteau et al. (2007), and it also resembles the first experiments that ultimately culminated in this thesis (Pauwels and Martens, 2010). The best performing chord and key configurations on the SEMA set and their corresponding scores for the different chord change models can be found in table 5.6a. As before, these configurations are then reused on the Isophonics set which leads to the results in table 5.6b. The differences with respect to table 5.5a, respectively table 5.5b, are added in brackets to the results.

Comparing the results with and without key acoustic model tells us that including a key acoustic model is beneficial for both key and chord estimation performance. For chord estimation, the effect is rather limited, setting α_k to zero only leads to a decrease less than 1% on the SEMA set and less than 2% on the Isophonics set. The decrease is consistent across chord change models, which proves the presence of a synergetic effect between key and chord estimation. The influence of the key acoustic model on the key estimation results is understandably much bigger. Switching it off leads to absolute decreases ranging between 5% and 20% on the SEMA set and between 3% and 37% for the Isophonics set[3]. The decrease in es-

[3]We ignore the key results of the uniform model here, because without key acoustic model

	optimal chord configuration			optimal key configuration		
	chord	key	$(\alpha_c, \alpha_k, \beta_c)$	chord	key	$(\alpha_c, \alpha_k, \beta_c)$
uniform	**71.16** (-0.01)	7.76 (-46.57)	$(1.2, 0, 0)$	71.16 (-0.01)	**7.76** (-46.57)	$(1.2, 0, 0)$
SEMA	**73.16** (-0.30)	63.49 (-7.66)	$(0.75, 0, 0.87)$	71.06 (-2.22)	**66.92** (-4.93)	$(1.78, 0, 0.75)$
Isophonics	**72.84** (-0.08)	55.27 (-7.16)	$(0.87, 0, 0.87)$	71.03 (-0.61)	**62.88** (-5.02)	$(1.54, 0, 1.07)$
9GDB	**72.88** (-0.27)	49.85 (-12.80)	$(0.75, 0, 0.87)$	71.24 (-1.23)	**57.28** (-10.19)	$(1.33, 0, 0.87)$
MySong	**73.37** (-0.06)	42.27 (-5.04)	$(0.87, 0, 2.05)$	70.68 (+0.93)	**51.43** (-7.10)	$(1.78, 0, 1.15)$
theoretical	**72.54** (-0.54)	32.33 (-18.86)	$(0.75, 0, 1.07)$	71.63 (+5.33)	**35.82** (-19.14)	$(0.65, 0, 1.24)$

(a) SEMA set

	optimal chord configuration			optimal key configuration		
	chord	key	$(\alpha_c, \alpha_k, \beta_c)$	chord	key	$(\alpha_c, \alpha_k, \beta_c)$
uniform	**75.20** (-1.25)	16.64 (-51.90)	$(1.2, 0, 0)$	75.20 (-1.25)	**16.64** (-51.90)	$(1.2, 0, 0)$
SEMA	**77.36** (-0.80)	**62.49** (-8.84)	$(0.87, 0, 0.87)$	76.15 (-2.03)	60.27 (-6.59)	$(1.54, 0, 1.07)$
Isophonics	**77.67** (-0.89)	**72.71** (-3.29)	$(0.75, 0, 0.87)$	75.48 (-3.35)	71.16 (-5.50)	$(1.78, 0, 0.75)$
9GDB	**76.98** (-1.43)	**58.19** (-8.46)	$(0.75, 0, 0.87)$	76.46 (-1.91)	55.13 (-11.65)	$(1.33, 0, 0.87)$
MySong	**77.16** (-1.34)	54.15 (-3.49)	$(0.87, 0, 2.05)$	75.38 (-0.46)	**61.41** (-3.82)	$(1.78, 0, 1.15)$
theoretical	**76.22** (-1.04)	**33.44** (-28.07)	$(0.75, 0, 1.07)$	75.44 (+5.51)	32.02 (-37.06)	$(0.65, 0, 1.24)$

(b) Isophonics set

Table 5.6: Key and chord score of the configurations without key acoustic model that maximise chord (left column) or key score (right column) for different chord change models. The best results per chord change model are set in boldface. The difference with respect to the corresponding configurations with key acoustic model is displayed in parenthesis.

timation performance also seems to reflect more or less the level of mismatch between chord change model and data set (as seen in table 4.8). The key acoustic model is thus able to compensate for the mismatch in chord change model to some extent. On the other hand, with a well-matched chord change model, the system already gets fair results without key acoustic model. In that ideal case, the latter configuration ($\alpha_k = 0$) even outperforms the best system with both key and chord acoustic models but no chord change model ($\beta_c = 0$, the uniform case of table 5.5a and table 5.5b). Naturally, the best results are achieved when all models are used together.

The results on the Isophonics set in table 5.6b show once more that the optimal configurations do not translate perfectly to other data sets, which is for example apparent in the fact that in most cases the optimal chord configuration also has a better key estimation score than the optimal key configuration. Most likely another configuration can be found that improves this score even more, but this cannot be done without performing a balance parameter search over the Isophonics set. Nonetheless, the trends for the key estimation score are the same as for the SEMA set, showing that the key acoustic model and the chord change model reinforce each other.

5.2.5 Impact of key change models

Finally, the influence of the key change models is assessed. Therefore we use the full modelling capacities of our system

$$P\left(K, C|\mathbf{X}\right) = \prod_{n=1}^{N_x} P\left(\mathbf{x_n}|k_n\right)^{\alpha_k} P\left(\mathbf{x_n}|c_n\right)^{\alpha_c} \begin{cases} P_{cd} & \text{if } k_n = k_{n-1} \wedge c_n = c_{n-1} \\ \left(1 - P_{cd}\right) P_{cc2}^{\beta_c} P_{kd} & \\ & \text{if } k_n = k_{n-1} \wedge c_n \neq c_{n-1} \\ \left(1 - P_{cd}\right) P_{cc2}^{\beta_c} \left(1 - P_{kd}\right) P_{kc2}^{\beta_k} & \\ & \text{if } k_n \neq k_{n-1} \wedge c_n \neq c_{n-1} \end{cases} \tag{5.27}$$

Two separate points are being examined here. The first is whether swapping the uniformly distributed key change model, used so far, for the Lerdahl-based model, presented in section 4.4, leads to a difference in the estimation of the local keys. The second question is whether the extra chord diatonicity constraints imposed around key changes (see section 4.2.2) change the estimated key output. To examine these issues, we let the key change parameter β_k gradually increase from zero such that the uniform key change model evolves into the Lerdahl-based one and we try each variant f_{kc2a},

and without chord change model, all keys are considered equally likely and stay constant. Due to implementation details, a constant A major is generated for each file. The better than random results are therefore just lucky guesses.

f_{kc2b} and f_{kc2c} of the diatonic constraints. These experiments are repeated for multiple chord change models.

Because the SEMA data set contains only 30 s snippets, and because key changes are relatively rare, only 3.52% of the files contain at least one key change. Any conclusions with respect to key change models drawn from the experiments performed on this set would therefore be unreliable. Consequently, we exceptionally carry out the parameter sweep over β_k on the Isophonics data. In this set, at least one key change is present in 19.5% of the files.

Using the Lerdahl key change model and adding the diatonic chord constraints around key changes turns out to have extremely little impact on the estimated keys. The maximal improvement compared to the uniform key model tested over all combinations of chord change models and key change model variants is 0.23%, which naturally is non-significant. On multiple occasions the Lerdahl key change model does not improve the estimation at all. Specifically comparing the f_{kc2a}, f_{kc2b} and f_{kc2c} variants to each other shows that their results all lie within 0.5% of each other. None of the variants is consistently better than the others and no difference is significant. A positive consequence of this is that the f_{kc2c} variant can be used by default without loss in performance, but it has the advantage that the number of key change candidates gets reduced by an order of magnitude compared to the f_{kc2a} variant. A significant decrease in the required computational power can therefore be achieved by exploiting these diatonicity constraints in the system implementation.

The fact that the Lerdahl key change model and the diatonicity constraints fail to change the estimated key output significantly does not mean that these models are a bad fit for the data. Strongly mismatched models would lead to a deterioration. Instead it is more likely that the other components of the system already cause the system to generate outputs that fit the key change model close enough such that it does not lead to additional improvements. Another obvious explanation for the lack of influence of the key change models would be that the combination of acoustic and duration models makes the system generate no key change at all. In this case, the variants would cause changes in the intermediate results, but nothing different would be visible in the generated output. This however is not the case in our experiments: the ratio of generated files with at least one key change ranges from 12.4% to 44.8%. The highest number appears when the uniform chord change model is used. Only in combination with the matched Isophonics chord change model does the number of files with at least one key change drop below 19.5%, the ratio in the annotations. All used system configurations therefore support the ability of the key change model to manipulate the estimation process, but not (yet) in a way that leads to improved results.

5.3 Conclusion

We presented a probabilistic framework for the simultaneous estimation of keys and chords. First, we formulated it as a combination of acoustic models, duration models and change models, where the latter two encode prior musical knowledge. Multiple alternatives for each of the components were proposed, where the change models are those we derived in the previous chapter. Through multiple experiments, we tested a number of configurations and the alternatives for each of the models. This way we could determine the impact of the different components on the system's performance. We started from a baseline acoustic-only system that embeds no prior musicological knowledge to examine the effect of the feature extraction. We extended the system with increasing amounts of prior knowledge. We discovered that the inclusion of information about the mean key and chord duration leads to a significant gain in performance for both key and chord estimation. On the other hand, extra context information in the form of chord change models only yields a small additional improvement of 2 to 3% to the chord performance. Since we first established this finding (Pauwels and Martens, 2010), it has been independently confirmed by others as well (Chen et al., 2012; Cho and Bello, 2014). This result seems to explain why a system that ignores all musicological context, the one of Oudre et al. (2011), was one of the best performing submissions at the MIREX 2009 chord estimation competition. Since then, there has been a tendency to move to more complex systems that contain much more context information, but there is no proof yet that the subsequent increase in performance is due to the advanced modelling of musicological context. It can as well be owed to improvements in acoustic or duration modelling.

However, in this chapter we showed that a chord change model that is well adapted to the data set helps to achieve a better key estimation. The improvement can be as high as 17.5%. Unfortunately, key estimation is also more susceptible to a mismatch between chord change model and data set. Here we could link our results to the data specificity of the model as studied in the previous chapter. Through an in-depth analysis of the errors our system makes, we have been able to explain this difference in system sensitivity between the chord and key estimation tasks, despite their strong intertwinement and the fact that they use the same models. Finally, we found out that the impact of the key change model is negligible. The supplementary prior knowledge it contains does not alter the system's output, so we conclude that the same information is already covered by the other system components.

6 A detailed investigation of duration modelling

In this chapter, we will examine the benefits of extending the system presented in the previous chapter with a better chord duration modelling. We perform experiments to deepen our understanding of the role the duration model plays in the estimation process and to assess the benefits of extended duration modelling. First, we consider making the prior mean chord duration a function of the relative interpretation of that chord in the associated key. Then, we explore the influence of alternative shapes for the prior chord duration distribution. In particular, we propose to use a negative binomial shape that fits the actual chord distribution observed for some data sets.

6.1 Relative chord-specific prior mean durations

One of the assumptions we made in the previous chapter is that each chord has the same prior mean duration. In the remainder of this section, we consider a prior duration that is chord and key specific. We believe that there is no theoretical foundation to assume different mean durations for absolute chords, not interpreted in a key, e.g. we do not believe that a *C*maj chord lasts longer on average than an *A*min chord. However, we feel that some distinction can be made when interpreting chords in a key. In a similar way as for the change models, we further let the prior duration only depend on the mode of the key, not on the tonic, such that $P_{cd} : (m, c') \mapsto [0, \ldots, 1]$.

This means that we add an extra $N_m N_c - 1$ parameters (95 for our choice of key and chord vocabularies), which represent the mean durations of each relative chord-mode combination.

We call the chord duration models with a value for each relative chord-mode combination *specific chord duration models*, as opposed to the original *global model* with a single constant value. An intuitive musicological assumption would be that the duration of a diatonic chord is on average longer than that of a non-diatonic chord, as it provides a perceptually stable point of support in a musical phrase. Such an intuition is hard to quantify however. That is why we simply measure mean durations in data sets instead.

We consider two chord duration models, one derived from the SEMA set and one derived from the Isophonics set. To this end, we measured the mean duration per relative chord-mode combination in both data sets. The minimum duration (equal to the observations shift $T_x = 220$ms) was assigned to every combination that has no observations. Especially relevant will be how good the SEMA chord duration model performs on the Isophonics set and the other way around. Our aim is to verify whether an increase in performance can be achieved universally, and is not just a consequence of the increase in degrees of freedom coming from the better fit of the data.

The two specific chord duration models replace the global models in each of the eleven chord change model configurations found in section 5.2.4.1 (the eleven configurations arise from the fact that the best chord and best key configurations are the same for the uniform chord change model). The newly formed 22 configurations are then run on each of the two data sets. The score differences with respect to the corresponding global duration results, as shown in table 5.5a and table 5.5b, are calculated. These differences are then aggregated per specific duration model as Tukey box-plots in figure 6.1 and figure 6.2 (where the whiskers indicate the extrema that lie within 1.5 times the interquartile distance from the quartiles).

The results indicate that key estimation generally benefits from having a relative chord specific duration model. This is most apparent from the experiments on the Isophonics set, where the improvement is almost unanimous, with an absolute average of $2 - 3\%$. The experiments on the SEMA set are less convincing, with an average increase of only 1% and with a number of decreasing scores. On closer inspection of the non-aggregated results, there are just two of the eleven configurations with specific chord durations that yield a strong decrease in performance. The configurations in question are the best-chord and best-key settings for the matched SEMA chord change model. These give the overall best results, but are overly optimistic because they are best tuned to the data set. The other, more real-world configurations are still able to exploit the specific chord durations to close the gap with the SEMA chord change configurations.

Furthermore, the specific duration models learned on one set seem to

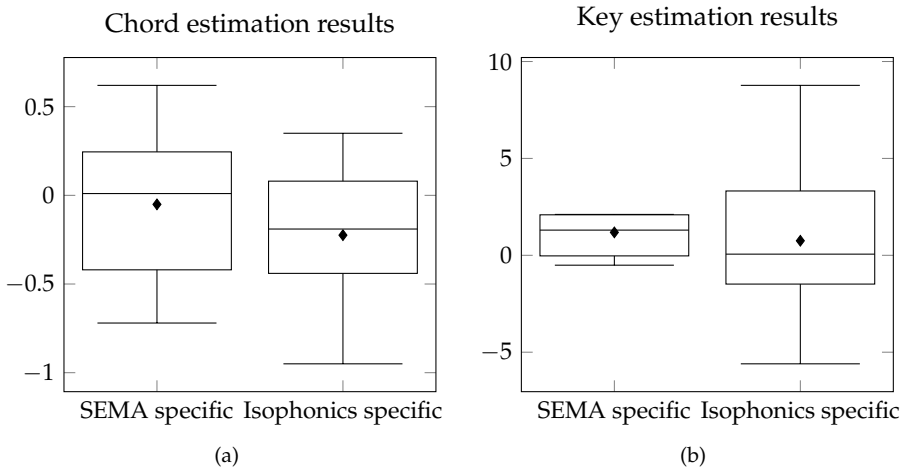


Figure 6.1: Aggregated difference in key and chord scores obtained on the SEMA set by replacing the global mean chord duration by relative chord specific durations retrieved from the SEMA and Isophonics sets

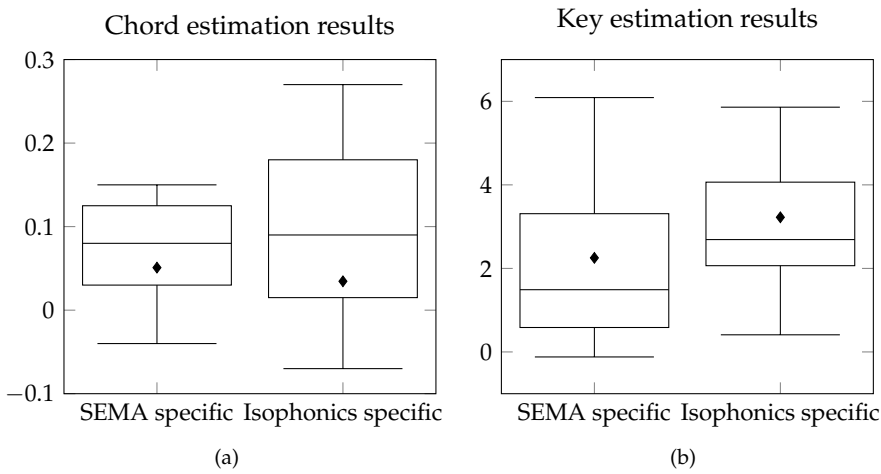


Figure 6.2: Aggregated difference in key and chord scores obtained on the Isophonics set by replacing the global mean chord duration by relative chord specific durations retrieved from the SEMA and Isophonics sets

generalise well to the other set. As can be expected, the cases where the specific chord durations are matched to the test set perform the best, but most of the improvement survives in the other cases. This implies that the specific chord duration models capture some musicological knowledge that is shared between the two data sets. A bit surprisingly, the chord estimation results are mostly insensitive to differences in the duration model, but then, this result confirms our previous finding that musicological modelling has only a small influence on chord estimation performance. An analysis of the key errors shows that most of the improvement comes from adjacent key errors and to a lesser extent from relative key errors that are being corrected. This means that the generated keys are not dramatically different in comparison to the global duration configurations. In this situation, the coupling between keys and chords is too weak to cause an indirect effect on the chord output.

6.2 Alternative chord duration distributions

6.2.1 Introducing explicit duration hidden Markov models

A consequence of the hidden Markov model (HMM) framework that we adopted up to now, is that the prior chord duration is implicitly modelled by a geometric distribution. We used single states for each key-chord combination, and the probability of staying in a state for n time steps is $P_{cd}(d) = P_c^{d-1}(1 - P_c)$, where P_c is the self-transition probability. This topology is shown in figure 6.3a. However, the shape of the geometric distribution implies that the shorter the chord, the more likely it is. This obviously does not agree with our musical intuition that a chord duration distribution should peak at a duration (or multiple durations) higher than zero.

A first effort to improve the chord duration model was made by Mauch and Dixon (2008). They substituted each single-state chord model by a linear left-to-right model of three states with equal emission probabilities (a type A topology according to Russell and Cook (1987)), which is visualised in figure 6.3b. Such an approach is called an expanded state hidden Markov model. This same procedure has been used in their later publications (Mauch and Dixon, 2010b; Mauch, 2010) and by Khadkevich and Omologo (2009b). Unfortunately, they produce no experimental data from which the effect of the duration modelling alone can be derived. Such a direct comparison between systems with and without duplication has been made by Cho et al. (2010) though. The conclusion was that the performance gain is similar to that of smoothing the output with a mean filter of 3

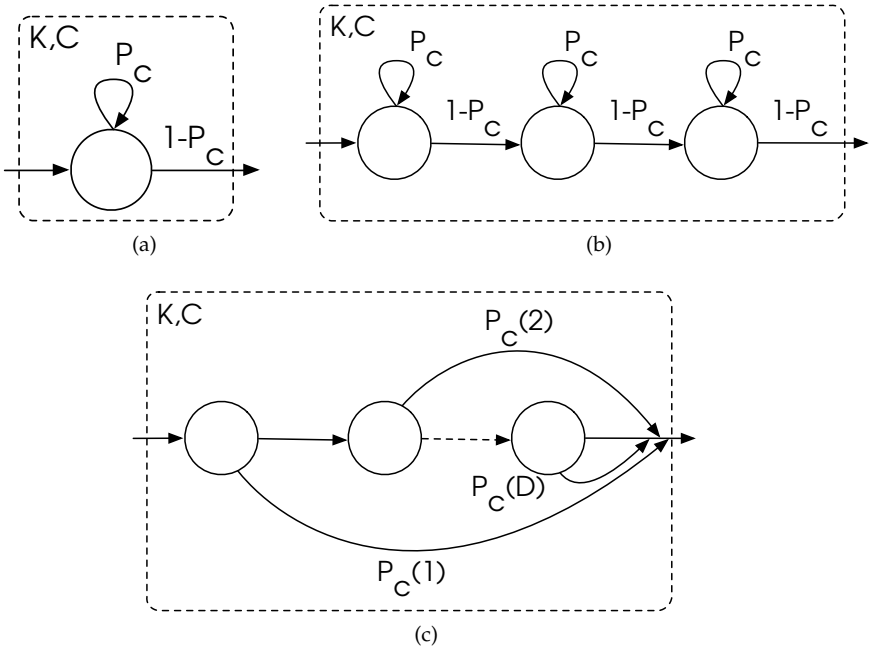


Figure 6.3: Expanded state topologies leading to different combined duration distributions

frames.

The reason for their small improvement is that they used a standard Viterbi algorithm for decoding, which gives rise to an overall state duration distribution of $P_{cd}(d) = 0, \forall d < 3$ and $P_{cd}(d) = P_c^{d-3}(1-P_c)^3, \forall d \geq 3$. The overall chord duration distribution is therefore a shifted and rescaled geometric distribution, which forces every chord to have a minimum duration of three frames, but that is the only difference compared to a single-state topology. The original intention of Mauch and Dixon (2008), however, was to construct an overall chord duration distribution that follows a negative binomial distribution. This distribution arises as the sum of independent geometric distributions. Decoding algorithms that estimate the optimal path by consider all paths through a state, such as pointwise maximum a posteriori (PMAP) decoding[1], do make this summation such that the topology in figure 6.3b effectively leads to an overall negative binomial duration distribution, but not the Viterbi algorithm. When the latter is used, all state self-transitions except one can be removed from the topology without affecting the outcome.

The property of the geometric distribution that makes it arise naturally in a state network, is that it is memoryless. The geometric probability decreases by P_c for every additional time step, regardless of the total number of time steps that have already passed. Therefore a state can have a constant self-transition probability and the total time spent in a state does not need to be recorded during the decoding. In order to change the prior state duration distribution to a distribution without the memorylessness property, the state self-transition probability needs to be a function of the time already spent in the state. One way to look at this, is that self-transition probabilities are removed from the network and that states emit observation sequences of a variable length. An explicit duration distribution $P_{cd}(d)$ then determines the duration probability associated with a state sequence length d, giving *explicit duration hidden Markov models (EDHMMs)* (Yu, 2010). Duration probabilities can only be calculated when a state is exited, i.e. a chord has ended in our case. For every possible duration, we need bookkeeping of previous states, which strongly increases the required computation time and memory. If we want to allow the possibility of a chord that never changes, this means that we should keep track of all probabilities and optimal paths all the way from the beginning of the music piece. Therefore, chords are in practice limited to a maximal duration of D time steps, after which a chord change is forced. An equivalent way of achieving general duration distributions, is to replace each single-state model with a chain of D states as presented in figure 6.3c, called a "Fergusson topology" by Russell and Cook (1987). The number of states in

[1]This decoding algorithm is also called posterior decoding or maximum gamma decoding, amongst many others. For a complete list of alternative names and more decoding algorithms, see Lember and Koloydenko (2014).

the network then strongly increases and the duration probabilities are determined by the different transitions that leave from the chain of expanded states.

Decoding EDHMMs requires a corresponding extended version of the Viterbi algorithm. In comparison to the regular Viterbi algorithm, the chord duration $P_{cd}(d)$ is now separated from the state change probabilities

$$P(k_n, c_n | k_{n-1}, c_{n-1}, c_n \neq c_{n-1})$$

In the expanded state interpretation, the state transitions within the same key-chord combination are handled differently from the transitions between key-chord combinations. The duration of a completed key-chord candidate gets evaluated together with all acoustic probabilities $P(\mathbf{x_v}|k_v, c_v)$ produced during the duration d of the chord (in a key) and the previous optimal path δ_{n-d} that *started* the key-chord d time steps ago. The search for the optimal path that *ends* the candidate key-chord does not only need to be performed over all previous key-chords, as is the case for the regular Viterbi algorithm, but additionally also over all allowed chord durations (up to D time steps). The exact formulation of the extended Viterbi algorithm can be found below. For comparison, the regular Viterbi algorithm is described in equation (5.15) on page 101.

$$\begin{aligned}
&\forall K_2 \in \mathbf{K}, \forall C_2 \in \mathbf{C}: \log_{10} \delta_n (k_n = K_2, c_n = C_2) = \\
&\max_{1 \leq d \leq D} \Bigg\{ \log_{10} P_{cd}(d) + \sum_{v=n-d+1}^{n} \log_{10} P(\mathbf{x_v}|k_v = K_2, c_v = C_2) \\
&+ \max_{\substack{K_1 \in \mathbf{K} \\ C_1 \in \mathbf{C} \setminus \{C_2\}}} \Big\{ \log_{10} \delta_{n-d} (k_{n-1} = K_1, c_n = C_1) \\
&\qquad + \log_{10} P(k_n = K_2, c_n = C_2 | k_{n-1} = K_1, c_{n-1} = C_1) \Big\} \Bigg\}
\end{aligned} \tag{6.1}$$

6.2.2 Using negative binomial prior chord distributions

EDHMMs can be used with arbitrary distributions, for example a duration histogram retrieved from a data set. However, in order to avoid a significant increase in the number of parameters, we stick to a fixed parametric distribution. We specifically focus on the negative binomial distribution, a family of distributions described by two parameters. The geometric distribution also belongs to this family. The negative binomial distribution has some appealing properties for our use. First and foremost, it can take a shape that approaches a typical duration distribution more closely than a

geometric distribution. Secondly, it allows us to apply a pruning step in the decoding which permits a significant reduction in computation time.

The probability mass function $f_{NB}(\tau, \rho, \varphi)$ of the negative binomial distribution[2] is

$$f_{NB}(\tau, \rho, \varphi) \equiv \binom{\tau + \rho - 1}{\tau} (1 - \varphi)^{\rho} \varphi^{\tau}, \forall \tau \geq 0 \tag{6.2}$$

where ρ is a shape parameter and φ a scaling parameter. For our application, it gives the probability that a chord lasts $d = \tau + 1$ time steps, corresponding to $(\tau + 1)\, T_x$ seconds,

As can be seen, the special case of $\rho = 1$ is equal to the geometric distribution. In order to test the influence of the duration distribution shape in isolation, we keep the shape parameter ρ as a controlling variable, but use the marginal duration distribution for a given φ, such that $P_{cd}(d, \rho) = f_{NB}(d - 1, \rho; \varphi)$. We derive φ such that the expected value of $f_{NB}(d - 1, \rho; \varphi)$ is equal to the mean chord duration $\overline{d_c}$. Just like in section 5.1.2.2, we consider a duration of 1.76 seconds for each relative chord. The mean of the negative binomial distribution is $\frac{\rho\varphi}{1 - \varphi}$, which gives us the following expression for φ, an extension of the geometric distribution case in equation (5.21):

$$\frac{\rho\varphi}{1 - \varphi} = \frac{\overline{d_c}}{T_x} - 1 \iff \varphi = \frac{\overline{d_c} - T_x}{\overline{d_c} + (\rho - 1)\, T_x} \tag{6.3}$$

where T_x is still the observations shift. In figure 6.4, the negative binomial distribution is depicted for a couple of values for ρ. We can see that its shape morphs from the geometric distribution into one that has a single peak which moves towards the mean.

Our usage of a parametric distribution is the main difference with the work of Chen et al. (2012), who performed experiments with an EDHMM for a chord-only estimation system. They examined a strongly data driven system which includes training an arbitrary global distribution on their data set through cross-validation. We, on the other hand, are more interested in a knowledge-based approach that examines the effect of letting the chord duration shape morph from a geometric distribution to one that fits the data better, without increasing the number of parameters.

[2] A number of equivalent definitions for the negative binomial distribution exist, that differ in their support domain and parameter interpretation. The variant we use arises as the probability that we need to observe τ successes in a sequence of independent and identically distributed Bernoulli trials with success probability φ, before ρ failures occur.

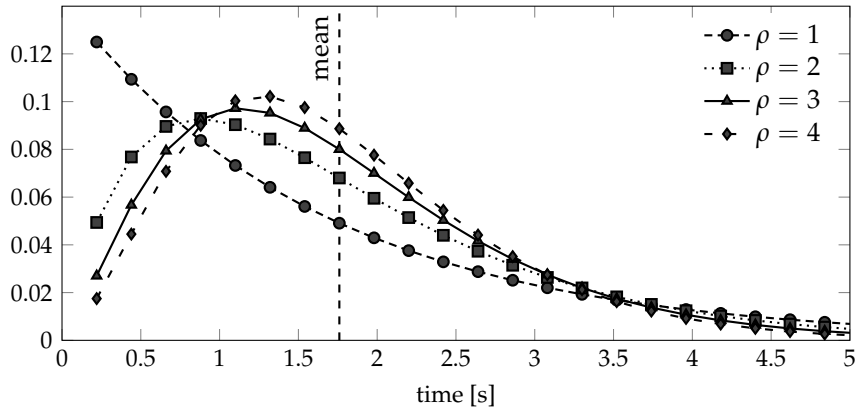


Figure 6.4: The class of negative binomial distributions

6.2.3 Optimising EDHMM decoding through path pruning

The decoding of an EDHMM can be significantly sped up if its duration distribution is *logarithmically convex*, meaning that the second derivative of its logarithm is strictly non-positive:

$$(\log P_{cd}(d))'' \leq 0, \forall d \tag{6.4}$$

In the case of the negative binomial distribution,

$$(\log f_{NB}(\tau, \rho, \varphi))'' \leq 0 \iff \varphi > \frac{\rho}{\rho^2} \tag{6.5}$$

For our choice of φ, this translates to the requirement $\overline{d_c} > 3T_x$, which is true for our configuration with $\overline{d_c} = 1.76$ and $T_x = 0.220$. We therefore can apply the pruning step proposed by Bonafonte et al. (1993) to reduce the search space of the modified Viterbi algorithm. Briefly summarised, it puts an adaptive upper limit on the state durations that need to be considered as candidates for the optimal path based on the previous optimal path. The pruning theorem says[3] that if the best path that exits a state at time n has entered the state at time η, then the best path that exits the same state at time $n+1$ will have entered the state no sooner than η. Instead of considering all state durations up to D to calculate $\delta_{n+1}(k_{n+1} = K_2, c_{n+1} = C_2)$ in equation (6.1), only the durations up to $n + 1 - \eta$ need to be investigated. This threshold time η is a function of (K_2, C_2) and depends on the

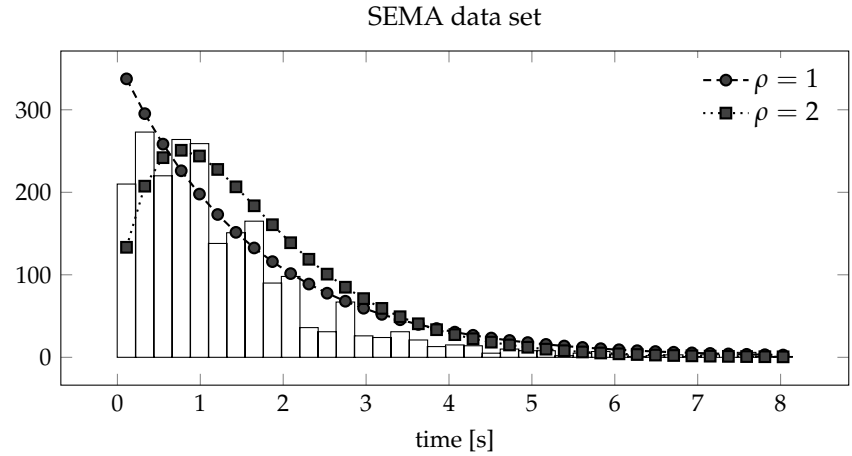
[3]Simplified version: the complete theorem also takes into account from which previous state the optimal path enters the last state to perform extra pruning, but this simplified version accounts for the majority of the speedup.

data, therefore no expected speedup can be calculated beforehand, nor can a minimum be guaranteed at all. The original authors report that they were able to reduce the required computation time to 3.2 times that of the standard Viterbi algorithm with geometric distribution. As a reference, decoding an EDHMM without pruning takes D times longer. In our experiments, the reduction was at least as significant, even going down to two times the standard case, although at least part of this is because we put far less effort into optimising our baseline HMM implementation. Even more important is that the factor two turns out to be almost independent of D, such that the trade-off between speed and accuracy is no longer an issue. The maximal state duration D now only influences the memory requirements, which are so modest that we can invariably set D to the entire number of frames in the music piece.

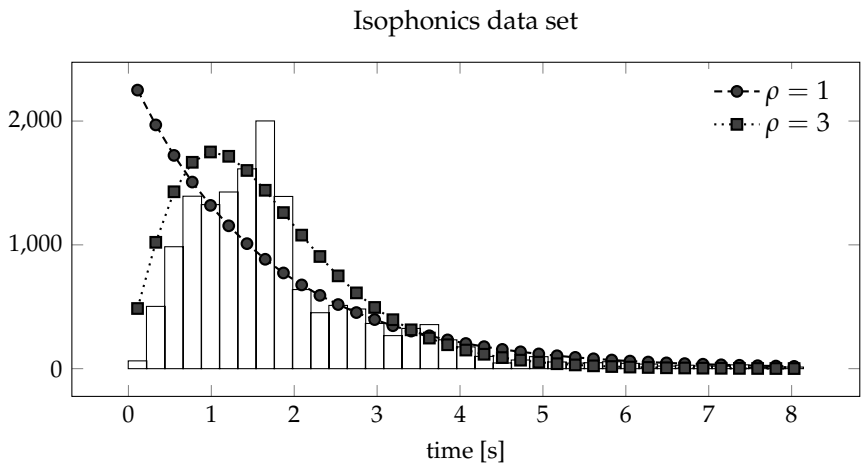
6.2.4 Experimental evaluation

Before we discuss our experimental results, we first take a look at the observed duration distributions in both the SEMA and Isophonics data sets. In figure 6.5, histograms of the measured chord durations are displayed together with the geometric distribution and the best-fitting negative binomial distribution for ρ between 2 and 7 (respectively $\rho = 2$ and $\rho = 3$). The fit between histogram and negative binomial distribution was calculated by means of the symmetric Kullback-Leibler divergence. We can see that the duration distributions differ in more than just their means (respectively 1.71 s and 2.15 s). The chord durations of the SEMA set follow the geometric distributions fairly closely, whereas those of the Isophonics set do not. Changing the prior chord duration distribution for the SEMA set from a geometric to a negative binomial duration distribution with $\rho = 2$ appears to improve the fit for the shortest chord durations a bit, but diminishes the match for chords between one and three seconds. This ambivance is also apparent in their divergence values, which are very similar with a slight advantage for $\rho = 1$ over $\rho = 2$. For the Isophonics data, a negative binomial chord distribution with $\rho = 3$ is able to model the peak in chord duration between one and two seconds. The improvements in chord estimation results will therefore likely be more significant than for the SEMA set.

We conducted an experiment in which ρ was changed from 1 (the geometric model baseline) to 7. For each value, the system ran with the four best chord change model configurations from section 5.2.4.1: uniform and best-chord SEMA, Isophonics and 9GDB. The corresponding estimation results on the SEMA and the Isophonics set are displayed in figure 6.6 and figure 6.7. The chord results show consistent trends over all configurations. As we predicted from the measured chord durations, the chord estimation performance on the SEMA set decreases with increasing ρ. The results on

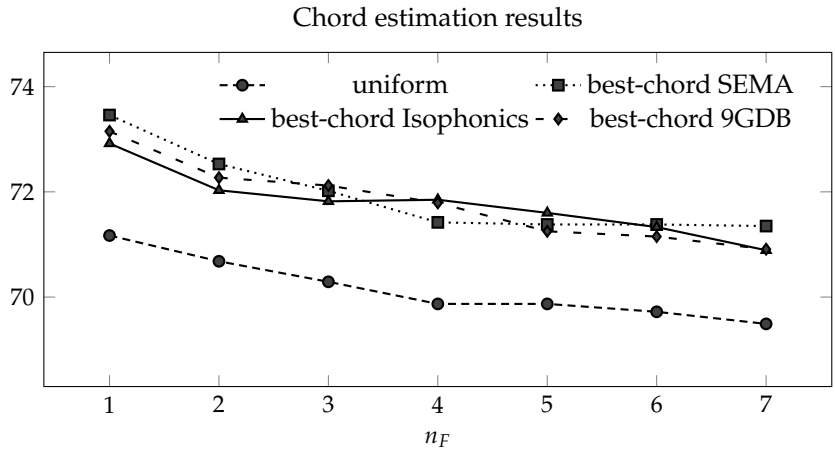


(a)

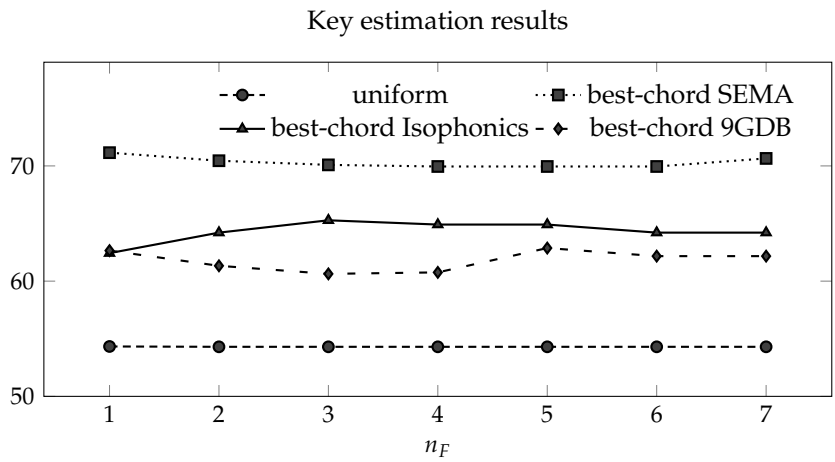


(b)

Figure 6.5: Chord duration histograms with geometric and best-fitting negative binomial distributions

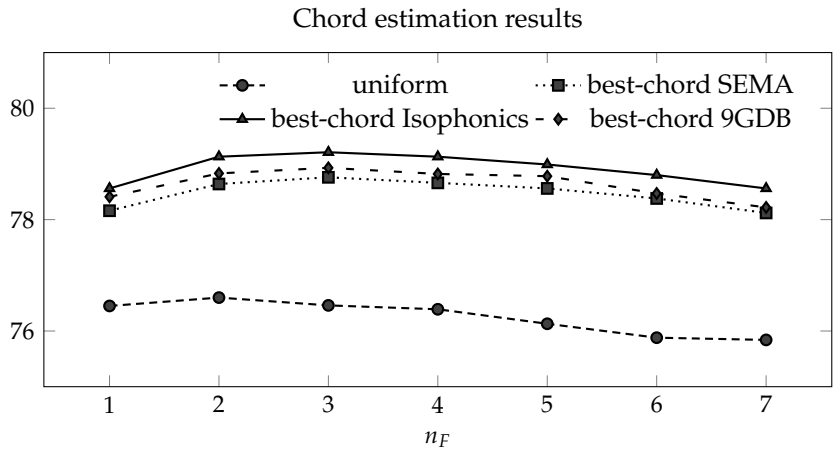


(a)

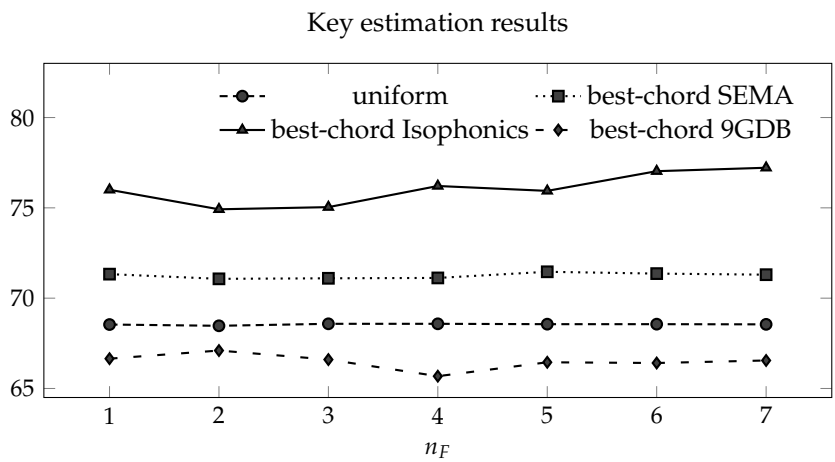


(b)

Figure 6.6: Influence of the chord duration distribution shape on SEMA data set



(a)



(b)

Figure 6.7: The evolution of key and chord results in function of a changing chord duration model. Each negative binomial shape was tested with four chord change model configurations on the Isophonics data set. The baseline geometric duration model is shown as $\rho = 1$.

the Isophonics set however, exhibit an increase up to $\rho = 3$, which corresponds to the best fit of the negative binomial distribution to the measured chord durations. Although the absolute increase at that point is only 0.6 percentage points, it is highly significant with $p < 10^{-4}$. For the key estimation, the effect of changing the duration model shape is unsystematic. The effect is therefore likely limited, but as we explained before, because key labels span a longer time, a single random change immediately has a visible effect the numerical results.

The final conclusion is that it is not recommended to swap in the negative binomial model at all times, because the results on the SEMA set show that in case of a mismatch with the data set, the results can degrade below those when a geometric model is used. However, if a rough estimate of the chord duration histogram of the data under test can be obtained, the Kullback-Leibler divergence can determine the best duration model. Although Chen et al. (2012) did not observe an increase in chord estimation when using a single, trained EDHMM, they were were able to improve their score slightly by training multiple systems on different chord duration distributions and then selecting the system that maximised the probability of the observations as the final feature decoder. Despite the fact their system operates under completely different assumptions (beat-synchronised features where the hypothesis is that the difference in duration distribution stems from different time signatures), this approach could be tested in future research on our own system as a possible solution for selecting the most appropriate chord duration model.

6.3 Conclusion

In this chapter, we performed a number of experiments that provide us a deeper insight into the influence of chord duration modelling on an automatic chord and key estimation system. These findings have also been used to improve the estimation performance, where applicable. We started by making the mean chord duration dependent on the relative chord instead of fixing it globally. This improved the key estimation performance and turned out to scale well to unseen data. Next, we experimented with alternative shapes for the chord duration distribution. We found that a better fitting distribution has the potential to slightly improve the chord estimation. However, the chord duration distribution can vary significantly from data set to data set. Therefore the risk of introducing a performance degrading mismatch is present when more specific distributions are used in the system. Finally, we established that the actual influence of the shape of the distribution used for chord duration modelling is rather limited.

7 Increasing the scope of the musicological context

After having delved deeper into the duration modelling part of our simultaneous key-chord estimation system, we now investigate a further extension of the context modelling in our system. Specifically, we propose a new system that maximally reuses components from the previous one, but which employs trigram change models, as described in Chapter 4, instead of bigram change models. There we saw that increasing the scope of the musicological context leads to models that are theoretically more powerful in predicting future chords, but that the generalisation to unseen data is weak. Now we will verify whether these models can help to increase chord and key estimation performance. We first explain the modifications we need to make in order to incorporate trigram models. Then we detail the supplementary steps we can take to mitigate the increase in computational requirements that come with these modifications. Finally, we assess the results of the new trigram system.

7.1 A probabilistic trigram framework

In order to integrate the additional information of a larger musicological context into our system, we need to revisit some of the simplifications that we made in Chapter 5. Remember that we are searching for the most likely sequence of states $\hat{Q} = \{\hat{q}_1, \ldots, \hat{q}_N\}$ for the acoustic observation sequence $\mathbf{X}$ of length N_x. We then established that this would be computed by maximising the product of two terms: the acoustic likelihood $P(\mathbf{X}|Q)$ and the prior probability $P(Q)$.

$$\hat{Q} = \arg\max_{Q} P(Q)P(\mathbf{X}|Q) \tag{7.1}$$

In this chapter, we keep the decomposition of the acoustic likelihood in separate key and chord terms as proposed in (5.3), but we reconsider the simplification of the prior probability in (5.4). We no longer make the first order Markov assumption, but the second order Markov assumption instead:

$$P(Q) = \prod_{n=1}^{N_x} P\left(q_n | q_{n-1}, q_{p(n)}\right) \tag{7.2}$$

This means that we assume that the transitions to the current state hypothesis q_n do not only depend on the immediately preceding state q_{n-1}, but also on the previous distinct state $q_{p(n)}$. Note that our index n again enumerates the feature segments. The previous distinct state of q_{n-1} is therefore not necessarily at $n-2$. We define $p(n)$ as the time index where the preceding distinct state ends: $\exists p\,(n) : q_{p(n)} \neq q_{n-1} \wedge q_{p(n)+1} = q_{n-1}$. A state still consists of the combination of a key and a chord, $q_n = (k_n, c_n)$, so the equivalent notation is

$$P(K, C) = \prod_{n=1}^{N_x} P(k_n, c_n | k_{n-1}, c_{n-1}, k_{p(n)}, c_{p(n)}, c_{n-1} \neq c_{p(n)}) \tag{7.3}$$

We keep the requirement that key changes can only occur together with chord changes. That is why $q_{n-1} \neq q_{p(n)}$ can be simplified to $c_{n-1} \neq c_{p(n)}$. Note that $k_{p(n)}$ represents the corresponding key of the previous chord $c_{p(n)}$ and not the previous key, so there is no constraint for k_n with respect to $k_{p(n)}$.

Just like in Chapter 5, we want to decompose the prior transition probability into four different submodels, a duration and a change model for chords and keys each, that are combined in different ways according to one of three cases. These cases are a simultaneous key and chord change, a chord change without key change, and chord and key invariance. To arrive at a formal decomposition into these terms, we follow a derivation similar to the one of the bigram system. Once more, we make the distinction between self-transitions ($k_n = k_{n-1} \wedge c_n = c_{n-1}$) and transitions to another state ($c_n \neq c_{n-1}$) and we use the law of total probability to separate those terms:

$$P\left(k_n, c_n | k_{n-1}, c_{n-1}, k_{p(n)}, c_{p(n)}, c_{n-1} \neq c_{p(n)}\right) =$$
$$\begin{cases} \underbrace{P\left(k_n = k_{n-1}, c_n = c_{n-1} | k_{n-1}, c_{n-1}, k_{p(n)}, c_{p(n)}, c_{n-1} \neq c_{p(n)}\right)}_{\text{trigram self-transition probabilities}} \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{if } k_n = k_{n-1} \wedge c_n = c_{n-1} \\ P\left(c_n \neq c_{n-1} | k_{n-1}, c_{n-1}, k_{p(n)}, c_{p(n)}, c_{n-1} \neq c_{p(n)}\right) \\ P\left(k_n, c_n | k_{n-1}, c_{n-1}, k_{p(n)}, c_{p(n)}, c_n \neq c_{n-1} \neq c_{p(n)}\right) \text{ if } c_n \neq c_{n-1} \end{cases} \tag{7.4}$$

Following a similar decomposition as we did in (5.7) and making use again of the requirement that key changes can only take place together with chord changes, the trigram self-transition probabilities also get reduced to *chord duration probabilities*

$$P\left(c_n = c_{n-1} | k_{n-1}, c_{n-1}, k_{p(n)}, c_{p(n)}, c_{n-1} \neq c_{p(n)}\right)$$

By assuming that this term does not depend on the preceding chord $c_{p(n)}$ and its key $k_{p(n)}$, it gets reduced to the same chord duration probabilities $P\left(c_n = c_{n-1} | k_{n-1}, c_{n-1}\right)$ as in the bigram system. This means that we can reuse the global or specific chord duration models proposed in the previous chapters to calculate these chord duration probabilities.

For the state-changing probabilities, we further decompose the joint probabilities and apply the above simplification of the chord duration probabilities:

$$\begin{aligned} &P\left(c_n \neq c_{n-1} | k_{n-1}, c_{n-1}, k_{p(n)}, c_{p(n)}, c_{n-1} \neq c_{p(n)}\right) \\ &P\left(k_n, c_n | k_{n-1}, c_{n-1}, k_{p(n)}, c_{p(n)}, c_n \neq c_{n-1} \neq c_{p(n)}\right) = \\ &\qquad \left(1 - \underbrace{P\left(c_n = c_{n-1} | k_{n-1}, c_{n-1}\right)}_{\text{bigram chord duration probabilities}}\right) \\ &\qquad \underbrace{P\left(c_n | k_{n-1}, c_{n-1}, k_{p(n)}, c_{p(n)}, c_n \neq c_{n-1} \neq c_{p(n)}\right)}_{\text{trigram chord change probabilities}} \\ &\qquad \underbrace{P\left(k_n | c_n, k_{n-1}, c_{n-1}, k_{p(n)}, c_{p(n)}, c_n \neq c_{n-1} \neq c_{p(n)}\right)}_{\text{trigram key transition probabilities}} \end{aligned} \tag{7.5}$$

The term $P\left(c_n | k_{n-1}, c_{n-1}, k_{p(n)}, c_{p(n)}, c_n \neq c_{n-1} \neq c_{p(n)}\right)$ in this expression are the trigram *chord change probabilities*, which we have encountered in Chapter 4 along with a number of options f_{cc3} to model them.

We then want to split the trigram key transition probabilities into duration and change probabilities, just like we did with the chord transition probabilities. This leads us to a final decomposition using the law of total probability:

$$P\left(k_n|c_n,k_{n-1},c_{n-1},k_{p(n)},c_{p(n)},c_n \neq c_{n-1} \neq c_{p(n)}\right) =$$
$$\begin{cases} \underbrace{P\left(k_n = k_{n-1}|c_n,k_{n-1},c_{n-1},k_{p(n)},c_{p(n)},c_n \neq c_{n-1} \neq c_{p(n)}\right)}_{\text{trigram key duration probabilities}} \\ \qquad\qquad\qquad\qquad \text{if } k_n = k_{n-1} \wedge c_n \neq c_{n-1} \neq c_{p(n)} \\ \left(1 - \underbrace{P\left(k_n = k_{n-1}|c_n,k_{n-1},c_{n-1},k_{p(n)},c_{p(n)},c_n \neq c_{n-1} \neq c_{p(n)}\right)}_{\text{trigram key duration probabilities}}\right) \\ \underbrace{P\left(k_n|c_n,k_{n-1},c_{n-1},k_{p(n)},c_{p(n)},c_n \neq c_{n-1} \neq c_{p(n)},k_n \neq k_{n-1}\right)}_{\text{trigram key change probabilities}} \\ \qquad\qquad\qquad\qquad \text{if } k_n \neq k_{n-1} \wedge c_n \neq c_{n-1} \neq c_{p(n)} \end{cases} \tag{7.6}$$

Similarly to the chord duration probabilities, we assume that the *trigram key duration probabilities* are independent of the preceding chord $c_{p(n)}$ and its key $k_{p(n)}$, such that they get reduced to the bigram key duration probabilities $P\left(k_n = k_{n-1}|c_n,k_{n-1},c_{n-1},c_n \neq c_{n-1}\right)$. The *trigram key change probabilities* are calculated by means of the f_{kc3} models proposed in Chapter 4.

In summary, just as for the bigram case, the entire prior probability can be broken down into four different cases that are calculated as the compound of up to four different probabilities: duration and change probabilities for both keys and chords. We simplified both duration probabilities by assuming that the value (not duration) of the preceding label influences the determination of the current label duration only indirectly. The calculation of the chord duration probabilities P_{cd} and key duration probabilities P_{kd} can therefore reuse the bigram models, as do the chord acoustic and key acoustic probabilities (P_{ca} and P_{ka}). Only the chord change probabilities P_{cc3} and key change probabilities P_{kc3} are changed from bigram to trigram. Their influence on the estimation results is what will be investigated in the remainder of this chapter. The way the different submodels work together to calculate the complete prior knowledge probability closely mirrors the bigram case, seen in (5.14), including the introduction of the balance factors β_c and β_k such that the relative importance of change models can be con-

trolled:

$$P\left(k_n, c_n | k_{n-1}, c_{n-1}, k_{p(n)}, c_{p(n)}\right) =$$
$$\begin{cases} 0 & \text{if } k_n \neq k_{n-1} \wedge c_n = c_{n-1} \vee c_{n-1} = c_{p(n)} \\ P_{cd} & \text{if } k_n = k_{n-1} \wedge c_n = c_{n-1} \wedge c_{n-1} \neq c_{p(n)} \\ (1 - P_{cd})\, P_{cc3}^{\beta_c} P_{kd} & \text{if } k_n = k_{n-1} \wedge c_n \neq c_{n-1} \wedge c_{n-1} \neq c_{p(n)} \\ (1 - P_{cd})\, P_{cc3}^{\beta_c} (1 - P_{kd})\, P_{kc3}^{\beta_k} & \text{if } k_n \neq k_{n-1} \wedge c_n \neq c_{n-1} \wedge c_{n-1} \neq c_{p(n)} \end{cases} \tag{7.7}$$

7.2 Exploiting musicological constraints to reduce computational requirements

Conceptually, the move from bigram to trigram prior information is not very complicated. In the previous sections, we went over the required changes and how we can maximally reuse components from the bigram system. The actual challenge lies in finding a method that can decode the optimal path in a feasible way, because the required changes bring along an exponential increase in the search space that needs to be considered. If N_k is the number of keys and N_c the number of chords, then the number of states in the bigram system is equal to the number of distinct key-chord combinations, $N_k N_c$ and maximally $N_k^2 N_c^2$ outgoing transitions have to be to tested per time step, because every key-chord combination can be reached from a previous key-chord combination. In the trigram case, we need to consider maximally $N_k^2 N_c^2$ states ($N_k N_c\,(N_k N_c - 1)$ to be exact because the previous state needs to be different) and $N_k N_c\,(N_k N_c - 1)^2$ outgoing transitions at every time step (because the states are no longer fully connected). The corresponding Viterbi algorithm to find $\delta_n\,(K_3, C_3, K_2, C_2)$, the probability of the best path arriving in (K_3, C_3, K_2, C_2) at time n, is now given by

$$\begin{aligned} &\forall K_2, K_3 \in \mathbf{K}, \forall C_2, C_3 \in \mathbf{C}: \\ &\delta_n\,(k_n = K_3, c_n = C_3, k_{n-1} = K_2, c_{n-1} = C_2) = \\ &\quad P\,(\mathbf{x_n} | k_n = K_3, c_n = C_3) \\ &\quad \max_{\substack{K_1 \in \mathbf{K} \\ C_1 \in \mathbf{C} \\ (K_1, C_1) \neq (K_2, C_2)}} \Big\{ \delta_{n-1}\left(k_{n-1} = K_2, c_{n-1} = C_2, k_{p(n)} = K_1, c_{p(n)} = C_1\right) \\ &\quad P\left(k_n = K_3, c_n = C_3 | k_{n-1} = K_2, c_{n-1} = C_2, k_{p(n)} = K_1, c_{p(n)} = C_1\right) \Big\} \end{aligned} \tag{7.8}$$

This Viterbi algorithm is generally applicable to any trigram based system, but there are properties of the specific system we derived in the previous section that can be exploited to reduce the computational requirements. The first is that key changes can only occur together with chord changes. This is already incorporated in (7.7) and it reduces the inherent trigram condition of distinct consecutive states to distinct consecutive chords, such there are only $N_k^2 N_c \left(N_c - 1\right)$ eligible combinations in (7.8) instead of $N_k N_c \left(N_k N_c - 1\right)$.

A further reduction in the number of transitions to examine can be achieved specifically for the key change models f_{kc3}, which we have proposed in Chapter 4. Recall that the proposed models have in common that they are essentially of bigram type, but with supplementary conditions imposed on their applicability. Concretely, the trigram key change model is used to forbid key labels that last only for the duration of a single chord. We can directly integrate these validity conditions into the decomposition of the prior musicological model into four distinct cases:

$$
\begin{aligned}
&P\left(k_n, c_n | k_{n-1}, c_{n-1}, k_{p(n)}, c_{p(n)}\right) = \\
&\begin{cases}
0 & \text{if } k_n \neq k_{n-1} \wedge \left(c_n = c_{n-1} \vee k_{n-1} \neq k_{p(n)}\right) \vee c_{n-1} = c_{p(n)} \\
f_{cd} & \text{if } k_n = k_{n-1} \wedge c_n = c_{n-1} \wedge c_{n-1} \neq c_{p(n)} \\
\left(1 - f_{cd}\right) f_{cc3}^{\beta_c} f_{kd} & \text{if } k_n = k_{n-1} \wedge c_n \neq c_{n-1} \wedge c_{n-1} \neq c_{p(n)} \\
\left(1 - f_{cd}\right) f_{cc3}^{\beta_c} \left(1 - f_{kd}\right) f_{kc2}^{\beta_k} & \\
& \text{if } k_n \neq k_{n-1} \wedge c_n \neq c_{n-1} \wedge k_{n-1} = k_{p(n)} \wedge c_{n-1} \neq c_{p(n)}
\end{cases}
\end{aligned}
\tag{7.9}
$$

The benefit of integrating the trigram key change constraints directly into the prior probability decomposition is that the number of transitions falling under the case of probability zero now form the majority. The four cases in which the prior probability is decomposed are far from uniformly distributed. This is illustrated by listing all possible pairwise combinations of the key and chord variables $k_{p(n)}, k_{n-1}, k_n$ and $c_{p(n)}, c_{n-1}, c_n$ in Table 7.1 on page 147. Only five combinations lead to a non-zero probability. A textual description of those five combinations is added together with the amount of transitions per time step they represent. The latter one is derived from the fact that the first key-chord combination $\left(k_{p(n)}, c_{(n)}\right)$ can be chosen freely and any subsequent equality between key or chord pairs happens once, whereas an inequality accounts for the other $N_k - 1$, respectively $N_c - 1$ times. To make these numbers a bit more concrete, we have listed them as the proportion of the total possible transitions they represent when $N_k = 24$ and $N_c = 48$, the configuration assumed throughout this thesis.

	pairwise key and chord combinations	description	number of transitions	proportion
allowed transitions	$k_{n-1} = k_{p(n)}, c_{n-1} \neq c_{p(n)}$ $k_n = k_{n-1}, c_n = c_{n-1}$	self transition after no key change	$N_k N_c (N_c - 1)$	0.004%
	$k_{n-1} \neq k_{p(n)}, c_{n-1} \neq c_{p(n)}$ $k_n = k_{n-1}, c_n = c_{n-1}$	self transition after key change	$N_k N_c (N_k - 1)(N_c - 1)$	0.081%
	$k_{n-1} = k_{p(n)}, c_{n-1} \neq c_{p(n)}$ $k_n = k_{n-1}, c_n \neq c_{n-1}$	chord change without key change after no key change	$N_k N_c (N_c - 1)^2$	0.166%
	$k_{n-1} \neq k_{p(n)}, c_{n-1} \neq c_{p(n)}$ $k_n = k_{n-1}, c_n \neq c_{n-1}$	chord change without key change after key change	$N_k N_c (N_k - 1)(N_c - 1)^2$	3.828%
	$k_{n-1} = k_{p(n)}, c_{n-1} \neq c_{p(n)}$ $k_n \neq k_{n-1}, c_n \neq c_{n-1}$	key and chord change (after no key change)	$N_k N_c (N_k - 1)(N_c - 1)^2$	3.828%
disallowed transitions	$k_{n-1} = k_{p(n)}, c_{n-1} = c_{p(n)}$ $k_n = k_{n-1}, c_n = c_{n-1}$	$\left(k_{p(n)}, c_{p(n)}\right)$ has to differ from (k_{n-1}, c_{n-1})	$N_k N_c$	0.000%
	$k_{n-1} = k_{p(n)}, c_{n-1} = c_{p(n)}$ $k_n \neq k_{n-1}, c_n = c_{n-1}$	$\left(k_{p(n)}, c_{p(n)}\right)$ has to differ from (k_{n-1}, c_{n-1})	$N_k N_c (N_k - 1)$	0.002%
	$k_{n-1} = k_{p(n)}, c_{n-1} = c_{p(n)}$ $k_n = k_{n-1}, c_n \neq c_{n-1}$	$\left(k_{p(n)}, c_{p(n)}\right)$ has to differ from (k_{n-1}, c_{n-1})	$N_k N_c (N_c - 1)$	0.004%
	$k_{n-1} = k_{p(n)}, c_{n-1} = c_{p(n)}$ $k_n \neq k_{n-1}, c_n \neq c_{n-1}$	$\left(k_{p(n)}, c_{p(n)}\right)$ has to differ from (k_{n-1}, c_{n-1})	$N_k N_c (N_k - 1)(N_c - 1)$	0.081%
	$k_{n-1} \neq k_{p(n)}, c_{n-1} = c_{p(n)}$ $k_n = k_{n-1}, c_n = c_{n-1}$	key change without chord change not allowed	$N_k N_c (N_k - 1)$	0.002%
	$k_{n-1} \neq k_{p(n)}, c_{n-1} = c_{p(n)}$ $k_n \neq k_{n-1}, c_n = c_{n-1}$	key change without chord change not allowed	$N_k N_c (N_k - 1)^2$	0.040%
	$k_{n-1} \neq k_{p(n)}, c_{n-1} = c_{p(n)}$ $k_n = k_{n-1}, c_n \neq c_{n-1}$	key change without chord change not allowed	$N_k N_c (N_k - 1)(N_c - 1)$	0.081%
	$k_{n-1} \neq k_{p(n)}, c_{n-1} = c_{p(n)}$ $k_n \neq k_{n-1}, c_n \neq c_{n-1}$	key change without chord change not allowed	$N_k N_c (N_k - 1)^2 (N_c - 1)$	1.873%
	$k_{n-1} = k_{p(n)}, c_{n-1} \neq c_{p(n)}$ $k_n \neq k_{n-1}, c_n = c_{n-1}$	key change without chord change not allowed	$N_k N_c (N_k - 1)(N_c - 1)$	0.081%
	$k_{n-1} \neq k_{p(n)}, c_{n-1} \neq c_{p(n)}$ $k_n \neq k_{n-1}, c_n = c_{n-1}$	key change without chord change not allowed	$N_k N_c (N_k - 1)^2 (N_c - 1)$	1.873%
	$k_{n-1} \neq k_{p(n)}, c_{n-1} \neq c_{p(n)}$ $k_n \neq k_{n-1}, c_n \neq c_{n-1}$	three key changes in a row not allowed	$N_k N_c (N_k - 1)^2 (N_c - 1)^2$	88.053%

Table 7.1: All pairwise combinations of key and chord variables in a trigram system with their proportion of occurrence

The remainder of the table contains all transitions that are disallowed according to (7.9), listed with the reason why. All possible combinations sum to $N_k^3 N_c^3$ possible transitions per time step. We can see that only 7.915% of the possible transitions can lead to a non-zero probability. Mostly, this stems from the rule that three key changes in a row are not allowed, which makes 88.053% of the transitions impossible. If we can implement a Viterbi algorithm in such a way that the 92.085% impossible transitions are not even calculated, a large speedup can be expected. However, the memory requirements stay the same. The number of valid trellis points per column stays equal, but the number of future points that can be reached is now variable. Even though the reduction of the number of transitions is significant (compared to an unrestricted trigram system), the increase in transitions with respect to a bigram system is still as large as

$$\frac{N_k N_c \left(N_c - 1\right) \left(2N_k N_c - N_k - N_c + 1\right)}{N_k^2 N_c \left(N_c - 1\right)} = 93.042$$

Incorporating the musicological constraints directly into the search procedure allows us to reduce the number of computations without compromising the theoretical optimality of the solution. To reduce the computational requirements even more, we can use a *beam search,* although the solution is then no longer guaranteed to be theoretically optimal. The underlying idea is that not every trellis point at time $n-1$ is considered as a candidate starting point for the transitions at time n, only those that are the most likely. We use as criterion to expand a starting point to all possible targets that its probability δ_{n-1} should be within a given distance, called the *beam width*, of the maximal δ_{n-1}. Through the combination of the techniques in this section, we managed to bring down the computational requirements of our implementation such that a music piece is processed in 431% of real time (for comparison, the bigram system takes 13 % of real time).

7.3 Experimental results

We tested the trigram system in combination with all trigram chord change models derived in Chapter 4. We also used a uniform chord change model such that we could test the effect of the trigram key change model in isolation. As before, we first used the SEMA data set to find the parameter combinations $(\alpha_c, \alpha_k, \beta_c)$ that optimise the chord performance and the key performance for each chord change model. The resulting seven configurations are then run on the Isophonics data set. The corresponding results for the SEMA and the Isophonics set can be found in Table 7.2a on page 151 and Table 7.2b on page 151. The differences with respect to their bigram

equivalents have been added in parenthesis, followed by the percentage of files in the data set whose output is changed compared to the bigram case.

The results once again show that key extraction is more sensitive to a chord change model than chord extraction. The chord results on the SEMA set generally display a slight increase in performance, but this is not statistically significant. The key result differences vary more in amplitude and in direction, but also turn out statistically not significant. The improvement of the key output predicted by the test set perplexities in Table 4.8 on page 90 fails to occur. Notable is the observation that the usage of trigram change models only influences the key and chord output in a restricted number of files, respectively 10–25 % and 50–70 %. The number of unique trigrams in the SEMA chord change model is so low (see Table 4.4 on page 83) that the low number of changed output files can be expected, as it means that the trigram model mostly backs off to the bigram probabilities anyway. However, using the more distinct 9GDB trigram chord change model does not necessarily result in more altered files.

Applying the trigram system to the Isophonics data set causes a greater number of files to change. At least one chord is altered in almost all files and a key is modified in 20–57 % of the pieces. After all, the songs in this set are longer such that the possibility of a modification increases and the fact that the configurations were optimised on the SEMA set likely causes more random variations in the Isophonics output. Apart from this, the general trend is the same: slight, not significant chord improvements and larger key fluctuations where only the optimal key configuration with the 9GDB model shows a significant improvement ($p < 0.03$). Analysing the error categories gives the same tendencies as for the bigram system, slightly intensified. A little more chords with one chroma wrong are corrected and the number of fifth shift chord errors increases a bit.

The only noticeable difference with the bigram system is that the complementary task suffers less when the system is configured either for optimal key or optimal chord output. The effect is that all chord change models improve both key and chord output in all configurations when compared to the uniform chord change model. The matched model obviously still leads to the biggest increase in performance. Whether this is due to the parameter configurations being more robust to data set changes or simply due to luck, or whether this means that trigram models are intrinsically better at capturing the key-chord relation cannot be concluded from these experiments however.

Finally, the identical performance of the uniform chord change model in comparison to the bigram system shows that the beam width is set sufficiently large and that the trigram *key* change model does not change the results. The latter could be expected, because the trigram key change model is identical to the bigram one with the additional condition that a key cannot last for the duration of just one chord. Because the bigram system never generated such unwanted output, the trigram model has no

influence. Similar to the bigram case, all variants f_{kc3a},f_{kc3b} and f_{kc3c} lead to the same results. This means that the diatonicity conditions imposed in f_{kc3c} could be directly integrated into (7.9) in order to achieve an additional speedup.

So far, other uses of higher order context models have been rarely reported in the literature.Cheng et al. (2008) use models up to tetragrams in a greedy-search chord-only decoder and Khadkevich and Omologo (2009b) rescore a bigram lattice with trigram or tetragram probabilities. Our search procedure is therefore more exhaustive in covering all trigram combinations. Khadkevich and Omologo (2009b) report that adding a trigram model to a baseline system comprising no contextual modeling at all yields a chord estimation improvement of less than two percentage points. However, substituting the trigram by a tetragram model does not offer any additional improvement. Surprisingly, the authors do not report any figures for a bigram model, while our experiments in the previous chapters show that this already increases the chord extraction performance by two percentage points. Another difference is that their system first estimates one global key before it starts to estimate chords. This means that the influence of higher order contextual modeling on key estimation was not investigated. Cheng et al. (2008) do report an additional increase in chord estimation of 3.5 percentage points when going from bigram to trigram models, but also see no further improvement when using tetragram models. However, their experiments are performed on a small data set of only 28 songs and they use absolute chords without keys.

7.4 Conclusion

In this chapter, we presented a system for automatic key and chord estimation that makes use of trigram context information. We formulated a new probabilistic framework in which the acoustic and duration models of the previous bigram system are reused. Only the key change and chord change models are different, such that we could separately test the effect of increasing the scope of the modelled context. The resulting trigram system needs significantly more computational power, however. We therefore explored some options to reduce this load, by integrating the desired musicological constraints directly into the search procedure and by searching for a close approximation of the optimal key and chord sequences instead of the theoretically optimal ones. We gather from the experimental evaluation that taking a trigram context into account has little impact on the generated keys and chords. The key output is altered in just a minority of the music pieces in comparison to the equivalent bigram system. The chord output changes in more than half of the files. Nonetheless, the key results display a larger numerical variation (up to five percentage points)

	optimal chord configuration		optimal key configuration	
	chord	key	chord	key
uniform	**71.17** (+0.00:0%)	54.33 (+0.00:0%)	71.17 (+0.00:0%)	**54.33** (+0.00:0%)
SEMA	**73.95** (+0.49:70%)	66.65 (-4.50:23%)	73.49 (+0.21:58%)	**71.64** (-0.21:10%)
Isophonics	**73.56** (+0.64:51%)	61.72 (-0.71:12%)	72.66 (+1.02:70%)	**69.47** (+1.57:23%)
9GDB	**73.23** (+0.08:58%)	63.01 (+0.36:15%)	73.02 (+0.55:62%)	**65.28** (-2.19:25%)

(a)

	optimal chord configuration		optimal key configuration	
	chord	key	chord	key
uniform	**76.44** (-0.01:1%)	68.38 (-0.17:4%)	76.44 (-0.01:1%)	**68.38** (-0.17:4%)
SEMA	78.24 (+0.08:91%)	**72.41** (+1.08:37%)	**78.27** (+0.09:96%)	71.86 (+4.99:57%)
Isophonics	79.31 (+0.75:97%)	71.76 (-4.24:50%)	**79.34** (+0.51:94%)	**78.68** (+2.01:36%)
9GDB	**78.41** (+0.00:94%)	70.12 (+3.46:20%)	78.01 (-0.36:96%)	**70.66** (+3.88:37%)

(b)

Table 7.2: Results of the optimal configurations with different chord change models on the SEMA set (a) and on the Isophonics set (b). The best results per chord change model are set in boldface. The differences with respect to the equivalent bigram configurations are displayed in parenthesis, followed by the percentage of music pieces whose output is changed.

than the chord results (less than one percentage point), but this is due to the inherently wider time span of keys. In general, both the differences in key and chord score are statistically not significant though. Therefore we conclude that extending an automatic key and chord estimation system with trigram context information only leads to a significant increase in the required computational power.

8 Conclusions and perspectives

This chapter summarises the work performed in this thesis and draws some general conclusions from it. We conclude with some directions for future research and some related work that uses automatically generated key and chord sequences.

8.1 Summary

In this thesis, we performed a thorough study of the role prior knowledge can play in the automatic estimation of keys and chords. We started by situating the field of automatic key and chord estimation and its applications. Then we rigorously defined the musical terminology of this text. A subsequent study of the available scientific literature showed that key and chord estimation can be done either by handling the two estimation tasks separately or by combining them in an integrated system. The motivation behind the latter approach is that it allows to take more musicological knowledge into account, as keys and chords are intrinsically linked and jointly reasoning about them resembles more closely the way scholars study harmony. However, so far handling both keys and chords together has not been demonstrated to effectively lead to better estimated output. Therefore a large part of the text has been dedicated to proposing ways to exploit this key-chord interdependence and to quantify their effect on the produced results. We chose to formulate our experiments in the framework of a hidden Markov model (HMM), both because it has been proven popular in the relevant literature and because it allows us to isolate the

musicological knowledge involved in a key and chord estimation system. This musicological knowledge covers a durational and a contextual aspect which have to be incorporated into the feature decoder, an HMM in our case. The decoder converts the stream of features produced during a feature extraction phase into a sequence of key and chord labels.

We first concentrated on modelling musicological context as the combination of a key change model and a relative chord change model. We explained the reasoning behind this decomposition and went over a number of options for calculating both models, based on either an annotated corpus or musicological theory as the knowledge source. Then we used information theoretical metrics to quantify them. The bigram change models were then used in combination with a simple chord duration model and a key duration model to build a simultaneous estimation system. The complexity of this system was gradually increased, such that the influence of each of its components could be measured. We began by testing the acoustic component and its relation to the features and we subsequently introduced different duration models, chord change models and key change models. We then went on to propose a chord duration models that is more musically informed. Finally, trigram models that capture more of the context are tested as replacements for the bigram models.

8.2 Conclusions

In general, we can conclude that musicological knowledge can be exploited to aid in estimating keys and chords from audio, but the gain it offers varies greatly between keys and chords. First of all, duration modelling is essential in order to achieve acceptable results, both for keys and for chords. This does not come as a surprise, as the duration model needs to stabilise the acoustic model whose output rate (equal to the observation rate) is much higher than the expected rate of the labels, particularly the key labels. On the other hand, the observation rate cannot be decreased without loosing temporal precision of the label changes. Adding more durational knowledge by differentiating expected chord duration on the basis of their role in a key, improves the key estimation results. This extra knowledge is universally applicable. Finally, adapting the chord duration model to better match the shape of the duration distribution in the data set leads to slightly better chord output, but requires that the actual distribution can be approximated in advance.

The influence of the chord change model on chord estimation differs considerably from its influence on key estimation. The inclusion of a bigram chord change model only leads to limited improvements in the chord results, but large improvements in the key results. Although the chord improvements are small, they are consistent across all proposed models.

A detailed error analysis showed that the reason is that a chord change model mostly corrects errors that are acoustically close, but harmonically far off. The probabilities involved in rejecting these chord confusion pairs form only a minor part of a chord change model and all proposed models capture this information sufficiently well. Incorporating any of them therefore has a beneficial result on chord estimation. For estimating keys on the other hand, the chord change model is of the greatest importance. A chord change model that fits the data well greatly improves the performance, but if there is a significant mismatch between model and data, the performance can become worse than in the case when no context information is used at all. Analysing the errors revealed that key estimation uses the full range of probabilities that constitute the chord change model, so the differences between the different models are more apparent in the resulting output. Because most of the key errors are confusions with related keys, the varying results for the different chord change model are not necessarily reflected in the chord results. Because related keys produce chords that are similar, an approximate key estimation is enough to reap the benefits of the chord change model on chord estimation. The key-chord coupling is therefore not very strong. Increasing the horizon of the context further by employing trigram models instead of bigram models only leads to small, statistically insignificant changes in both the key and chord output. Moreover, moving to trigram models requires a strong increase in computational power.

Finally, a key change model is not able to influence either key or chord estimation in a significant way. Including it leads to the same output as when it is left out, although the diatonicity constraints of some variants can be used to reduce the decoding search space.

8.3 Future research prospects

8.3.1 Extending the chord or key vocabulary

An obvious candidate for further extension of our system is the addition of more chord types and/or key modes to the estimation vocabulary. The bigger the vocabularies, the richer the produced output. The current vocabularies of four chord types and two modes allows to give a basic transcription of the harmony in a music piece, but a greater level of detail is necessary for some applications.

A vocabulary extension will bring along both opportunities and possible pitfalls. On the plus side, the more acoustic models are introduced, the less variation they need to account for. In the current situation, all chords based on the same triad are grouped together, but the additional

chromas in complex chords can result in chords that are harmonically similar, but acoustically different. We did not include these complex chords in our evaluation for precisely this reason, but this is not an option in real life applications, although the occurrence of such complex chords depends strongly on the music genre.

On the other hand, more acoustic models increase the possibility of confusing two labels, especially when some labels are subsets of others, e.g. when mixing triads and tetrads. The challenge for the context modelling will then be to deal with chord types that are different in form, but nevertheless have essentially the same harmonic function. This will definitely be the case when taking chord tensions into consideration. When only triads are used, the relation between harmonic form and harmonic function is relatively straightforward, i.e. swapping a chord for another leads to a significant difference in perception, so their probabilities should differ to reflect this. When complex chords are used however, some chords can be swapped for each other without significantly changing the chord sequence. For example, removing the *D* chroma in a *C9* chord (i.e. the 9th) gives a *C7* chord that is acoustically different, but this substitution does not significantly alter the harmony. Since the knowledge captured by the chord change model should be based on its harmonic function, the transition probabilities that involve such equality pairs (or tuples) should be equal too. If the model is learned from a data set, one of the equivalent chords might be favoured, based on its non-relevant higher frequency in the set. The risk of overfitting therefore increases. An option to overcome this would be to use chord change models that are truly based on harmonic function instead of full chord type by tying probabilities of such transitions. Determining the exact transitions to pair requires a whole new amount of musicological knowledge though. The non-functional aspects of a chord, such as its tensions, are in that case determined solely by the acoustic models.

8.3.2 Focussing on acoustic modelling and feature design

So far, we have used standard acoustic features and knowledge-based acoustic models in order to keep our study of the influence of prior musicological knowledge as unbiassed as possible. In our analysis of the chord errors, we saw that a significant proportion of errors is most likely introduced in the feature extraction stage. Musicological knowledge cannot be used to recover from these errors because they are also musically sensible. There will always remain some rare ambiguities in the acoustic signal that can only be resolved by musicological modelling, such as determining the root of augmented chords, but the current performance leaves plenty of room for improvement in acoustic modelling and feature design. Based on a

study of the literature and the MIREX results, the most promising directions seem to be to use multiple feature streams and data-driven acoustic modelling.

8.4 Further processing of estimated key and chord sequences

In spite of the limitations of the current system, leading to imperfect transcriptions in terms of a restricted vocabulary, its output can already be used as input for other processes. First of all, we mention some possible post-processing to unearth information that is already present in the generated key and chord sequences themselves, but not yet in an explicit manner. An example is the field of automatic functional analysis (Pachet, 2000; de Haas et al., 2013a; Keller et al., 2013), where the goal is to represent harmony in a way that is closer to a human interpretation. Currently, this is done as a post-processing step on the deterministic output, but as mentioned previously, probabilistically integrating such an analysis into the system could be a way to overcome the problems associated with increasing the chord vocabulary. Another example are pitch spelling algorithms (Meredith, 2004; Chew and Chen, 2005). This step is usually not included in estimation systems, because intermediate key and chord results are not helpful for determining the pitch spelling of the final output. If the resulting keys and chords are to be read by humans however, applying this post-processing is advised.

For most *direct* applications (see the terminology introduced chapter 1), the currently produced output is too restricted to be useful. When users want to learn a music piece, they want a transcription that is 100 % correct, not close to correct. In addition, the limited chord vocabulary becomes a problem for all but simple pop music. The same is true when large databases of estimated sequences need to be analysed for the intricacies of certain genres or composers, such as the specific patterns and rules in jazz music (Steedman, 1984; Gillick et al., 2010). Often, the most interesting parts for musicologists are the details and the exceptions, not the rough outline that automatic estimation is currently able to provide.

These limitations are a lot less important for *indirect* applications, where the generated keys and chord labels are used as input to achieve a higher objective. Some of the following applications were originally reported together with a specific implementation of a key or chord estimation system, others were presented using manually annotated labels, but the techniques they use are general enough to be applied in combination with any key and/or chord estimation system.

A first example of an indirect, small-scale application is the system for

chord-based genre classification proposed by Pérez-Sancho et al. (2009). Its underlying idea is that certain chords or chord sequences are indicative of a genre. Different symbolic chord representations were tested for their ability to classify music pieces into three categories, and further into eight subcategories. The representations varied in chord complexity (using only triads or also more complex chords), in the usage of absolute chords versus relative chords in a key or in the length of the chord subsequences that were considered. One of the conclusions was that a decrease in chord complexity can be compensated by the use of longer subsequences.

A similar effort has been undertaken by Cheng et al. (2008) for emotion classification. Low level spectral features were compared with features derived from chord sequences for assigning a positive or a negative valence to a music piece by means of a nearest neighbour classifier. These chord features were either chord histograms or the longest common chord sequences between the song query and each of the songs in the emotion annotated database.

Chord sequences have been used by Bello and Pickens (2005) and de Haas et al. (2013b) to derive a structural segmentation of music. In both papers, a way to find the boundaries of structural segments and to label similar segments with a same, non-semantic identifier (i.e. "A", "B" instead of "chorus", "verse") is presented. The approach of Bello and Pickens (2005) consists of clustering short-time histograms obtained by sliding an analysis window over an estimated chord sequence, whereas de Haas et al. (2013b) relies on a suffix-tree based algorithm to detect repetition in a chord sequence. Finally, harmony is also one of the controlling parameters for the visualiser of the interactive-listening web-service Songle[1] (Goto et al., 2011).

In the category of indirect, large-scale applications, the work of Hanna et al. (2009) and de Haas et al. (2013c) is aimed at finding songs with similar harmony in a data set. The method of Hanna et al. (2009) is based on a local alignment between all possible pairs of chord sequences. A dynamic programming algorithm calculates a score based on how many elementary operations (insertions, deletions or modifications of labels) have to be performed in order to transform one sequence into the other. De Haas et al. (2013c) introduce a geometric distance between chord sequences, which extends Lerdahl (2001)'s distance between chord pairs, to calculate the degree of similarity. For both methods, different representations of chord sequences are examined, varying in complexity from single roots only to complex chords. Both absolute chord representations or ones that are relative to a key are tried. The local alignment approach consistently outperforms the geometric distance based one, but the latter has a much more favourable runtime, which is another relevant point for the deployment of large-scale applications.

[1] `http://songle.jp/`

Similarity in harmony is also more specifically used to discover cover versions or live versions of a song in a database of music. İzmirli (2005b); Lee (2006b) use a Dynamic Time Warping algorithm to this end, while Bello (2007) examined approximate string-matching techniques to improve the robustness against estimation errors and key and tempo changes. Recognising the dependency of the previous approaches on pairwise comparisons between all songs in the data set, which makes scaling to large data sets impractical, Khadkevich and Omologo (2013a) propose an approach based on Locality Sensitive Hashing to overcome this problem.

A
Mapping of chord types to four triad types

```
Input: chordttype
Output: triadtype
begin
    triadtype ← undefined
    if chordtype = no-chord then
        triadtype ← no-chord
    else if M3 ∈ chordtype then
        if A5 ∈ chordtype ∧ P5 ∉ chordtype then
            triadtype ← aug
        else if P5 ∈ chordtype ∨ d5 ∉ chordtype then
            triadtype ← maj
        end
    else if m3 ∈ chordtype then
        if d5 ∈ chordtype ∧ P5 ∉ chordtype then
            triadtype ← dim
        else if P5 ∈ chordtype ∨ A5 ∉ chordtype then
            triadtype ← min
        end
    end
end
```

Algorithm 1: Triad type classification algorithm

B
List of songs in the SEMA data set

The SEMA (harmony) set is a collection of 142 song excerpts of 30 seconds in one of nine genres: Pop (37), Classical (26), World (24), New age/Ambient (13), Rock (11), Jazz/Blues (11), Dance/Techno (10), Rap/Hip-hop (6) and R&B/Reggae (4). It is a subset of the complete SEMA (metre) set that contains 161 song excerpts annotated with full metrical structure, described in Varewyck and Martens (2007).

ID	Artist	Title	Genre
1	Barry White	You're The First, The Last, My Everything	Pop
2	Billy Bragg	A New England	Pop
3	Chitra	Kehna Hi Kya	World
5	Beijing Opera	Monkey King	World
6	Perotinus	Viderunt Omnes	Classical
7	MC Solaar	Ganster Moderne	Rap/Hip-hop
9	Gasconge	Courante	World
10	Wim Mertens	Struggle For Pleasure	Classical
11	Jean-Baptiste Lully	Air Pour Madame La Dauphine (Idylle Sur La Paix)	Classical
13	Peru	El Contrapunto	World
14	Erykah Badu	Green Eyes	New age/Ambient
15	Duke Ellington	Black Beauty	Jazz/Blues
16	Björk	Human Behavior	Pop
17	De Kift	Molenaar	Pop
18	John Coltrane	Giant Steps	Jazz/Blues

ID	Artist	Title	Genre
20	Tielman Susato	Passe E Medio, Den Iersten Gaillard	Classical
21	Adriano Banchieri	Il Diletto Moderno Licenza E Di Nuovo Invita	Classical
22	Maurice Ravel	Bolero	Classical
23	Enya	Orinoco Flow	New age/Ambient
24	Portishead	Sour Times	New age/Ambient
25	Tracy Chapman	Fast Car	Pop
26	Faithless	Insomnia	Dance/Techno
27	Waldemar Bastos	Mbiri-Mbiri	World
28	NAS	Ouchie Wally Wally	R&B/Reggae
29	Urban Trad.	Quimper-Moscou	World
30	Jesus and Mary Chain	April Skies	Rock
32	China Dolls	Wo Ai Ni	Pop
33	Prince	Cream	Pop
34	Carrapicho	Tic Tic Tac	World
35	Art Garfunkel	Bright Eyes	Pop
36	Ella Fitzgerald	A Tisket A Tasket	Jazz/Blues
37	Angelo Badalamenti	Laurens Walking	Pop
38	Bob Marley	Corner Stone	R&B/Reggae
39	Gotan Project	Queremos Paz	Jazz/Blues
40	Olla Vogala	Samra	World
41	Mostafa Amar	Habib Hayati	Pop
42	Einsturzende Neubauten	Blume	New age/Ambient
43	Dolly Parton	Jolene	Pop
44	Nine Inch Nails	March Of The Pigs	Rock
45	Lucilla Galeazzi	Ah, Vita Bella	World
47	Astor Piazzolla	Oblivion	Classical
48	Johannes Brahms	Hungarian Dance No. 1 In G Minor	Classical
49	Gustav Mahler	Sehr Behaglich (Symphonie No. 4 In G Major)	Classical
51	Värttinä	Itkin	New age/Ambient
52	Maurice Ravel	Pavane De La Belle Au Bois Dormant (Ma Mère L'oye)	Classical
53	Aphex Twin	Film	Dance/Techno
54	Wolfgang Amadeus Mozart	Dies Irae (Requiem Kv 626)	Classical

ID	Artist	Title	Genre
55	Novastar	Never Back Down	Pop
57	Lou Reed	Perfect Day	Pop
59	Astor Piazzolla	Libertango	Classical
60	St. Germain	Land Of…	Jazz/Blues
61	Cirque du Soleil	Alegria	Pop
62	Buena Vista Social Club	Chan Chan	World
63	Sean Paul	Get Busy	Rap/Hip-hop
64	Bob Dylan	Hurricane	Pop
65	Claudio Monteverdi	Deus In Adjutorium (Vespro Della Beata Vergine)	Classical
66	Jonathan Richman	Egyptian Reggae	Pop
67	Christoph Willibald von Gluck	Che Faro Senza Euridice (Orfeo Ed Euridice)	Classical
68	Orbital	The Box	New age/Ambient
69	Richard Wagner	Vorspiel (Die Walküre, First Act)	Classical
71	Korea	Piri	World
72	16 Horsepower	Haw	Rock
73	Johannes Ciconia	Le Ray Au Soleil	Classical
74	Salomon Islands	Panflute Polyphony	World
76	DJ Tiësto	Traffic	Dance/Techno
77	Blur	Song 2	Rock
78	Franz Schubert	Allegro (Stringquartett No. 14 In D Minor "Der Tod Und Das Madchen")	Classical
79	Django Reinhardt	Minor Swing	Jazz/Blues
80	Zillertaler Schürzenjäger	Komm Nach Tirol	Pop
81	Daler Mehndi	Tunak Tunak Tun	Dance/Techno
82	Las Ketchup	The Ketchup Song	Pop
83	O-zone	Dragosta Din Tei	Dance/Techno
84	King Sunny Adé	Kiti Kiti	World
85	Rammstein	Du Hast	Rock
86	Madredeus	O Pastor	Classical
89	Kraftwerk	The Robots	Dance/Techno
90	Air	Cherry Blossom Girl	New age/Ambient
91	David Rosenboom	The Seduction Of Sapientia	Classical
92	Massive Attack	Teardrop	New age/Ambient
93	Gipsy Kings	A Tu Vera	World

ID	Artist	Title	Genre
94	Velvet Underground	Sunday Morning	Pop
95	Goran Bregovic	Kalashnikov	World
97	Tatu	Not Gonna Get Us	Dance/Techno
98	Fuyumi Sakamoto	Yumehotaru	World
99	Children of Bodom	Warheart	Rock
100	Lata Mangeshkar & Udit Narayan	Dil To Pagal Hai	World
101	Usher	Yeah!	Rap/Hip-hop
102	Ludwig van Beethoven	Allegro Con Brio (Stringquartett In F Minor Op. 95 "Serioso")	Classical
103	Duke Ellington	Caravan	Jazz/Blues
105	Enigma	Age Of Loneliness	New age/Ambient
106	Beastie Boys	Intergalactic	Rap/Hip-hop
107	Coldplay	Yellow	Rock
108	Nick Cave and The Bad Seeds	Into My Arms	Pop
109	John Coltrane	My Favorite Things	Jazz/Blues
111	Johann Sebastian Bach	Kommt, Ihr Töchter, Helft Mir Klagen (Mattäus-Passion, BWV 244)	Classical
112	Mano Negra	Le Bruit Du Frigo	World
113	Wolfgang Amadeus Mozart	Andante Grazioso, Variation III (Klaviersonata In A Major KV 331)	Classical
114	John Lee Hooker	Boom Boom	Jazz/Blues
115	Anonymous	Folia	Classical
116	Carl Orff	O Fortuna	Classical
117	Athanasius Kircher	La Carpinese	Classical
118	PJ Harvey	Hair	Rock
119	Joe Zawinul	Potato Blues	New age/Ambient
120	Temple of the Dog	All Night Thing	Pop
121	Costa Cordalis	Eleni	Classical
122	Johan Hoogewijs	Witse Twijfelt	New age/Ambient
123	Stefan Remmler	Vogel Der Nacht	World
124	Stooges	We Will Fall	World

ID	Artist	Title	Genre
125	King Kalakaua (adapt. John Kalapana)	Akahi Hoi	Jazz/Blues
126	Sezen Aksu	Gel Kavusalim Artik	World
128	Toto Cotugno	Adulele	R&B/Reggae
129	Baby VOX	Nae Sarang Ee Gi Reul	Pop
130	Sezen Aksu	Tutuklu Kaldim	New age/Ambient
131	Marco Borsato	Waarom Nou Jij	Pop
132	Sezen Aksu	Yanarim	World
133	Mini Moni	I Love Blues	Jazz/Blues
134	Weezer	Half Japanese Girl	Rock
135	Boyzone	Love Me For A Reason	Pop
136	Akino Arai	Kirei Na Kanjou	Pop
137	Manu Chao	Mama Call	World
138	Ayumi Hamasaki	Seasons	Pop
139	Baby V.O.X.	Betrayal	Pop
140	Derek R. Audette	Right In The Face Plate	Rock
141	Hanson	Penny And Me	Rock
142	Ayumi Hamasaki	Powder Snow	New age/Ambient
144	Mini Moni	Mini Moni Bus Guide	Pop
145	Rie Tanaka	Hitomi No Tonneru	Pop
146	Turbo	Tonight	Pop
147	The Pretenders	Don't Get Me Wrong	Pop
148	FinKL	Noon Dong Ja	R&B/Reggae
149	Beastie Boys	Intergalactic	Rap/Hip-hop
150	MC Solaar	La Vie Est Belle	Rap/Hip-hop
151	INXS	Elegantly Wasted	Pop
152	Megumi Hayashibara	Kimi Sae Ireba	Pop
153	Baby V.O.X.	Come To Me	Pop
154	Zone	Shiroi Hana	Pop
155	Calin vs Fantastic Plastic Machine	Samba De Minha Namoradinho	World
156	Madonna	True Blue	Pop
157	Fragile State	Every Day A Story (4 Hero Electric Fusion Remix)	Dance/Techno
158	Harry Anand	Kantaa Laga (DJ Doll Remix)	Pop
159	Junior Jack	Da Hype (Original Club Mix)	Dance/Techno
160	MC Sar & The Real McCoy	Another Night	Dance/Techno

Bibliography

Acoustics – standard tuning frequency (standard musical pitch), 1975. URL `http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=3601`.

Juan Pablo Bello. Audio-based cover song retrieval using approximate chord sequences: testing shifts, gaps, swaps and beats. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, pages 239–244, 2007.

Juan Pablo Bello and Jeremy Pickens. A robust mid-level representation for harmonic content in music signals. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 304–311, 2005.

Emmanuel Bigand, Richard Parncutt, and Fred Lerdahl. Perception of musical tension in short chord sequences: The influence of harmonic function, sensory dissonance, horizontal motion, and musical training. *Perception & Psychophysics*, 58(1):125–141, 1996.

Antonio Bonafonte, Xavier Ros, and Jose B. Mariño. An efficient algorithm to find the best state sequence in HSMM. pages 1547–1550, Berlin, Germany, 19–23 September 1993.

Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Audio chord recognition with Recurrent Neural Networks. In *Proceedings of the 14th Conference of the International Society for Music Information Retrieval (ISMIR)*, 2013.

Judith C. Brown. Calculation of a constant Q spectral transform. *Journal of the Acoustical Society of America*, 89(1):425–434, January 1991. doi: 10.1121/1.400476.

Judith C. Brown and Miller C. Puckette. An efficient algorithm for the calculation of a constant Q transform. *Journal of the Acoustical Society of America*, 92(5):2698–2701, November 1992. doi: 10.1121/1.404385.

John Ashley Burgoyne and Lawrence K. Saul. Learning harmonic relationships in digital audio with Dirichlet-based hidden Markov models. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 438–443, 2005.

John Ashley Burgoyne, Laurent Pugin, Corey Kereliuk, and Ichiro Fujinaga. A cross-validated study of modelling strategies for automatic chord recognition in audio. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, pages 251–254, 2007.

Giordano Cabral, Jean-Pierre Briot, and François Pachet. Impact of distance in pitch class profile computation. In *Proceedings of the Brazilian Symposium on Computer Music (SBCM)*, pages 319–324, Belo Horizonte, MG, Brazil, October 3–6 2005.

Benoit Catteau, Jean-Pierre Martens, and Marc Leman. A probabilistic framework for audio-based tonal key and chord recognition. In Reinhold Decker and Hans-Joachim Lenz, editors, *Advances in Data Analysis; Proceedings of the 30th Annual Conference of the Gesellschaft für Klassifikation e.V.*, volume X of *Studies in Classification, Data Analysis, and Knowledge Organization*, pages 637–644. Springer Berlin Heidelberg, Freie Universität Berlin, March 8—10 2006 2007. doi: 10.1007/978-3-540-70981-7_73.

Wei Chai. *Automated analysis of musical structure*. PhD thesis, Massachusetts Institute of Technology, September 2005.

Wei Chai and Barry Vercoe. Detection of key change in classical piano music. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 468–473, 2005.

Ruofeng Chen, Weibin Shen, Ajay Srinivasamurthy, and Parag Chordia. Chord recognition using duration-explicit hidden Markov models. In *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR)*, pages 445–450, 2012.

Heng-Tze Cheng, Yi-Hsuan Yang, Yu-Ching Lin, I-Bin Liao, and Homer H. Chen. Automatic chord recognition for music classification and retrieval. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, pages 1505–1508, Hannover, 2008. doi: 10.1109/ICME.2008.4607732.

Elaine Chew. The spiral array: an algorithm for determining key boundaries. In Christina Anagnostopoulou, Miguel Ferrand, and Alan Smaill, editors, *Proceedings of the 2nd International Conference on Music and Artificial Intelligence (ICMAI)*, volume 2445 of *Lecture Notes in Computer Science*, pages 18–31, Edinburgh, Scotland, UK, September 12—14 2002. Springer Berlin Heidelberg. doi: 10.1007/3-540-45722-4_4.

Elaine Chew and Yun-Ching Chen. Real-time pitch spelling using the spiral array. *Computer Music Journal*, 29(2):61–76, Summer 2005. doi: 10.1162/0148926054094378.

Taemin Cho. *Improved techniques for automatic chord recognition from music audio signals*. PhD thesis, New York University, 2013.

Taemin Cho and Juan P. Bello. On the relative importance of individual components of chord recognition systems. *IEEE Transactions on Audio, Speech and Language Processing*, 22(2):477–492, February 2014. doi: 10.1109/TASLP.2013.2295926.

Taemin Cho and Juan Pablo Bello. A feature smoothing method for chord recognition using recurrence plots. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, pages 651–656, 2011.

Taemin Cho, Ron J. Weiss, and Juan P. Bello. Exploring common variations in state of the art chord recognition systems. In *Proceedings of the Sound and Music Computing Conference*, pages 1–8, Barcelona, Spain, 2010. SMC.

Ching-Hua Chuan and Elaine Chew. Fuzzy analysis in pitch class determination for polyphonic audio key finding. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 296–303, 2005.

Richard Cohn. Introduction to Neo-Riemannian theory: a survey and a historical perspective. *Journal of Music Theory*, 42(2):167–180, Autumn 1998.

Norman D. Cook and Takefumi Hayashi. The psychoacoustics of harmony perception. *American Scientist*, 96(4):311–319, July–August 2008. doi: 10.1511/2008.73.3845.

Matt Cranitch, Derry Fitzgerald, and Matthew Hart. Key signature estimation. In *Proceedings of the Irish Signals and Systems Conference (ISSC)*, Derry, Ireland, 13–14 September 2007.

Lola L. Cuddy and Annabel J. Cohen. Recognition of transposed melodic sequences. *Quarterly Journal of Experimental Physiology*, 28(2):255–270, 1976. doi: 10.1080/14640747608400555.

W. Bas de Haas, José Pedro Magalhães, Frans Wiering, and Remco C. Veltkamp. Automatic functional harmonic analysis. *Computer Music Journal*, 37(4):37—-53, Winter 2013a. doi: 10.1162/COMJ_a_00209.

W. Bas de Haas, Anja Volk, and Frans Wiering. Structural segmentation of music based on repeated harmonies. In *Proceedings of the IEEE International Symposium on Multimedia (ISM)*, pages 255–258, Anaheim, CA, USA, 9–11 December 2013b. IEEE. doi: 10.1109/ISM.2013.48.

W. Bas de Haas, Frans Wiering, and Remco C. Veltkamp. A geometrical distance measure for determining the similarity of musical harmony. *International Journal of Multimedia Information Retrieval*, 2(3):189–202, September 2013c. doi: 10.1007/s13735-013-0036-6.

Alessio Degani, Marco Dalai, Riccardo Leonardi, and Pierangelo Migliorati. Comparison of tuning frequency estimation methods. *Multimedia Tools and Applications*, March 2014. doi: 10.1007/s11042-014-1897-2.

Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. doi: http://www.jstor.org/stable/2984875.

Karin Dressler and Sebastian Streich. Tuning frequency estimation using circular statistics. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 357–360, 2007.

Daniel P.W. Ellis and Graham Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages IV–1429 – IV–1432, April 15–20 2007. doi: 10.1109/ICASSP.2007.367348.

Shai Fine, Yoram Singer, and Naftali Tishby. The hierarchical hidden Markov model: analysis and applications. *Machine Learning*, 32:41–62, 1998. doi: 10.1023/A:1007469218079.

James L Flanagan and RM Golden. Phase vocoder. *Bell System Technical Journal*, 45(9):1493–1509, November 1966. doi: 10.1002/j.1538-7305.1966.tb01706.x.

Harvey Fletcher. Loudnes, pitch and the timbre of musical tones and their relation to the intensity, the frequency and the overtone structure. *Journal of the Acoustical Society of America*, 6(2):59–69, October 1934. doi: 10.1121/1.1915704.

Harvey Fletcher and W. A. Munson. Loudness, its definition, measurement and calculation. *Journal of the Acoustical Society of America*, 5:82–108, October 1933. doi: 10.1121/1.1915637.

Takuya Fujishima. Realtime chord recognition of musical sound: a system using Common Lisp Music. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 464–467, Ann Arbor, MI, USA, 1999. MPublishing, University of Michigan Library. doi: 2027/spo.bbp2372.1999.446.

Jon Gillick, Kevin Tang, and Robert M. Keller. Machine learning of jazz grammars. *Computer Music Journal*, 34(3):56—-66, Fall 2010. doi: 10.1162/COMJ_a_00006.

Nikolay Glazyrin. Mid-level features for audio chord estimation using stacked denoising autoencoders. In *Proceedings of the RusSIR Young Scientist Conference*, Kazan, Russia, September 16–20 2013.

Nikolay Glazyrin and Alexander Klepinin. Chord recognition using Prewitt filter and self-similarity. In *Proceedings of the Sound and Music Computing Conference*, pages 480 – 485, Copenhagen, Denmark, 2012.

Masataka Goto and Yoichi Muraoka. Real-time beat tracking for drumless audio signals: chord change detection for musical decisions. *Speech Communication*, 27(3–4):311–335, April 1999. doi: 10.1016/S0167-6393(98)00076-4.

Masataka Goto, Kazuyoshi Yoshii, Hiromasa Fujihara, Matthias Mauch, and Tomoyasu Nakano. Songle: a web service for active music listening improved by user contributions. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, pages 311–316, 2011.

Emilia Gómez. Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing, Special Cluster on Computation in Music*, 18(3):294–304, Summer 2006a. doi: 10.1287/ijoc.1040.0126.

Emilia Gómez. *Tonal description of music audio signals*. PhD thesis, Universitat Pompeu Fabra, 2006b.

Emilia Gómez and Perfecto Herrera. Estimating the tonality of polyphonic audio files: cognitive versus machine learning modelling strategies. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, pages 92–95, 2004.

Pierre Hanna, Matthias Robine, and Thomas Rocher. An alignment based system for chord sequence retrieval. In *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL)*, pages 101–104, New York, NY, USA, 2009. ACM. doi: 10.1145/1555400.1555417.

Christopher Harte, Mark Sandler, Samer Abdallah, and Emilia Gómez. Symbolic representation of musical chords: a proposed syntax for text annotations. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 66–71, 2005.

Christopher Harte, Mark Sandler, and Martin Gasser. Detecting harmonic change in musical audio. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia (AMCMM)*, pages 21–26, New York, NY, USA, 2006. ACM. doi: 10.1145/1178723.1178727.

Christopher A. Harte and Mark B. Sandler. Automatic chord identification using a quantised chromagram. In *Proceedings of the 118th Convention of the Audio Engineering Society*, Barcelona, Spain, May 28–31 2005.

Nicki Holighaus, Monika Dörfler, Gino Angelo Velasco, and Thomas Grill. A framework for invertible, real-time constant-Q transforms. *IEEE Transactions on Audio, Speech and Language Processing*, 21(4):775–785, April 2013. doi: 10.1109/TASL.2012.2234114.

Diane J. Hu and Lawrence K. Saul. A probabilistic topic model for unsupervised learning of musical key-profiles. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR)*, pages 441–446, 2009a.

Diane J. Hu and Lawrence K. Saul. A probabilistic topic model for music analysis. In *Proceedings of the NIPS Applications for Topic Models Workshop*, 2009b.

Eric J. Humphrey and Juan P. Bello. Rethinking automatic chord recognition with Convolutional Neural Networks. In *Proceedings of the IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 357–362, Boca Raton, FL, 12–15 December 2012. doi: 10.1109/ICMLA.2012.220.

Eric J. Humphrey, Taemin Cho, and Juan P. Bello. Learning a robust Tonnetz-space transform for automatic chord recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 453–456. IEEE, 2012. doi: 10.1109/ICASSP.2012.6287914.

Katsutoshi Itoyama, Tetsuya Ogata, and Hiroshi G. Okuno. Automatic chord recognition based on probabilistic integration of acoustic features, bass sounds, and chord transition. In He Jiang, Wei Ding, Moonis Ali, and Xindong Wu, editors, *Advanced Research in Applied Artificial Intelligence; Proceedings of the 25th International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems (IEA/AIE)*, volume 7345 of *Lecture Notes in Computer Science*, pages 58–67. Springer Berlin Heidelberg, Dalian, China, June 9-12 2012. doi: 10.1007/978-3-642-31087-4_7.

Nanzhu Jiang, Peter Grosche, Verena Konz, and Meinard Müller. Analyzing chroma feature types for automated chord recognition. In *Proceedings of the AES 42nd International Conference: Semantic Audio*, pages 7–2, Ilmenau, Germany, 22–24 July 2011. Audio Engineering Society.

Kunio Kashino and Norihiro Hagita. A music scene analysis system with the MRF-based information integration scheme. In *Proceedings of the 13th International Conference on Pattern Recognition (ICPR)*, volume 2, pages 725–729. IEEE, IEEE, 1996. doi: 10.1109/ICPR.1996.546918.

Kunio Kashino, Kazuhiro Nakadai, Tomoyoshi Kinoshita, and Hidehiko Tanaka. Application of Bayesian probability network to music scene analysis. In David F. Rosenthal and Hiroshi G. Okuno, editors, *Computational Auditory Scene Analysis; Proceedings of the IJCAI-95 Workshop*, pages 115–137. Lawrence Erlbaum Associates, Mahwah, NJ, USA, May 1 1998. doi: http://www.crcpress.com/product/isbn/9780805822830.

Haruhiro Katayose, M. Imai, and Seiji Inokuchi. Sentiment extraction in music. In *Proceedings of the 9th International Conference on Pattern Recognition (ICPR)*, volume 2, pages 1083–1087. IEEE, 1988. doi: 10.1109/ICPR.1988.28447.

Robert Keller, Alexandra Schofield, August Toman-Yih, Zachary Merritt, and John Elliott. Automating the explanation of jazz chord progressions using idiomatic analysis. *Computer Music Journal*, 37(4):54—69, Winter 2013. doi: 10.1162/COMJ_a_00201.

Maksim Khadkevich and Maurizio Omologo. Phase-change based tuning for automatic chord recognition. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 506–509, Como, Italy, September 1–4 2009a.

Maksim Khadkevich and Maurizio Omologo. Use of hidden Markov models and factored language models for automatic chord recognition. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR)*, pages 561–566, 2009b.

Maksim Khadkevich and Maurizio Omologo. Time-frequency reassigned features for automatic chord recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 181 – 184, Prague, May 22–27 2011. doi: 10.1109/ICASSP.2011.5946370.

Maksim Khadkevich and Maurizio Omologo. Large-scale cover song identification using chord profiles. In *Proceedings of the 14th Conference of the International Society for Music Information Retrieval (ISMIR)*, pages 233–238, 2013a.

Maksim Khadkevich and Maurizio Omologo. Reassigned spectrum-based feature extraction for GMM-based automatic chord recognition. 2013 (15):1–12, June 2013b. doi: 10.1186/1687-4722-2013-15.

Reinhard Kneser and Hermann Ney. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 1, pages 181–184, Detroit, MI, USA, 9–12 May 1995. doi: 10.1109/ICASSP.1995.479394.

Carol Krumhansl. *Cognitive foundations of musical pitch*. Oxford University Press, 1990. doi: 10.1093/acprof:oso/9780195148367.001.0001.

Carol L. Krumhansl and Edward J. Kessler. Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychological Review*, 89(4):334–368, July 1982. doi: 10.1037/0033-295X.89.4.334.

Kyogu Lee. Automatic chord recognition using enhanced pitch class profile. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 306–313, Ann Arbor, MI, USA, 2006a. MPublishing, University of Michigan Library. doi: 2027/spo.bbp2372.2006.064.

Kyogu Lee. Identifying cover songs from audio using harmonic representation. In *Proceedings of the Music Information Retrieval EXchange (MIREX)*, 2006b.

Kyogu Lee and Malcolm Slaney. Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio. *IEEE Transactions on Audio, Speech and Language Processing*, 16(2): 291–301, February 2008. doi: 10.1109/TASL.2007.914399.

Jüri Lember and Alexey A. Koloydenko. Bridging Viterbi and posterior decoding: a generalized risk approach to hidden path inference based on hidden Markov models. *Journal of Machine Learning Research*, 15(1): 1–58, 2014.

Alexander Lerch. On the requirement of automatic tuning frequency estimation. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 212—-215, 2006.

Fred Lerdahl. Tonal pitch space. *Music Perception*, 5(3):315–349, Spring 1988. doi: 10.2307/40285402.

Fred Lerdahl. *Tonal pitch space*. Oxford University Press, New York, 2001. doi: 10.1093/acprof:oso/9780195178296.001.0001.

Ernest Li and Juan P. Bello. Key-independent classification of harmonic change in musical audio. In *Proceedings of the 123rd Convention of the Audio Engineering Society (AES)*, New York, NY, USA, 5–8 October 2007.

Namunu Chinthaka Maddage. Automatic structure detection for popular music. *IEEE Multimedia*, 13(1):65–77, 2006. doi: 10.1109/MMUL.2006.3.

Keith D. Martin. A blackboard system for automatic transcription of simple polyphonic music. Technical Report 385, Massachusetts Institute of Technology, Media Laboratory Perceptual Computing Section, July 1996.

Matthias Mauch. *Automatic chord transcription from audio using computational models of musical context*. PhD thesis, School of Electronic Engineering and Computer Science Queen Mary, University of London, March 23 2010.

Matthias Mauch and Simon Dixon. A discrete mixture model for chord labelling. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*, pages 45–50, 2008.

Matthias Mauch and Simon Dixon. Using musical structure to enhance automatic chord transcription. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR)*, pages 231–236, 2009.

Matthias Mauch and Simon Dixon. Approximate note transcription for the improved identification of difficult chords. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, pages 135–140, 2010a.

Matthias Mauch and Simon Dixon. Simultaneous estimation of chords and musical context from audio. *IEEE Transactions on Audio, Speech and Language Processing*, 18(6):1280–1289, August 2010b. doi: 10.1109/TASL.2009.2032947.

Matthias Mauch, Chris Cannam, Matthew Davies, Simon Dixon, Chris Harte, Sefki Kolozali, Dan Tidhar, and Mark Sandler. OMRAS2 metadata project 2009. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR)*, 2009.

Lesley Mearns, Emmanouil Benetos, and Simon Dixon. Automatically detecting key modulations in J.S. Bach chorale recordings. In *Proceedings of the Sound and Music Computing Conference*, 2011.

David Meredith. Comparing pitch spelling algorithms on a large corpus of tonal music. In Uffe Kock Wiil, editor, *Second International Symposium on Computer Music Modeling and Retrieval, Revised Papers*, volume 3310 of *Lecture Notes in Computer Science*, pages 173–192, Esbjerg, Denmark, May 26–29 2004. Springer Berlin Heidelberg. doi: 10.1007/978-3-540-31807-1_14.

Joshua Morman and Lawrence Rabiner. A system for the automatic segmentation and classification of chord sequences. In *Proceedings of the 1st ACM Workshop on Audio and Music Computing Multimedia (AMCMM)*, pages 1–10. ACM, 27 October 2006. doi: 10.1145/1178723.1178725.

Meinard Müller and Sebastian Ewert. Towards timbre-invariant audio features for harmony-based music. *IEEE Transactions on Audio, Speech and Language Processing*, 18(3):649–662, March 2010. doi: 10.1109/TASL.2010.2041394.

S. Hamid Nawab, Salma Abu Ayyash, and Robert Wotiz. Identification of musical chords using constant-Q spectra. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 5, pages 3373–3376, Salt Lake City, UT, USA, 7–11 May 2001. IEEE. doi: 10.1109/ICASSP.2001.940382.

Yizhao Ni, Matt McVicar, Raúl Santos-Rodríguez, and Tijl De Bie. An end-to-end machine learning system for harmonic analysis of music. *IEEE Transactions on Audio, Speech and Language Processing*, 20(6):1771–1783, August 2012. doi: 10.1109/TASL.2012.2188516.

Eric Nichols, Dan Morris, and Sumit Basu. Data-driven exploration of musical chord sequences. In *Proceedings of the 14th International Conference on Intelligent User Interfaces (IUI)*, pages 227–236, New York, NY, USA, 2009. ACM. doi: 10.1145/1502650.1502683.

Katy Noland and Mark Sandler. Key estimation using a hidden Markov model. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pages 121–126, 2006.

Katy Noland and Mark Sandler. Influences of signal processing, tone profiles and chord progressions on a model for estimating the musical key from audio. *Computer Music Journal*, 33(1):MIT Press Journals, Spring 2009. doi: 10.1162/comj.2009.33.1.42.

Nobutaka Ono, Kenichi Miyamoto, Jonathan Le Roux, Hirokazu Kameoka, and Shigeki Sagayama. Separation of a monaural audio signal into harmonic/percussive components by complementary diffusion on spectrogram. Lausanne, Switzerland, August 25–29 2008. EURASIP.

Laurent Oudre, Yves Grenier, and Cédric Févotte. Chord recognition by fitting rescaled chroma vectors to chord templates. *IEEE Transactions on Audio, Speech and Language Processing*, 19(7):2222 – 2233, September 2011. doi: 10.1109/TASL.2011.2139205.

François Pachet. *Readings in Music and Artificial Intelligence*, chapter Computer analysis of jazz chord sequences: Is Solar a blues?, pages 1–21. Contemporary Music Studies. Harwood Academic Publishers, August 8 2000. doi: http://www.routledge.com/books/details/9789057550942/.

Hélène Papadopoulos and Geoffroy Peeters. Large-scale study of chord estimation algorithms based on chroma representation and HMM. In

Proceedings of the International Workshop on Content-Based Multimedia Indexing (CBMI), pages 53–60, Bordeaux, France, June 25–27 2007. doi: 10.1109/CBMI.2007.385392.

Hélène Papadopoulos and Geoffroy Peeters. Joint estimation of chords and downbeats from an audio signal. *IEEE Transactions on Audio, Speech and Language Processing*, 19(1):138–152, January 2011. doi: 10.1109/TASL.2010.2045236.

Hélène Papadopoulos and Geoffroy Peeters. Local key estimation from an audio signal relying on harmonic and metrical structures. *IEEE Transactions on Audio, Speech and Language Processing*, 20(4):1297–1312, May 2012. doi: 10.1109/TASL.2011.2175385.

Bryan Pardo and William P. Birmingham. Algorithms for chordal analysis. *Computer Music Journal*, 26(2):27–49, Summer 2002. doi: 10.1162/014892602760137167.

Johan Pauwels and Jean-Pierre Martens. Integrating musicological knowledge into a probabilistic system for chord and key extraction. In *Proceedings of the 128th Convention of the AES*, London, UK, May 22–25 2010.

Johan Pauwels and Geoffroy Peeters. Evaluating automatically estimated chord sequences. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013. doi: 10.1109/ICASSP.2013.6637748.

Steffen Pauws. Musical key extraction from audio. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, pages 96–99, 2004.

Geoffroy Peeters. Musical key estimation of audio signal based on hidden Markov modelling of chroma vectors. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, pages 127–131, Montreal, Quebec, Canada, September 18–20 2006a.

Geoffroy Peeters. Chroma-based estimation of musical key from audio-signal analysis. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pages 115–120, 2006b.

Hendrik Purwins, Benjamin Blankertz, and Klaus Obermayer. A new method for tracking modulations in tonal music in audio data format. In *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Network (IJCNN)*, volume 6, pages 270–275, Como, Italy, 24–27 July 2000. IEEE. doi: 10.1109/IJCNN.2000.859408.

Carlos Pérez-Sancho, David Rizo, and José M. Iñesta. Genre classification using chords and stochastic language models. *Connection science*, 21(2–3): 145–159, June–September 2009. doi: 10.1080/09540090902733780.

Christopher Raphael and Josh Stoddard. Harmonic analysis with probabilistic graphical models. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)*, 2003.

Matthias Robine, Thomas Rocher, and Pierre Hanna. Improvements of symbolic key finding methods. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 1–4, Ann Arbor, MI, USA, 2008. MPublishing, University of Michigan Library. doi: 2027/spo.bbp2372.2008.131.

Thomas Rocher, Matthias Robine, Pierre Hanna, and Robert Strandh. Dynamic chord analysis for symbolic music. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 41–48, Montreal, Canada; publisher: Ann Arbor, MI, USA, August 16–21 2009. MPublishing, University of Michigan Library. doi: 2027/spo.bbp2372.2009.009.

Thomas Rocher, Matthias Robine, Pierre Hanna, and Laurent Oudre. Concurrent estimation of chords and keys from audio. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, pages 141–146, 2010.

Martin Rohrmeier and Thore Graepel. Comparing feature-based models of harmony. In *Proceedings of the 9th International Symposium on Computer Music Modelling and Retrieval (CMMR)*, pages 357–370, 2012.

Martin J. Russell and Anneliese E. Cook. Experimental evaluation of duration modelling techniques for automatic speech recognition. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 12, pages 2376–2379. IEEE, 1987. doi: 10.1109/ICASSP.1987.1169918.

Matti P. Ryynänen and Anssi P. Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32(3):72–86, Fall 2008. doi: 10.1162/comj.2008.32.3.72.

Christian Sailer and Katja Rosenbauer. A bottom-up approach to chord detection. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 612–615, Ann Arbor, MI, USA, 2006. MPublishing, University of Michigan Library. doi: 2027/spo.bbp2372.2006.125.

Ricardo Scholz, Emmanuel Vincent, and Frédéric Bimbot. Robust modeling of musical chord sequences using probabilistic n-grams. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 53–56, 2009. doi: 10.1109/ICASSP.2009.4959518.

Björn Schuller, Benedikt Hörnler, Dejan Arsic, and Gerhard Rigoll. Audio chord labeling by musiological modeling and beat-synchronization. In *Proceedings of the IEEE International Conference on Multimedia and Expo*

(ICME), pages 526–529, New York, NY, USA, June 28–July 3 2009a. IEEE. doi: 10.1109/ICME.2009.5202549.

Björn Schuller, Alexander Lehmann, Felix Weninger, Florian Eyben, and Gerhard Rigoll. Blind enhancement of the rhythmic and harmonic sections by NMF: does it help? In *Proceedings of the International Conference on Acoustics (NAG/DAGA)*, pages 361–364, Rotterdam, Netherlands, 23–26 March 2009b.

Christian Schörkhuber, Anssi Klapuri, Nicki Holighaus, and Monika Dörfler. A Matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution. In *Proceedings of the 53rd International Audio Engineering Society Conference: Semantic Audio*, London, UK, January 27–29 2014. Audio Engineering Society.

Alexander Sheh and Daniel P.W. Ellis. Chord segmentation and recognition using EM-trained hidden Markov models. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)*, pages 183–189, 2003.

Arun Shenoy and Ye Wang. Key, chord, and rhythm tracking of popular music recordings. *Computer Music Journal*, 29(3):75–86, 2005. doi: 10.1162/0148926054798205.

Arun Shenoy, Roshni Mohapatra, and Ye Wang. Key determination of acoustic musical signals. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME)*, volume 3, pages 1771–1774. IEEE, 27-30 June 2004. doi: 10.1109/ICME.2004.1394598.

Roger N. Shepard. Circularity in judgement of relative pitch. *Journal of the Acoustical Society of America*, 36(12):2346–2353, December 1964. doi: 10.1121/1.1919362.

Ian Simon, Dan Morris, and Sumit Basu. MySong: automatic accompaniment generation for vocal melodies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 725–734, Florence, Italy, 5–10 April 2008. ACM. doi: 10.1145/1357054.1357169.

Mark J. Steedman. A generative grammar for jazz chord sequences. *Music Perception*, 2(1):52–77, Fall 1984. doi: 10.2307/40285282.

Michael Stein, Benjamin M. Schubert, Matthias Gruhne, Gabriel Gatzsche, and Markus Mehnert. Evaluation and comparison of audio chroma feature extraction methods. In *Proceedings of the 126th Convention of the Audio Engineering Society (AES)*, Munich, Germany, May 7–10 2009.

Borching Su and Shyh-Kang Jeng. Multi-timbre chord classification using wavelet transform and self-organized map neural networks. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 5, pages 3377–3380, Salt Lake City, UT, USA, 2001. IEEE. doi: 10.1109/ICASSP.2001.940383.

Kouhei Sumi, Katsutoshi Itoyama, Kazuyoshi Yoshii, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. Automatic chord recognition based on probabilistic integration of chord transition and bass pitch estimation. In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*, pages 39–44, 2008.

Jiayin Sun, HaiFeng Li, and Li Lei. Key detection through pitch class distribution model and ANN. In *Proceedings of the 16th International Conference on Digital Signal Processing (ICDSP)*, pages 1–6. IEEE, 2009. doi: 10.1109/ICDSP.2009.5201119.

Annie H. Takeuchi. Maximum key-profile correlation (MKC) as a measure of tonal structure in music. *Perception & Psychophysics*, 56(3):335—-346, 1 May 1994. doi: 10.3758/BF03209767.

David Temperley. What's key for key? The Krumhansl-Schmuckler key-finding algorithm reconsidered. *Music Perception*, 17(1):65–100, 1999. doi: 10.2307/40285812.

David Temperley and Daniel Sleator. Modeling meter and harmony: a preference rule approach. *Computer Music Journal*, 23(1):10–27, Spring 1999. doi: 10.1162/014892699559616.

George Tzanetakis and Perry Cook. MARSYAS: a framework for audio analysis. *Organised sound*, 4(3):Cambridge University Press, 2000. doi: 10.1017/S1355771800003071.

Yushi Ueda, Yuki Uchiyama, Takuya Nishimoto, Nobutaka Ono, and Shigeki Sagayama. HMM-based approach for automatic chord detection using refined acoustic features. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5518–5521, Dallas, TX, USA, 14-19 March 2010. doi: 10.1109/ICASSP.2010.5495218.

Matthias Varewyck and Jean-Pierre Martens. Assessment of state-of-the-art meter analysis systems with an extended meter description model. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, pages 311–314, 2007.

Matthias Varewyck, Johan Pauwels, and Jean-Pierre Martens. A novel chroma representation of polyphonic music based on multiple pitch

tracking techniques. In *Proceedings of the 16th ACM International Conference on Multimedia (ACM MM)*, pages 667–670, New York, NY, USA, 2008. ACM. doi: 10.1145/1459359.1459455.

Andrew J. Viterbi. Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE Transactions on Information Theory*, 13(2):260–269, April 1967. doi: 10.1109/TIT.1967.1054010.

Piet G. Vos and Erwin W. Van Geenen. A parallel processing key-finding model. *Music Perception*, 14(2):185–223, Winter 1996. doi: 10.2307/40285717.

Gregory Wakefield. Mathematical representation of joint time-chroma distributions. In Franklin T. Luk, editor, *Proceedings of the 9th SPIE Conference on Advanced Signal Processing Algorithms, Architectures, and Implementations*, volume 3807, pages 637–645, November 2 1999. doi: 10.1117/12.367679.

Yun-Sheng Wang. Toward segmentation of popular music. In *Proceedings of the 3rd ACM International Conference on Multimedia Retrieval (ICMR)*, pages 345–348. ACM, 16 April 2013. doi: 10.1145/2461466.2461534.

Yun-Sheng Wang and Harry Wechsler. Musical keys and chords recognition using unsupervised learning with infinite Gaussian mixture. In *Proceedings of the 2nd ACM International Conference on Multimedia Retrieval (ICMR)*, Hong Kong, China, 5–8 June 2012. ACM; New York, NY, USA. doi: 10.1145/2324796.2324834.

Yun-Sheng Wang and Harry Wechsler. Unsupervised audio key and chord recognition. In *Proceedings of the International Conference on Digital Audio Effects (DAFx)*, Maynooth, Ireland, 2–5 September 2013.

Jan Weil, Thomas Sikora, Jean-Luc Durrieu, and Gaël Richard. Automatic generation of lead sheets from polyphonic music signals. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR)*, pages 603–608, 2009.

Adrian Weller, Daniel Ellis, and Tony Jebara. Structured prediction models for chord transcription of music audio. In *Proceedings of the IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 590 – 595, Miami Beach, FL, December 13–15 2009. doi: 10.1109/ICMLA.2009.132.

Frank Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, December 1945. doi: 10.2307/3001968.

Takuya Yoshioka, Tetsuro Kitahara, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. Automatic chord transcription with concurrent

recognition of chord symbols and boundaries. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR)*, pages 100–105, 2004.

Robert W. Young. Terminology for logarithmic frequency units. *Journal of the Acoustical Society of America*, 11(1):134–139, July 1939. doi: 10.1121/1.1916017.

Shun-Zeng Yu. Hidden semi-Markov models. *Artificial intelligence*, 174(2): 215–243, February 2010. doi: 10.1016/j.artint.2009.11.011.

Veronika Zenz and Andreas Rauber. Automatic chord detection incorporating beat and key detection. In *Proceedings of the International Conference on Signal Processing and Communications (ICSPC)*, pages 1175–1178, Dubai, UAE, 24-27 November 2007. IEEE. doi: 10.1109/ICSPC.2007.4728534.

Yongwei Zhu and Mohan S. Kankanhalli. Precise pitch profile feature extraction from musical audio for key detection. *IEEE Transactions on Multimedia*, 8(3):575 – 584, June 2006. doi: 10.1109/TMM.2006.870727.

Yongwei Zhu, Mohan S. Kankanhalli, and Sheng Gao. Music key detection for musical audio. In *Proceedings of the 11th International Multimedia Modelling Conference (MMM)*, pages 30–37. IEEE, 12–14 January 2005. doi: 10.1109/MMMC.2005.56.

Giorgio Zoia, Ruohua Zhou, and Daniel Mlynek. A multi-timbre chord/harmony analyzer based on signal processing and neural networks. In *Proceedings of the 6th IEEE Workshop on Multimedia Signal Processing (MMSP)*, pages 219–222. IEEE, 2004. doi: 10.1109/MMSP.2004.1436532.

Özgür İzmirli. Template based key finding from audio. In Andres Lewin-Richter and Xavier Serra, editors, *Proceedings of the International Computer Music Conference (ICMC)*, pages 211–214, San Francisco, 2005a. International Computer Music Association. doi: http://hdl.handle.net/2027/spo.bbp2372.2005.181.

Özgür İzmirli. Tonal similarity from audio using a template based attractor model. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 540–545, 2005b.

Özgür İzmirli. Audio key finding using low-dimensional spaces. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pages 127–132, 2006.

Özgür İzmirli. Localized key finding from audio using non-negative matrix factorization for segmentation. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, pages 195–200, 2007.

Özgür İzmirli and Roger B. Dannenberg. Understanding features and distance functions for music sequence alignment. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, pages 411–416, 2010.