Analyzing Chroma Feature Types for Automated Chord Recognition

Nanzhu Jiang[1], Peter Grosche[1], Verena Konz[1], and Meinard Müller[1]

[1]*Saarland University and MPI Informatik, Campus E1.4, 66123 Saarbrücken, Germany*

Correspondence should be addressed to Nanzhu Jiang (njiang@mpi-inf.mpg.de)

ABSTRACT

The computer-based harmonic analysis of music recordings with the goal to automatically extract chord labels directly from the given audio data constitutes a major task in music information retrieval. In most automated chord recognition procedures, the given music recording is first converted into a sequence of chroma-based audio features and then pattern matching techniques are applied to map the chroma features to chord labels. In this paper, we analyze the role of the feature extraction step within the recognition pipeline of various chord recognition procedures based on template matching strategies and hidden Markov models. In particular, we report on numerous experiments which show how the various procedures depend on the type of the underlying chroma feature as well as on parameters that control temporal and spectral aspects.

1. INTRODUCTION

A chord is defined as the simultaneous sounding of two or more different notes [16]. The progression of chords over time closely relates to the harmonic content of a piece of music, which plays a central attribute of Western tonal music. Such harmonic chord progressions are not only of musical importance, but also constitute a powerful mid-level representation for the underlying musical signal and can be applied for various tasks such as music segmentation, cover song identification, or audio matching. There are many variants for defining the task of chord recognition depending on the music representation, the temporal resolution, the level of abstraction, or the chords to be considered in the analysis. In this paper, we consider the case of audio representations where a piece of music is given in the form of a recorded performance. Here, the chord recognition task consists in first splitting up the recording into segments and then assigning a chord label to each segment. The segmentation specifies the start time and end time of a chord, and the chord label specifies which chord is played during this time period.

Chord recognition is one of the central tasks in the field of music information retrieval (MIR), which is also reflected by numerous contributions see, e. g., [2, 3, 5, 8, 16, 21, 22, 23, 24]. Most of the described chord recognition procedures proceed in a similar fashion. In the first step, the given music recording is converted into a sequence of chroma-based audio features. These features are often further processed, for example, by applying suitable smoothing filters to even out temporal outliers or by applying logarithmic compression or whitening procedures to enhance small yet perceptually relevant spectral components. In the next step, pattern matching techniques are applied to map the chroma features to chord labels that correspond to the various musical chords to be considered. In the last step, further post-filtering techniques are applied to smooth out local misclassifications. Often, hidden Markov models (HMMs) are used which jointly perform the pattern matching and temporal filtering steps within one optimization procedure.

Even though numerous procedures for automated chord labeling have been described in the literature, the delicate interplay of the various feature extraction, filtering, and pattern matching components is still not sufficiently investigated and understood. The situation is complicated by the fact that the components' behavior may crucially depend on a variety of parameters that allow for adjusting temporal, spectral, or dynamical aspects. In [22], the influence of various aspects and parameters of a typical HMM-based chord recognizer is investigated. In [3], a detailed investigation is described to better understand the interrelation of different chord recognition compo-

nents with a focus on the impact of filtering and pattern matching strategies. However, the impact of different feature extraction strategies was not investigated being left for future work.

In this paper, we continue this strand of research by analyzing the impact of various types of chroma features in the context of the chord recognition task. In particular, we report on numerous experiments which show how different recognition procedures substantially depend on the underlying chroma representation and on parameters that control temporal and spectral aspects. The remainder of this paper is organized as follows. We first give an overview of different chord recognition procedures (Section 2) and describe various types of chroma features (Section 3). As the main contribution of this paper, we then report on our extensive experiments (Section 4) and discuss the results. The paper concludes with some perspectives on future work (Section 5).

2. CHORD RECOGNITION

As mentioned in the introduction, a typical chord recognition system consists of two main steps, see Fig. 1 for an illustration. In the first step, the given audio recording (Fig. 1(b)) is transformed into a sequence $X = (x_1, x_2, \ldots, x_N)$ of feature vectors $x_n \in \mathscr{F}$, $n \in [1:N] := \{1, \ldots, N\}$. Here, $\mathscr{F}$ denotes a suitable feature space. Most recognition systems are based on so-called chroma features or pitch class profiles (see Fig. 1(c)), which we discuss in detail in Section 3. In the second step, using suitable pattern matching techniques, each feature vector x_n is mapped to a chord label $\lambda_{x_n} \in \Lambda$, see Fig. 1(d). Here, Λ denotes a suitably defined set of all possible chord labels. In the following, we consider the case that Λ consists of the twelve major and minor triads, i. e.,

$$\Lambda = \{\mathtt{C}, \mathtt{C}^\sharp, \ldots, \mathtt{B}, \mathtt{Cm}, \mathtt{C}^\sharp\mathtt{m}, \ldots, \mathtt{Bm}\}. \quad (1)$$

The restriction to these 24 chord classes, even though problematic from a musical point of view, is often made in the chord recognition literature.

There are many ways of performing the pattern matching step based on template-based matching strategies [5], hidden Markov models (HMMs) [23, 24, 3], or more complex Bayesian networks [16]. For an overview of such methods and the influence of various model parameters, we refer to [3, 16, 22]. Since in this paper we focus on the feature extraction step, we now give a brief summary of the pattern matching techniques used in the subsequent experiments and refer to the literature for details.

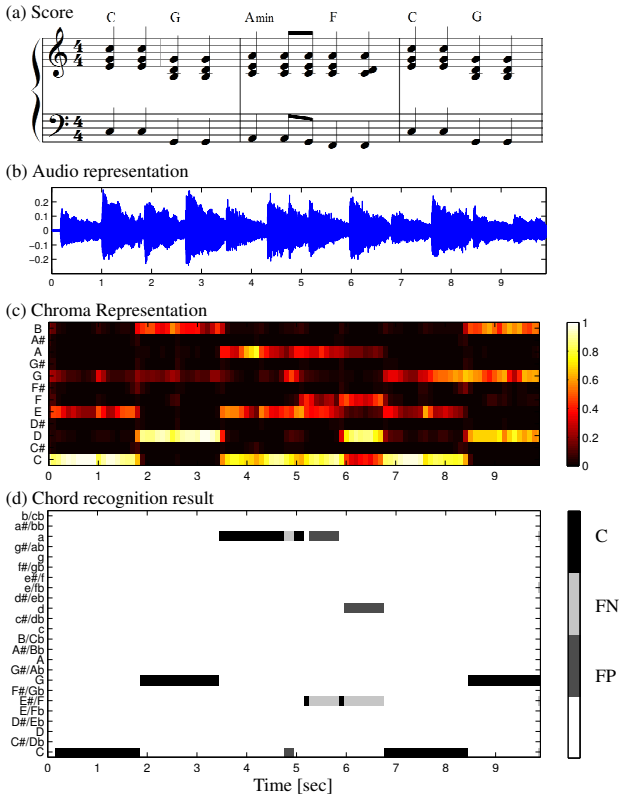

Fig. 1: Chord recognition task illustrated by the first measures of the Beatles song "Let It Be". **(a):** Score of the first three measures. **(b):** Audio representation of these measures. **(c):** Chroma representation. **(d):** Chord recognition result indicating correct (C), false negative (FN), and false positive (FP) labels.

2.1. Template-based Approach

As the first pattern matching technique, we use a simple template-based labeling strategy. Here, the idea is to precompute a set $\mathscr{T} \subset \mathscr{F}$ of templates that correspond to the set of chord labels. The elements of $\mathscr{T}$ are denoted by $\mathbf{t}_\lambda \in \mathscr{T}$, $\lambda \in \Lambda$. Intuitively, each template is given in the form of a kind of prototype chroma vector that corresponds to a specific musical chord. Furthermore, we fix a distance measure $d : \mathscr{F} \times \mathscr{F} \to \mathbb{R}$ that allows for comparing different chroma features. In the following, we use the cosine measure defined by

$$d(x,y) = 1 - \frac{\langle x|y\rangle}{\|x\| \cdot \|y\|}, \quad (2)$$

for $x, y \in \mathscr{F} \setminus \{0\}$. In the case $x = 0$ or $y = 0$, we set $d(x,y) = 1$. Here, $\|\cdot\|$ denotes the Euclidean norm (also referred to as ℓ^2-norm).

Then, the template-based chord recognition procedure consists in assigning the chord label that minimizes the distance between the corresponding template and the given feature vector $x = x_n$:

$$\lambda_x := \operatorname*{argmin}_{\lambda \in \Lambda} d(\mathbf{t}_\lambda, x). \quad (3)$$

Note that this procedure works in a purely framewise fashion without considering any temporal context.

There are several strategies for determining suitable chord templates based on musical knowledge or learning procedures using labeled training data. In the following, we consider binary templates and averaged templates. The set $\mathscr{T}^b$ consists of 24 binary templates, each of which being a 12-dimensional binary vector with three non-zero entries equal to one. These non-zero entries correspond to the three chromas the corresponding chord is composed of. For example, the binary template corresponding to the major chord $\mathtt{C} = \{\mathrm{C}, \mathrm{E}, \mathrm{G}\}$ is given by

$$\mathbf{t}^b_{\mathtt{C}} = (1,0,0,0,1,0,0,1,0,0,0,0)^T. \quad (4)$$

Furthermore, the set $\mathscr{T}^a$ consists of averaged templates, which are learned from training material by averaging suitable chroma vectors obtained from labeled audio data. For example, the averaged template $\mathbf{t}^a_{\mathtt{C}}$ is obtained by averaging all chroma vectors from the training set labeled as C.

The two template-based chord recognition approaches are denoted by $\mathtt{T}^{\mathtt{b}}$ and $\mathtt{T}^{\mathtt{a}}$, respectively.

2.2. Gaussian-based Approach

Next, we introduce a chord recognition procedure based on Gaussian distributions. Here, the chord templates are replaced by chord models each specified by a multivariate Gaussian distribution given in terms of a mean vector μ and a covariance matrix Σ. As for the averaged templates, μ and Σ are learned from labeled audio data. Then, the distance of a given chroma vector to a chord model is expressed by a Gaussian probability value and the assigned label is determined by the probability-maximizing chord model (instead of the cost-minimizing chord template), see [3]. The Gaussian-based chord recognition approach is denoted by `GP`.

2.3. HMM-based Approach

Finally, we summarize an HMM-based chord recognition procedure, which was originally suggested by Sheh and Ellis [23] and is now the most widely used chord labeling approach. The strength of this approach is that HMMs also account for the temporal context in the classification stage, which can be considered as a kind of context-aware filtering of the matching results. To this end, in addition to the Gaussian models, one needs transition probabilities that express the likelihood of passing over from one chord label to any of the other chord labels. These probabilities are given by a transition matrix $\Omega \in [0,1]^{24\times 24}$, which can be specified manually based on musical knowledge or automatically by using a training procedure reverting to suitable training material. For the labeling procedure, one then needs a Viterbi decoding algorithm to determine a chord label sequence that jointly maximizes the output probabilities defined by the Gaussian distributions and the transition probabilities, see [23]. The determination of the transition matrix also plays a crucial role in the chord recognition context and has been studied in various contributions [2, 22, 3]. In our experiments, we determine Ω using training data with annotated chord labels, see Section 4.1. The HMM-based chord recognition approach is denoted by `HMM`.

The HMM-based approach used in our experiments is conceptually state-of-the-art. However, as the focus of our evaluation lies on the feature side, we revert to a very basic variant. More advanced implementations are introduced in [3, 24, 16].

3. FEATURE EXTRACTION

Chroma-based audio features, sometimes also referred to as pitch class profiles, are a well-established tool in processing and analyzing music data [1, 6, 17] and were introduced to the chord recognition task by Fujishima [5]. Assuming the equal-tempered scale, the chroma correspond to the set $\{\mathrm{C}, \mathrm{C}^\sharp, \mathrm{D}, \ldots, \mathrm{B}\}$ that consists of the twelve pitch spelling attributes as used in Western music notation. A chroma vector can be represented as a 12-dimensional vector $x = (x(1), x(2), \ldots, x(12))^T$, where $x(1)$ corresponds to chroma C, $x(2)$ to chroma $\mathrm{C}^\sharp$, and so on. Normalized chroma-based features indicate the short-time energy distribution among the twelve chroma and closely correlate to the harmonic progression of the underlying piece. This is the reason why basically every chord recognition procedure relies on some type of chroma feature.

There are many ways for computing chroma features. For example, the transformation of an audio recording

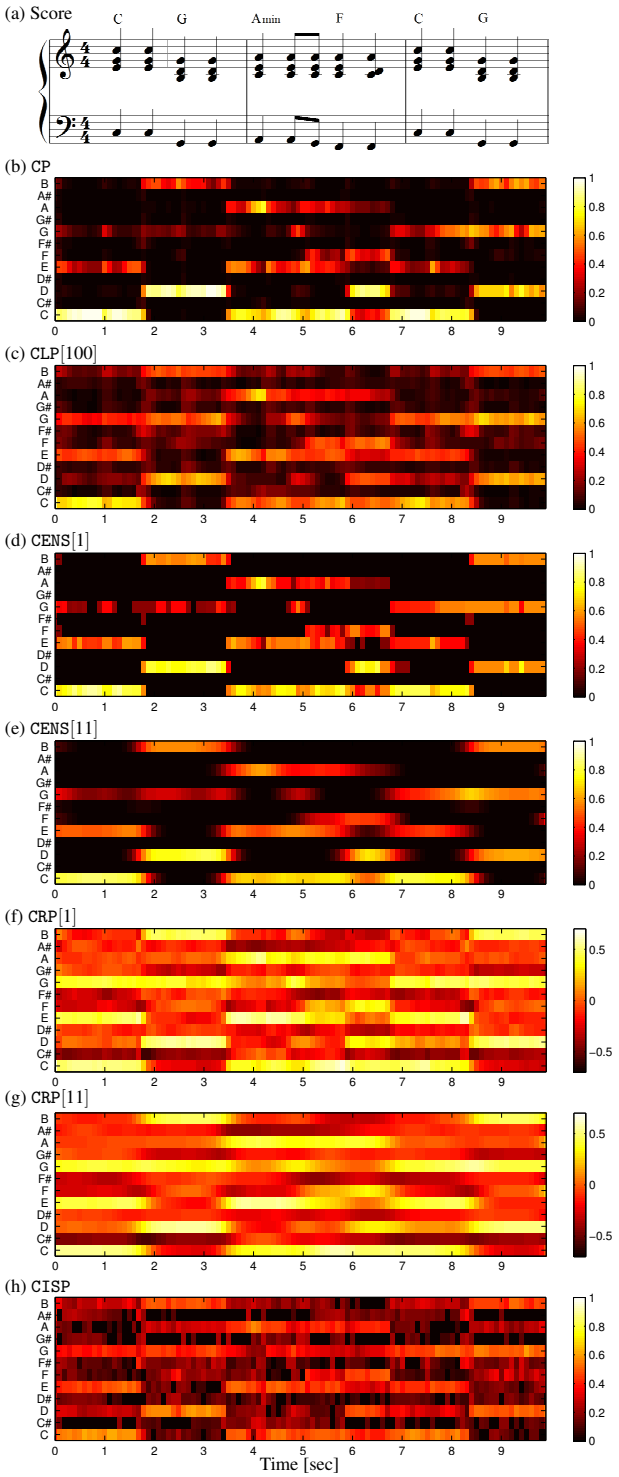

Fig. 2: Score and various feature representations of the first 10 seconds (corresponding to the first three measures) of *Let it Be* by The Beatles.

into a chroma representation (or chromagram) may be performed either by using short-time Fourier transforms in combination with binning strategies [1] or by employing suitable multirate filter banks [17]. Furthermore, the properties of chroma features can be changed by introducing suitable post- and pre-processing steps modifying spectral, temporal, and dynamical aspects. This leads to a large number of feature types which can behave quite differently depending on the subsequent analysis task. In this section, we summarize the types of chroma features that will be used in our subsequent experiments. Note that there are many more chroma variants. However, our selection covers interesting variants that demonstrate the importance of the feature extraction step.

3.1. Pitch Features

As basis for the chroma feature extraction, we first decompose a given audio signal into 88 frequency bands with center frequencies corresponding to the pitches $A0$ to $C8$ (MIDI pitches $p = 21$ to $p = 108$). For deriving this decomposition, we use a multirate filter bank consisting of elliptic filters as described in [17]. Then, for each subband, we compute the short-time mean-square power (i. e., the samples of each subband output are squared) using a rectangular window of a fixed length and an overlap of 50 %. In the following, we use a window length of 200 milliseconds leading to a feature rate of 10 Hz (10 features per second). The resulting features, which we denote as `Pitch`, measure the local energy content of each pitch subband and indicate the presence of certain musical notes within the audio signal, see [17] for further details. To account for tuning problems, we employ a tuning strategy similar to [6]. To this end, one computes an average spectral vector and estimates the tuning deviation parameter from the maximum spectral coefficient. This tuning deviation parameter is then used to suitably shift the center frequencies of the subband-filters of the above multirate filter bank. A similar approach is described in [19].

3.2. CP Feature

From the `Pitch` representation, one can obtain a chroma representation by simply adding up the corresponding values that belong to the same chroma. To archive invariance in dynamics, we normalize each chroma vector with respect to the Euclidean norm. The resulting features are referred to as *Chroma-Pitch* denoted by `CP`, see Fig. 2(b).

3.3. CLP Features

To account for the logarithmic sensation of sound inten-

sity [14, 25], one often applies a logarithmic compression when computing audio features [11]. To this end, the local energy values e of the pitch representation are logarithmized before deriving the chroma representation. Here, each entry e is replaced by the value $\log(\eta \cdot e + 1)$, where η is a suitable positive constant. Then, the chroma values are computed as explained in Section 3.2. The resulting features, which depend on the compression parameter η, are referred to as *Chroma-Log-Pitch* denoted by $\texttt{CLP}[\eta]$, see Fig. 2(c).

3.4. CENS Features

Adding a further degree of abstraction by considering short-time statistics over energy distributions within the chroma bands, one obtains CENS (Chroma Energy Normalized Statistics) features, which constitute a family of scalable and robust audio features. These features have turned out to be very useful in audio matching and retrieval applications [20, 13]. In computing CENS features, a quantization is applied based on logarithmically chosen thresholds. This introduces some kind of logarithmic compression similar to the $\texttt{CLP}[\eta]$ features. Furthermore, these features allow for introducing a temporal smoothing. Here, feature vectors are averaged using a sliding window technique depending on a window size denoted by w (given in frames) and a downsampling factor denoted by d, see [17] for details. In the following, we do not change the feature rate and consider only the case $d = 1$ (no downsampling). Therefore, the resulting feature only depends on the parameter w and is denoted by $\texttt{CENS}[w]$, see Fig. 2(d) and Fig. 2(e).

3.5. CRP Features

To boost the degree of timbre invariance, a novel family of chroma-based audio features has been introduced in [18]. The general idea is to discard timbre-related information in a similar fashion as pitch-related information is discarded in the computation of mel-frequency cepstral coefficients (MFCCs). Starting with the `Pitch` features, one first applies a logarithmic compression and transforms the logarithmized pitch representation using a DCT. Then, one only keeps the upper coefficients of the resulting pitch-frequency cepstral coefficients (PFCCs), applies an inverse DCT, and finally projects the resulting pitch vectors onto 12-dimensional chroma vectors. These vectors are referred to as CRP (Chroma DCT-Reduced log Pitch) features. The upper coefficients to be kept are specified by a parameter $p \in [1:120]$. In our experiments, we use $p = 55$. Furthermore, similar to the $\texttt{CENS}[w]$ features, we apply temporal smoothing by introducing a window parameter w that is used to average the CRP features in a band-wise fashion. The resulting features are denoted by $\texttt{CRP}[w]$, see Fig. 2(f) and Fig. 2(g).

3.6. CISP Features

Finally, we use a chroma type, where the tonal components are enhanced and the spectral resolution is increased by considering instantaneous frequencies. These features were originally introduced by Ellis and have been used in the chord recognition context as well as for cover song identification [4]. The basis for these features is a spectrogram.[1] To enhance the spectral resolution, the instantaneous frequency for each coefficient is estimated exploiting the phase information. Furthermore, based on the instantaneous frequencies, a separation of noise and harmonic components is performed and only harmonic components are preserved. Finally, to account for tuning deviations, the mapping of spectral coefficients to chroma bins is globally adjusted by up to ± 0.5 semitones to minimize the deviations of the instantaneous frequency values from the chroma bin centers using a histogram-based technique. To obtain the final features, denoted by `CISP`, adjacent frames are averaged in 100 ms windows to yield a feature rate of 10 Hz, see Fig. 2(h).

4. EXPERIMENTS

In this section, we examine the behavior of the four chord labeling procedures in dependence on the underlying feature types and associated parameter settings. We start by describing the experimental setup (including the data collection and evaluation measure) and then report on various series of experiments.

4.1. Experimental Setup

In our experiments, we use a collection of Beatles songs, which is a widely used benchmark dataset with publicly available ground-truth chord annotations [15]. Although this dataset is limited to only one artist, the results still show certain tendencies of the chord recognition accuracies. The collection, which we denote as $\mathscr{D}$, consists of 180 songs. We further partition $\mathscr{D}$ into three subcollections $\mathscr{D}_k$, $k \in \{1,2,3\}$, by first ordering the recordings alphabetically according to the songs' titles, and

[1] In our experiments, we use an implementation available in the ISP toolbox `http://kom.aau.dk/project/isound/`. Here, discrete Fourier transforms are calculated over windowed frames of length 93 ms with 75% overlap. Consequently, each frame corresponds to 23 ms of the audio and each coefficient covers a frequency range of 10.8 Hz.

then by putting the first 60 recordings into $\mathscr{D}_1$, the second 60 recordings into $\mathscr{D}_2$, and the last 60 recordings into $\mathscr{D}_3$.

The original annotations supplied by Harte [9] were reduced to the 24 chord labels following the widely spread convention of the MIREX Audio Chord Estimation task.[2] Here, only the first two intervals of each chord are considered, where augmented chords are mapped to major chords and diminished chords to minor chords. In some cases, there are passages where no meaningful chord information exists. Such regions are annotated as "N" and are left unconsidered in our evaluation (i. e., having no influence on the recognition accuracy).

In our evaluation, we first quantize and segment the chord annotations to match the frames being specified by the feature extraction step. The evaluation is then performed framewise using standard precision and recall measures by comparing the automatically generated labels with the reference labels. More precisely, a reference label is considered *correct* (C) if it agrees with the computed label, otherwise it is called a *false negative* (FN). Each incorrectly computed label is called a *false positive* (FP), see also Fig. 1(d). From this one obtains precision, recall, and F-measure defined by

$$\mathrm{P} = \frac{\mathrm{C}}{\mathrm{C}+\mathrm{FP}}, \quad \mathrm{R} = \frac{\mathrm{C}}{\mathrm{C}+\mathrm{FN}}, \quad \mathrm{F} = \frac{2\cdot\mathrm{P}\cdot\mathrm{R}}{\mathrm{P}+\mathrm{R}} \tag{5}$$

for each song.

In our evaluation, we employ a 3-fold cross validation. Here, two of the three sub-collections are used to train the recognizer that is then tested on the remaining one. F-measure values are averaged over all songs of the respective sub-collection $\mathscr{D}_k$. The final F-measure for the overall dataset $\mathscr{D}$ is the mean of the values obtained for the three sub-collections.

For determining the averaged templates to be used in $\mathtt{T}^{\mathtt{a}}$ as well as μ and Σ to be used in GP and HMM, we revert to the observation by Goto [7] that the twelve cyclic shifts of a 12-dimensional chroma vector correspond to the twelve possible transpositions. Therefore, exploiting the reference chord labels, we first transpose all chroma features to C or Cm, then determine the models for these two chords, and finally obtain models for all 24 chords by suitably transposing the C and Cm models. This procedure guarantees the same amount of training data for all major and minor chords, respectively.

[2] http://www.music-ir.org/mirex/wiki/2010:Audio_Chord_Estimation

To generate the transition matrix Ω, we first determine for each frame the corresponding reference label. Then, for all $\lambda_i, \lambda_j \in \Lambda$ we define the transition probabilities $\Omega(\lambda_i, \lambda_j) = \frac{C(\lambda_i,\lambda_j)}{\sum_{\lambda_k\in\Lambda} C(\lambda_i,\lambda_k)}$. Here, $C(\lambda_i, \lambda_j)$ specifies the number of chord transitions from label λ_i to the label λ_j, and $\sum_{\lambda_k\in\Lambda} C(\lambda_i, \lambda_k)$ serves as a normalization counting the transition from λ_i to all labels $\lambda_k \in \Lambda$ including itself.

4.2. Dependency on Feature Type

In a first experiment, the dependency of the chord recognition results on the underlying feature type is investigated. Fig. 3 summarizes the results of the evaluation for the five different feature types in combination with the four recognizers. In this experiment, we use the compression parameter $\eta = 100$ for $\mathtt{CLP}[\eta]$ and the window parameters $w = 1$ and $w = 11$ for $\mathtt{CENS}[w]$ and $\mathtt{CRP}[w]$. The role of these parameters is further analyzed in Section 4.3 and Section 4.4.

To better understand the influence of the 3-fold cross validation used in our experiments, Fig. 3(a)-(c) shows the recognition accuracies for the three folds independently. Averaging over the results of the three folds, one obtains the final results of the cross validation shown in Fig. 3(d). The F-measure values for the different parts of $\mathscr{D}$ are very consistent, e. g., using CP together with HMM leads to $\mathrm{F} = 0.531$ for $\mathscr{D}_3$ (Fig. 3(a)), $\mathrm{F} = 0.528$ for $\mathscr{D}_2$ (Fig. 3(b)), $\mathrm{F} = 0.522$ for $\mathscr{D}_1$ (Fig. 3(c)), and in average $\mathrm{F} = 0.527$ for the entire dataset $\mathscr{D}$ (Fig. 3(d)). This indicates that the partition and selection of training and testing data only has a marginal influence on the overall chord recognition results for this particular dataset.

In the following experiments, we only revert to the average values of the cross validation. As Fig. 3(d) reveals, the chord recognition accuracies depend on the complexity of the respective recognizer. For example, in the case of $\mathtt{CLP}[100]$, one obtains an F-measure value of $\mathrm{F} = 0.553$ for the basic binary template-based method $\mathtt{T}^{\mathtt{b}}$. Considering training data to learn averaged templates, the accuracy of $\mathtt{T}^{\mathtt{a}}$ is increased to $\mathrm{F} = 0.610$. Further adding covariance information as used in GP gives only slight improvements ($\mathrm{F} = 0.615$). However, when using the most advanced method HMM one gets the highest accuracy of $\mathrm{F} = 0.725$. The reason for this is that HMM introduces a context-aware smoothing in the classification stage. Considering the temporal context of chords leads to better results in comparison to the methods working in a purely framewise fashion.

Our results also reveal that the chord recognition quality

(a)

	T^b	T^a	GP	HMM
CP	0.449	0.427	0.434	0.531
CLP[100]	0.540	0.607	0.616	0.716
CENS[1]	0.458	0.435	0.439	0.581
CENS[11]	0.534	0.552	0.562	0.633
CRP[1]	0.523	0.573	0.573	0.708
CRP[11]	0.607	0.659	0.661	0.712
CISP	0.426	0.482	0.505	0.707

(b)

	T^b	T^a	GP	HMM
CP	0.476	0.416	0.423	0.528
CLP[100]	0.566	0.610	0.612	0.720
CENS[1]	0.474	0.425	0.420	0.588
CENS[11]	0.565	0.557	0.567	0.650
CRP[1]	0.532	0.592	0.591	0.729
CRP[11]	0.615	0.676	0.678	0.733
CISP	0.436	0.484	0.501	0.721

(c)

	T^b	T^a	GP	HMM
CP	0.456	0.412	0.408	0.522
CLP[100]	0.552	0.613	0.618	0.738
CENS[1]	0.441	0.430	0.413	0.583
CENS[11]	0.540	0.552	0.543	0.631
CRP[1]	0.529	0.583	0.577	0.711
CRP[11]	0.615	0.665	0.657	0.714
CISP	0.425	0.480	0.490	0.713

(d)

	T^b	T^a	GP	HMM
CP	0.460	0.418	0.421	0.527
CLP[100]	0.553	0.610	0.615	**0.725**
CENS[1]	0.458	0.430	0.424	0.584
CENS[11]	0.546	0.554	0.557	0.638
CRP[1]	0.528	0.583	0.581	0.716
CRP[11]	**0.612**	**0.667**	**0.665**	0.720
CISP	0.429	0.482	0.499	0.714

(e)

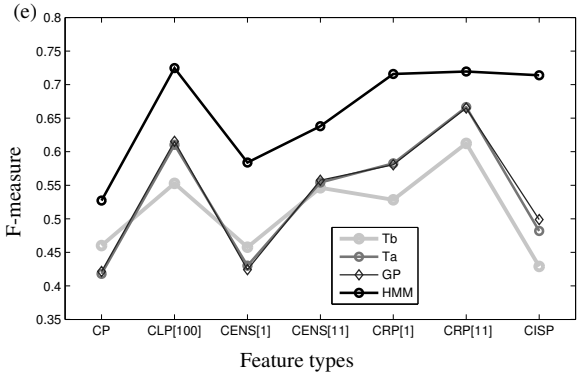

Fig. 3: Dependency of recognition rate on feature type using **(a):** Training: $\mathscr{D}_1 \cup \mathscr{D}_2$, Test: $\mathscr{D}_3$, **(b):** Training: $\mathscr{D}_1 \cup \mathscr{D}_3$, Test: $\mathscr{D}_2$, **(c):** Training: $\mathscr{D}_2 \cup \mathscr{D}_3$, Test: $\mathscr{D}_1$, and **(d):** 3-fold cross validation on $\mathscr{D}$ (averaged over (a),(b) and (c)). **(e):** Visual representation of (d).

substantially depends on the used feature type and implementation details. For example, using the most basic feature CP results in very low F-measure values (e. g., F = 0.527 for CP with HMM) regardless of which recognizer is used. However, simply applying a logarithmic compression enhancing weaker components of the feature leads to a significant increases in F-measure (e. g., F = 0.725 for CLP[100] with HMM). We will further analyze the effect of the compression parameter η in Section 4.3.

CENS[1] shows a very similar behavior as CP ($w = 1$ actually disables the temporal smoothing on the feature side). This indicates that the internal quantization of these features is not beneficial for chord recognition. However, when applying a temporal smoothing by setting the window parameter $w = 11$ (corresponding to one second) the recognition accuracy significantly increases for all recognizers. This effect is even noticeable for HMM which already involves a smoothing in the classification step (F = 0.584 for CENS[1] and F = 0.638 for CENS[11]). We will further investigate the choice of the window parameter w in Section 4.4.

CRP[1] is designed to boost timbre invariance. These features already incorporate an internal logarithmic compression leading to similar results as for CLP[100] (F = 0.716 for CRP[1] with HMM). Further adding temporal smoothing on the feature side, the F-measure increases to F = 0.720 for CRP[11] with HMM. In particular, these features lead to high F-measures, even in the case of the simple framewise recognizers (e. g., F = 0.612 in the case of T^b). Here, a carefully designed feature seems to lessen the influence of the recognizer on the chord recognition accuracy, see also Fig. 3(e) for a visual representation of the recognition results.

CISP attempts to emphasize harmonic components of the signal. This should improve the chord recognition quality for all recognizers. However, in practice, CISP shows a special behavior. On the one hand, using CISP results in high F-measure values in combination with HMM (F = 0.714). On the other hand, combining CISP features with any of the framewise recognizers T^b, T^a, and GP results in very low F-measure values (e. g., F = 0.429 for T^b). Here, one reason is that for this feature type the intensities of chroma bands corresponding to chord notes are only slightly more pronounced than those corresponding to non-chord notes, see Fig. 2(h). In general, the ratio of chroma intensities of chord notes to those of non-chord notes seems to have a large influence on the chord recognition results. In particular the frame-wise recognizers tend to be very sensitive to this ratio. Here, high ratios (as in CP and CENS[w]) as well as low ratios (as in CISP) lead to poor recognition results. HMM, however, is able to compensate for the low intensity ratios

	$\mathtt{T}^{\mathtt{b}}$	$\mathtt{T}^{\mathtt{a}}$	GP	HMM
CP	0.460	0.418	0.421	0.527
CLP[1]	0.508	0.511	0.522	0.643
CLP[10]	0.544	0.578	0.586	0.700
CLP[100]	**0.553**	**0.610**	0.615	0.725
CLP[1000]	0.541	0.606	**0.620**	**0.734**
CLP[10000]	0.521	0.579	0.608	0.730

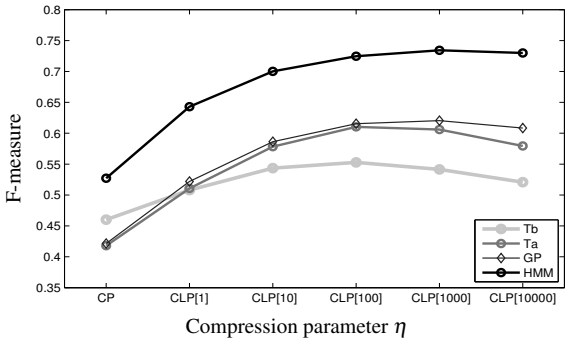

Fig. 4: Dependency of recognition rate on compression parameter η for CLP$[\eta]$ (using 3-fold cross validation on $\mathscr{D}$).

of CISP, but not for the high intensity ratios of CP and CENS$[w]$, see Fig. 3(e).

The results of the experiments discussed in this section show that the choice of the chroma feature has a significant influence on the different recognition procedures. Even the most advanced recognizer HMM has a substantial dependence on the underlying feature type. Note that using an HMM-based recognizer in combination with a poor choice of chroma feature leads to results of lower quality than using a basic recognizer with a good feature (e. g., F = 0.527 for HMM with CP but F = 0.553 for $\mathtt{T}^{\mathtt{b}}$ with CLP[100]). In particular, a logarithmic compression of the intensities as well as a temporal smoothing on the feature side have a beneficial effect, regardless of the recognizer used.

4.3. Dependency on Logarithmic Compression

In this section, we further investigate the role of the logarithmic intensity compression of the chroma features. Fig. 4 shows the chord recognition results for CP (no compression) and CLP$[\eta]$ for $\eta \in \{1, 10, 100, 1000, 10000\}$. The experiments show that logarithmic compression (or similar enhancement procedures such as spectral whitening) is an essential step in all chord recognition procedures.

For example, using HMM with CP one obtains F = 0.527. Simply applying a logarithm one gets F = 0.643 for CLP[1]. Enlarging the compression parameter η steadily increases the chord recognition accuracy, e. g., F = 0.725 for $\eta = 100$ and HMM. Here one reason for this effect is that weak spectral components, which are often relevant in view of the perception of harmony, are enhanced by the compression. See also Fig. 2(b) and Fig. 2(c) for illustration of this effect. However, for very large compression factors such as $\eta = 10000$, the chord recognition accuracy decreases (F = 0.730). Here, the enhancement of irrelevant noise-like components outweigh the harmonically relevant components.

For $\mathtt{T}^{\mathtt{a}}$ and GP, the logarithmic compression has the same significant effect on the recognition accuracy as for HMM. However, $\mathtt{T}^{\mathtt{b}}$ does not benefit from the logarithmic compression in the same way as the other methods do, see the visual representation in Fig. 4. One reason for this is that the weaker components enhanced by the compression typically correspond to harmonics. Higher harmonics, however, are not taken into account by idealized binary templates as used in $\mathtt{T}^{\mathtt{b}}$, whereas $\mathtt{T}^{\mathtt{a}}$, GP, and HMM, however, adapt to the harmonics in the training stage.

4.4. Dependency on Smoothing

In this section we continue the investigation of the temporal smoothing of the features controlled by the window parameter w. Fig. 5 and Fig. 6 show the chord recognition accuracies for different choices of the window parameter $w \in \{1, 3, 5, \ldots, 25\}$ for CENS$[w]$ and CRP$[w]$, respectively, using the four chord recognizers.

As the experiments in Section 4.2 already revealed, temporal smoothing is an essential step for the framewise chord recognition procedures ($\mathtt{T}^{\mathtt{b}}$, $\mathtt{T}^{\mathtt{a}}$, and GP) yielding significant improvements. This observation is confirmed by the results in Fig. 5. For example, using $\mathtt{T}^{\mathtt{a}}$ with CENS$[w]$ setting $w = 1$ (no temporal smoothing) results in F = 0.430. Enlarging w, the recognition accuracy gradually increases and reaches a value of F = 0.599 for $w = 23$ (corresponding to 2.3 sec). Further enlarging w leads to a decrease in accuracy again. For CRP$[w]$ one observes a very similar effect, see Fig. 6. The reason for this notable improvement is that smoothing removes temporal fluctuations and local outliers in the features. On the other hand, however, smoothing reduces the temporal resolution and may prevent the recognizers to detect chords of short durations. For this particular dataset, a smoothing window corresponding to roughly two seconds of the audio turns out to be the best trade-off between increased robustness to outliers and decreased temporal resolution. This trade-off, however, is data-dependent and depends on the chord change rate of the audio material.

	T^b	T^a	GP	HMM
CENS[1]	0.458	0.430	0.424	0.584
CENS[3]	0.484	0.472	0.468	0.592
CENS[5]	0.501	0.499	0.495	0.599
CENS[7]	0.518	0.521	0.518	0.609
CENS[9]	0.533	0.540	0.538	0.617
CENS[11]	0.546	0.555	0.554	0.623
CENS[13]	0.557	0.568	0.567	0.626
CENS[15]	0.566	0.578	0.578	0.628
CENS[17]	0.573	0.586	0.587	0.631
CENS[19]	0.578	0.592	0.593	**0.631**
CENS[21]	0.582	0.596	0.597	0.631
CENS[23]	0.585	**0.599**	**0.601**	0.629
CENS[25]	**0.587**	0.583	0.588	0.617

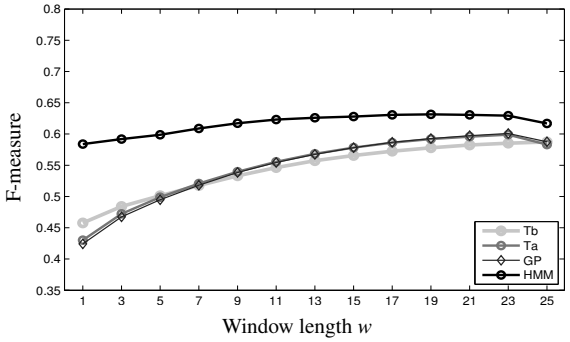

Fig. 5: Dependency of recognition rate on window parameter w for CENS$[w]$ (using 3-fold cross validation on $\mathscr{D}$).

	T^b	T^a	GP	HMM
CRP[1]	0.528	0.583	0.581	0.716
CRP[3]	0.556	0.611	0.611	0.716
CRP[5]	0.575	0.631	0.632	0.718
CRP[7]	0.591	0.646	0.649	0.718
CRP[9]	0.603	0.658	0.662	0.720
CRP[11]	0.612	0.666	0.671	**0.720**
CRP[13]	0.620	0.673	0.679	0.718
CRP[15]	0.625	0.677	0.685	0.718
CRP[17]	0.630	0.679	0.689	0.716
CRP[19]	0.632	0.680	0.691	0.714
CRP[21]	**0.633**	**0.680**	**0.692**	0.712
CRP[23]	0.632	0.679	0.692	0.709
CRP[25]	0.631	0.676	0.676	0.691

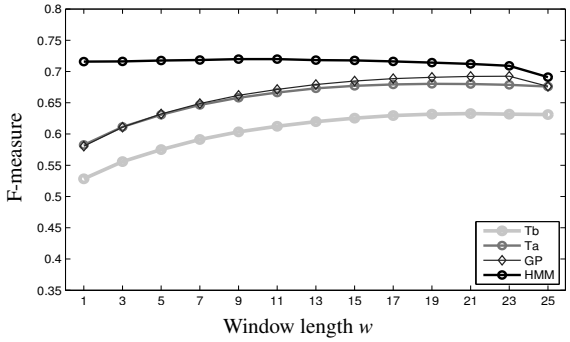

Fig. 6: Dependency of recognition rate on window parameter w for CRP$[w]$ (using 3-fold cross validation on $\mathscr{D}$).

In the case of HMM, temporal smoothing of the features has a less significant effect on the chord recognition accuracy. This recognizer already incorporates a kind of context-aware smoothing in the classification stage. Using HMM, as shown in Fig. 5, the combination of two conceptually different smoothing strategies only slightly improves the recognition rates from F = 0.584 for CENS[1] to F = 0.631 for CENS[19]. For CRP$[w]$, as shown in Fig. 6, the improvements are marginal. But even in this case adding some temporal smoothing on the feature side does not worsen the chord recognition quality (e. g., F = 0.716 for CRP[1] and F = 0.720 for CRP[11].

Interestingly, GP with smoothing on the feature side has a similar effect as HMM without smoothing on the feature side (e. g., F = 0.692 for GP with CRP[21] and F = 0.716 for HMM with CRP[1]). This indicates that the actual choice of smoothing strategy has only a small influence on the final chord recognition rates. Similar effects are also observed in the experiments in [3].

5. CONCLUSIONS

In this paper, we analyzed the chord recognition quality of different automatic chord recognition procedures in combination with different feature types. As our experimental results showed, small differences in the implementation of the chroma variants can have a significant influence on the chord recognition accuracy. In particular, a logarithmic compression step in the chroma extraction turned out to be crucial. Furthermore, our results reveal that temporal feature smoothing plays an important role in chord recognition in particular for recognizers that work in a purely framewise fashion. The Viterbi-decoding in the HMM-based recognizer also introduces a different kind of smoothing in the classification step. The combination of the two conceptually different smoothing strategies only adds a small improvement. In summary, one can note that the Gaussian-based framewise recognizer in combination with an appropriate feature smoothing yields already good recognition rates. Exploiting musical knowledge, e.g. in the form of statistical priors as used in HMMs or more general graphical models [16], one can further improve the recognition results.

For the future, we plan to extend our evaluation to comprise larger datasets of different genres [10]. In particular for building more complex statistical models, more comprehensive datasets with chord annotations are needed. Furthermore, we plan to extend our evaluation to not

only express the chord recognition quality using simple F-measures for an entire dataset, but to investigate the quality for each song (or even bar/beat) individually. This allows for detecting error-prone passages and analyzing the underlying musical or physical effects frequently leading to chord recognition errors. Furthermore, inspecting consistencies and inconsistencies in the recognition results of different recognition strategies and features, one could get a deeper insight and better understanding of the limitations of current state-of-the-art automatic chord recognition. Another promising line of research in this direction is the development of multi-layered analysis methods that exploit the availability of multiple versions and representations of a given musical work [12].

Acknowledgement. The authors are supported by the Cluster of Excellence on *Multimodal Computing and Interaction* at Saarland University.

6. REFERENCES

[1] M. A. Bartsch and G. H. Wakefield. Audio thumbnailing of popular music using chroma-based representations. *IEEE Transactions on Multimedia*, 7(1):96–104, Feb. 2005.

[2] J. P. Bello and J. Pickens. A robust mid-level representation for harmonic content in music signals. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, London, UK, 2005.

[3] T. Cho, R. J. Weiss, and J. P. Bello. Exploring common variations in state of the art chord recognition systems. In *Proceedings of the Sound and Music Computing Conference (SMC)*, pages 1–8, Barcelona, Spain, 2010.

[4] D. P. W. Ellis and G. E. Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Honolulu, Hawaii, USA, 2007.

[5] T. Fujishima. Realtime chord recognition of musical sound: A system using common lisp music. In *Proc. ICMC*, pages 464–467, Beijing, 1999.

[6] E. Gómez. Tonal description of polyphonic audio for music content processing. *INFORMS Journal on Computing*, 18(3):294–304, 2006.

[7] M. Goto. A chorus-section detecting method for musical audio signals. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 437–440, Hong Kong, China, 2003.

[8] C. Harte and M. Sandler. Automatic chord identification using a quantised chromagram. In *Proceedings of the 118th AES Convention*, Barcelona, Spain, 2005.

[9] C. Harte, M. Sandler, S. Abdallah, and E. Gómez. Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, London, GB, 2005.

[10] H. Kaneko, D. Kawakami, and S. Sagayama. Functional harmony annotation database for statistical music analysis. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR): Late Breaking session*, 2010.

[11] A. P. Klapuri, A. J. Eronen, and J. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech and Language Processing*, 14(1):342–355, 2006.

[12] V. Konz, M. Müller, and S. Ewert. A multi-perspective evaluation framework for chord recognition. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR)*, pages 9–14, Utrecht, Netherlands, 2010.

[13] F. Kurth and M. Müller. Efficient Index-Based Audio Matching. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(2):382–395, Feb. 2008.

[14] B. Logan. Mel frequency cepstral coefficients for music modeling. In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, Plymouth, Massachusetts, 2000.

[15] M. Mauch, C. Cannam, M. Davies, S. Dixon, C. Harte, S. Kolozali, D. Tidhar, and M. Sandler. OMRAS2 metadata project 2009. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Kobe, Japan, 2009.

[16] M. Mauch and S. Dixon. Simultaneous estimation of chords and musical context from audio. *IEEE Trans. Audio, Speech, and Language Processing*, 18(6):1280–1289, 2010.

[17] M. Müller. *Information Retrieval for Music and Motion*. Springer Verlag, 2007.

[18] M. Müller and S. Ewert. Towards timbre-invariant audio features for harmony-based music. *IEEE Transactions on Audio, Speech, and Language Processing (TASLP)*, 18(3):649–662, 2010.

[19] M. Müller, P. Grosche, and F. Wiering. Robust segmentation and annotation of folk song recordings. In *Proceedings of the 10th International Society for Music Information Retrieval Conference (ISMIR)*, pages 735–740, Kobe, Japan, Oct. 2009.

[20] M. Müller, F. Kurth, and M. Clausen. Audio matching via chroma-based statistical features. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 288–295, 2005.

[21] L. Oudre, Y. Grenier, and C. Févotte. Template-based chord recognition: Influence of the chord types. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR)*, Kobe, Japan, 2009.

[22] H. Papadopoulos and G. Peeters. Large-scale study of chord estimation algorithms based on chroma representation and HMM. In *Content-Based Multimedia Indexing (CBMI)*, pages 53–60, 2007.

[23] A. Sheh and D. P. W. Ellis. Chord segmentation and recognition using EM-trained hidden Markov models. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Baltimore, USA, 2003.

[24] Y. Ueda, Y. Uchiyama, T. Nishimoto, N. Ono, and S. 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, 2010.

[25] E. Zwicker and H. Fastl. *Psychoacoustics, facts and models*. Springer Verlag, New York, NY, US, 1990.