IMPROVING AUDIO CHORD TRANSCRIPTION BY EXPLOITING HARMONIC AND METRIC KNOWLEDGE

W. Bas de Haas
Utrecht University
`W.B.deHaas@uu.nl`

José Pedro Magalhães
University of Oxford
`jpm@cs.ox.ac.uk`

Frans Wiering
Utrecht University
`F.Wiering@uu.nl`

ABSTRACT

We present a new system for chord transcription from polyphonic musical audio that uses domain-specific knowledge about tonal harmony and metrical position to improve chord transcription performance. Low-level pulse and spectral features are extracted from an audio source using the Vamp plugin architecture. Subsequently, for each beat-synchronised chromagram we compute a list of chord candidates matching that chromagram, together with the confidence in each candidate. When one particular chord candidate matches the chromagram significantly better than all others, this chord is selected to represent the segment. However, when multiple chords match the chromagram similarly well, we use a formal music theoretical model of tonal harmony to select the chord candidate that best matches the sequence based on the surrounding chords. In an experiment we show that exploiting metrical and harmonic knowledge yields statistically significant chord transcription improvements on a corpus of 217 Beatles, Queen, and Zweieck songs.

1. INTRODUCTION

Chord labels are an indispensable and ubiquitous aid for modern musicians. Although classically trained performers still rely mainly on printed scores, describing in high detail how a piece of music should be performed, the emergence of jazz, improvised, and popular music gave rise to the need for more flexible and abstract representations of musical harmony. This led to a notational vehicle often referred to as a *lead sheet*. A lead sheet typically contains only the melody of a composition accompanied with the essential harmonic changes denoted with chord labels. It can be considered a rather informal map that guides the performers and specifies the boundaries of the musical playground. Also, in music theory, music education, composition, and harmony analysis, chord labels have proven to be a convenient way of abstracting from individual notes in a score. Hence, these days chord labels are omnipresent: there are publishers that specialise in publishing lead sheets, and many lead sheets circulate on the internet.

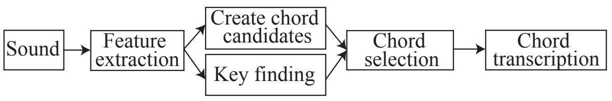

Figure 1. A schematic outline of the MPTrEE system.

The many possible applications of chord labels have sparked research focusing specifically on chords. Many Music Information Retrieval (MIR) tasks, like similarity estimation, genre detection, or query by humming, can benefit from some reduction of the raw audio signal into a manageable symbolic representation. Although much progress has been made, multiple fundamental frequency (F0) estimation, the holy grail in polyphonic music transcription, might still be considered too unreliable and imprecise for many MIR tasks. Chord transcription, which deals with transforming polyphonic audio into musically feasible symbolic annotations, has offered a welcome alternative. For example, in automatic harmony analysis [8], and similarity estimation [5], chord labels are used as primary data representation.

In this paper we present a novel system, named MPTrEE, [1] that automatically transcribes chord labels from a polyphonic musical audio source. This system is different from most other chord transcription systems, e.g. [12], in that it does not rely on statistical learning. Although machine learning has brought chord transcription (and MIR in general) many merits, we believe that there is a limit to what can be learned from musical data alone [6]. Certain musical segments can only be annotated correctly when musical knowledge not exhibited in the data is taken into account as well. Moreover, Hidden Markov Models (HMMs), frequently used to model the transitions between chords, model only the transition between a small number of subsequent chords, and have a bias towards sequences they have been trained on. Our system, on the other hand, relies on a knowledge-based model of tonal harmony. The HarmTrace [2] harmony model [4] is explicitly designed for modelling the relations between chords, also over a long time span. In this paper we show how this harmony model can be employed to improve chord transcription.

A global outline of the system is presented in Figure 1. We start by briefly reviewing some important literature in Section 2. Next, we give a complete outline of the MPTrEE

[1] (Musical) Model Propelled TRanscription of Euphonic Entities
[2] Harmony Analysis and Retrieval of Music with Type-level Representations of Abstract Chords Entities

system in Section 3. In Sections 4 and 5 we discuss the experiments and results. Finally, we conclude the paper by recapitulating the main advantages and disadvantages of the MPTREE system, and highlight some directions for future research.

Contribution. In this paper we bridge the gap between top-down symbolic music analysis and bottom-up audio feature extraction. We show that exploiting metrical position and a model of tonal harmony yields significant chord transcription improvements on 217 songs by the Beatles, Queen, and Zweieck.

2. RELATED WORK

The first computational approaches to automatic chord transcription from musical audio emerged at the end of the 1990s. The first audio chord transcription system was developed by Fujishima [3]. In general, the outline of Fujishima's system is not so different from the chord transcription systems nowadays developed and also no so different from the system presented here. First, chroma features [15] are extracted at every frame, representing the intensities of the twelve different pitch classes as found in the spectrum. Next, the chroma vectors are matched with chord profiles; in Fujishima's case this is done with an Euclidean distance. Although the used digital signal processing parameters may vary, most approaches towards automatic chord transcription use a chroma feature based representation and differ in other aspects, like chroma tuning, noise reduction, chord transition smoothing, and harmonics removal. For an elaborate review of related work on automatic chord transcription we refer to Mauch [9].

From 2008 on, chord transcription has received a considerable amount of attention in the yearly benchmarking challenge MIREX. [3] Each year, between 7 and 19 different chord transcription algorithms were evaluated. In 2008, the system of Bello and Pickens [1], which was the first to synchronise chroma vectors at every beat, performed the best. The following year, Mauch et al. [10] presented a system that gave good results by structurally segmenting a piece and combining chroma information from multiple occurrences of the same segment type. In 2010, Mauch et al. [11] improved their previous results by using an approximate note transcription technique. In 2011, the system of Ni et al. [12], using only machine learning techniques, gave comparable results.

3. SYSTEM OUTLINE

An outline of the MPTREE system is shown in the flowchart of Figure 2. First, we extract chroma features and beat locations from the audio signal and synchronise the chroma features at the beat positions (Section 3.1). The chroma features are used to estimate the global key and possible modulations in the musical audio (Section 3.4), and for creating a sequence of chord candidate lists (Section 3.2). These candidate lists contain the chords that match the chroma well a particular beat position. If there is a lot

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

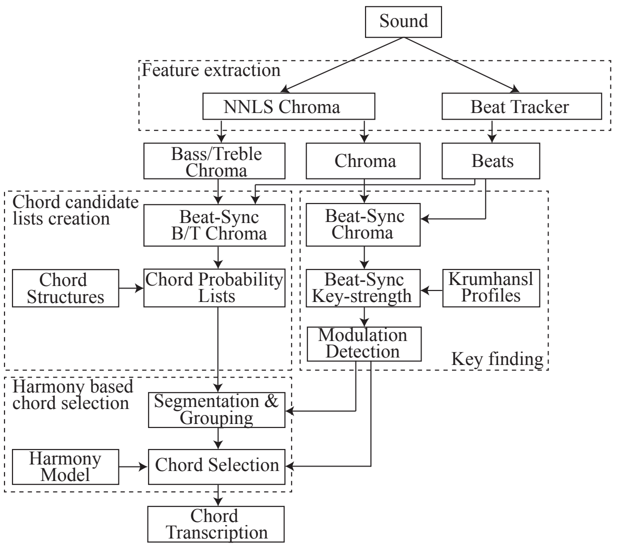

Figure 2. A schematic outline of the MPTREE system. The boxes with dotted lines denote the high level modules as outlined in Figure 1.

of uncertainty in the data, these lists might contain multiple chords; however, if there is a strong match between the spectrum and one particular chord candidate, the lists will contain a single candidate. Subsequently, the sequence of chord candidate lists is segmented (Section 3.5). Finally, the best matching sequence per segment is selected by expanding all possible sequences, and preferring the sequence with the fewest harmony errors (Section 3.6).

3.1 Feature extraction front-end

Our work depends heavily on the Vamp plugin architecture. [4] As feature extraction front-end we rely on the NNLS Chroma Vamp plugin [5] developed by Mauch [9]. The NNLS Chroma plugin transforms an audio signal into two 12-dimensional chroma vectors representing the harmonic content at each frame. The first chroma vector (*bass*) represents the low notes and emphasises the lower frequencies, while the second (*treble*) represents the higher notes, emphasising higher frequencies. The idea behind this separation is to model the prominent role of the bass note in chord transcription. We present a brief overview of the most important properties and parameters of the NNLS plugin. For specific signal processing details we refer to Mauch [9].

We use the sonic-annotator [6] (version 0.6) as Vamp host, and sample the audio tracks at 44,100 Hz. If the audio file contains two stereo channels, the mean of both channels is used for analysis. We set the sonic-annotator to use a Hann window of 16,384 samples and a hop size, i.e. the amount of samples that overlap between two subsequent frames, of 2,048 samples. Next, the spectrogram is calculated at each frame using a discrete-time Fourier transform and mapped to a spectrogram with bins that are linearly spaced in log-frequency (similar to a constant-Q transform). The NNLS

[4] http://www.vamp-plugins.org
[5] http://isophonics.net/nnls-chroma
[6] http://omras2.org/SonicAnnotator

C:Maj	1	0	0	0	1	0	0	1	0	0	0	0
D:Min	0	0	1	0	0	1	0	0	0	1	0	0
	C	C♯	D	E♭	E	F	F♯	G	G♯	A	B♭	B

Table 1. A binary chord structure of a C major and a D minor chord, which are matched against the chroma features.

				0.93 C^7			
				0.96 Am			
			0.94 G	0.97 C		0.94 Bm	
…	1.00 C	1.00 F	1.00 Gm	1.00 Em	1.00 F	1.00 B	…
…	1	2	3	4	1	2	…

Table 2. An excerpt of a sequence of chord candidate lists. The number to the left of the chord label represents its normalised Euclidean distance to the current beat aligned chroma vector. Below the candidate lists the beat position within the bar is printed.

Chroma Vamp plugin also accounts for tuning differences in the audio signal. Also, the NNLS plugin accounts for harmonics other then the F0 of chord notes by estimating which pitch activation generates an interference pattern that best matches the partials found in a spectrum.

3.2 Beat-synchronous chord probability estimation

After obtaining bass and treble chroma features, we beat-synchronise them by averaging the feature vectors between two beats. For this, we obtain a list of beat positions by using the Queen Mary, University of London, Bar and Beat Tracker plugin [2]. [7] Besides beat timestamps, this beat tracker also outputs the position of the beat inside the bar.

To estimate the probability of a particular chord sounding at a beat position, we assemble a dictionary of chords that we expect to occur in the music. A chord is represented as a binary 12-dimensional vector in which the simultaneously sounding pitch classes are denoted with a 1 (see the examples in Table 1). This allows us to model any possible chord within one octave. Currently, we use a limited chord dictionary with three chord structures: major, minor, and dominant seventh. We chose these three chords because they map nicely to the chord classes used by the HARMTRACE harmony model. In HARMTRACE, chords are categorised in four classes: major chords, minor chords, dominant seventh chords, and diminished seventh chords (see [4, Chapter 4] for details). However, because diminished seventh chords are not very common in pop music, we ignored this class in this study. The bass note of the chord is modelled with an additional 12-dimensional vector containing only one pitch for the bass note, to match the bass chroma vector as outputted by the NNLS chroma plugin. Next, we generate the chord dictionary by cyclically rotating all chord structures for all twelve semitones, yielding 48 different chord candidates, and a "no chord" structure containing only 0's.

Having a matrix of beat-synchronised bass and treble chromagrams and a chord dictionary, we estimate the probability of a chord sounding at a particular beat by calculating the Euclidean distance between the chord structures and the chroma feature. These distances are calculated for every chord candidate at every beat. Next, we sort the chord candidates by descending Euclidean distance. To obtain a relative measure of the fit between a chord candidate and the chroma vector in the range $[0, 1]$, the distances are normalised by dividing them by distance of the best matching chord candidate. In case the information in the spectrum clearly favours a certain chord candidate, the initial differences in normalised distances will be relatively large and will decrease quickly after the head position. Hence, we can use these differences as a measure of relative chord candidate preference. If this preference is very strong, the top chord candidate will be selected to represent that beat. If this preference is less pronounced, we use the HARMTRACE harmony model to decide which of the chord candidates would make most sense, harmonically. Typically, this results in a sequence of chord candidates similar to the one shown in Table 2. The selection is performed by cutting off the chord candidate list at a fixed distance. The cut-off value is an important parameter to the model, influencing both the speed and the transcription quality of the system. After some experimentation we found that a cut-off value of 0.9 gives good results.

[7] http://vamp-plugins.org/plugin-doc/qm-vamp-plugins.html\#qm-barbeattracker

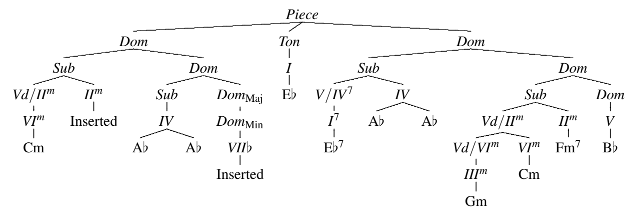

Figure 3. An excerpt of the HARMTRACE analysis of *The long and winding road* by the Beatles (of which the ground-truth chord annotations were used for parsing). The Vd/X represents a diatonic fifth succession, and a V/X^7 denotes a secondary dominant.

3.3 A model of tonal harmony

Given a list of chord candidates, we select a harmonically sensible sequence by exploiting a formal model of tonal harmony. This model, which is elaborately explained by De Haas [4], takes a sequence of symbolic chord labels as input and automatically derives a tree structure explaining the function of the chords in their tonal context. Figure 3 depicts an excerpt of the harmony analysis of *The long and winding road* by the Beatles.

Extending the ideas of Rohrmeier [13], a piece is modelled as a sequence of tonic and dominant nodes (*Ton* and *Dom*, respectively) that represent the global patterns of harmonic tension and release. Every *Dom* node can be preceded by a subordinate sub-dominant (*Sub*) building up the tension towards the dominant. Finally, a branch will always end in a scale degree node, representing the relation between the actual chord and the key of the piece, and the leaves of the tree show the actual input chord labels. On the path from functional annotation (*Ton*, *Dom*, and *Sub*) to chord label, various harmonic annotations, like secondary dominants, tritone substitutions, diatonic fifth chains, diminished seventh chord transpositions, etc., can occur, explaining the role of a chord label in its tonal context. If a sequence does not match the harmonic specification, like in the first phrase of Figure 3, an input chord label is automatically deleted or inserted to match the specification. Hence, for a sequence of chords we can always derive an automatic harmonic analysis.

3.4 Key-finding

To be able to use the HARMTRACE harmony model for the selection of chord sequences that are music theoretically realistic, we require information about the key of the piece. To fulfil this requirement, we present a key-finding algorithm inspired by the ideas of Temperley [14, Chapter 7] and Krumhansl [7, Chapter 4]. Again, for feature extraction we depend on the NNLS chroma Vamp plugin, which allows for exporting different kind of audio features. For key-finding we export a single tuned chroma feature without the NNLS pitch activation estimation.

To estimate the key of a piece and the possible modulations, we use a *key-profiles* based algorithm. A key-profile is a 12 value vector representing the stability of the twelve pitch classes relative to a given key. The values of these profiles are based on empirical measurements of Krumhansl and Kessler [7], in which subjects were asked to rate how well a pitchclass fits a previously established tonal context on a 1 to 7 scale. Given the major and minor key profiles, a key-strength table K is created. This table stores the estimated strength of all 24 keys at every beat position. The key strength is estimated by calculating the Pearson correlation coefficient, r. A value of r close to 0 indicates that there is little to no relation between the key-profile and chroma vector, whereas a value close to 1 or -1 indicates a positive or negative linear dependence between the key-profile and the chroma vector, respectively.

Matching the key-profiles at every beat does not yet result in the desired key assignment; because beat size segments are rather small, key changes occur too often. To overcome this problem, we use a simple dynamic programming algorithm based on the algorithm in [14] to smooth the key changes. We create a table M storing the cumulative key-strength of every key at every beat, and minimise the number of modulations. Switching to another key, i.e. changing the column j in M, is penalised. This behaviour is captured in the following recursive formula:

$$\begin{array}{rcl} M[0,j] & = & K[0,j] \\ M[i,j] & = & \max\left\{\begin{array}{l} M[i-1,j]+K[i,j], \\ M[i-1,j]+K[i,k]+p, \end{array}\right. \\ & & \text{where } \{k \mid \forall x : K[i,x] \leqslant K[i,k]\} \end{array}$$

Here, M stores the cumulative key-strength for every ith beat and every jth key. Similarly, K stores the correlation between every ith chroma vector and jth key profile. k denotes the index of the best matching key at beat i. The parameter p specifies the modulation penalty. We found a value of 1 for p to give good results. Finally, we obtain the definite key assignment by keeping track of the maximum cumulative key-strength at every beat, and constraining the key segments to be at least 16 beats long.

3.5 Segmentation and grouping

Given a sequence of chord candidate lists, we analyse all possible chords sequence combinations with HARMTRACE and select the simplest analysis with the least amount of errors. However, the number of possible combinations grows exponentially with the number of candidate lists. Hence, it is vital to split our sequence of chord candidate lists into smaller, musically meaningful segments.

Also, from a musical point of view, it is unrealistic to expect chords to change at every beat. Therefore, we reduce the space of analysed sequences by merging subsequent chord candidate lists that contain the same chords. The candidate lists are merged by taking the intersection between two adjacent lists, if the intersection contains at least one chord. In this procedure we take into account that chords are more likely to change on strong metrical positions by adding two additional constraints: when two candidate lists are merged, the first and leftmost list must be positioned either at the first or third beat of the bar. The merging procedure is executed sequentially, and merged candidate lists can be merged again with the subsequent chord candidate list. For example, if the candidate lists at beat position 1 and 2 are merged, the merged list can again merge with beat position 3 if the intersection contains at least one chord. Finally, the probabilities of the merged candidate lists are summed, and the lists are sorted by descending probability.

Subsequently, the sequence of chord candidate lists is segmented on the basis of the estimated key, resulting in segments that contain only a single key assignment. Nevertheless, these sequences are still rather long for analysing all possible combinations. Within the HARMTRACE harmony model, a piece is viewed as a collection of tonics (*Ton*) and dominants (*Dom*) nodes. Hence, from the parsing point of view, splitting a chord sequence into segments that match the subtrees rooted by a *Ton* or *Dom* seems natural. Because local information about the key is available, we can calculate the key-relative scale degrees of the chords and split a sequence at every beat where a *I* or *V* scale degree occurs in a chord candidate list. This gives us sequences that are short, but still musically meaningful. In case our key-finding method is off the mark, and we still end up with a rather long sequence, we enforce the sequences to be no longer than 12 chords, and expand into no more than 30 different candidate sequences.

3.6 Chord selection by parsing

Now that we have access to both a segmented sequence of chord candidate lists and local key information, we are ready to apply the HARMTRACE harmony model. For every segment we parse all possible combinations of chord sequences and select the sequence that has the lowest error-ratio. The error-ratio is the number of insertions and deletions of the error-correcting parser divided by the number of chords. When two sequences have the same error-ratio, we select the most simple solution by picking the sequence that returns the smallest parse tree. In case the parse tree size is also identical, we select the sequence returning the parse tree of least depth.

The harmony model used in MPTREE is not the exact same model as the one described by De Haas [4, Chapter 4]. The original HARMTRACE harmony model exhibits a bias towards jazz harmony. Therefore, we made several adaptations, but the majority of the specifications remained unchanged. The original harmony model was designed to do an automatic harmonic analysis of a chord sequences and could explain a vast amount of exotic harmonic phenomena. Within the pop dataset on which the MPTREE system is evaluated, some of these specifications are unnecessary. Hence, we remove some of the specifications accounting for jazz-specific chord changes. [8] Furthermore, we add two rules that account for some blues phenomena. [9] The Haskell code of both the HARMTRACE models and the MPTREE system is freely available online. [10]

4. EXPERIMENTS

To measure the effect of the various modules on chord transcription performance we evaluate four different versions of the system described before. The simplest system, named SIMPLE, always selects the chord that best matches the bass and treble chroma vectors. The second system, GROUP, also picks the best matching chord candidate, but does incorporate the grouping as described in Section 3.5. The third system is the full MPTREE system, including key-finding. Finally, we include a fourth system, MPTREE$^{\text{key}}$, to measure the effect of the key-finding. MPTREE$^{\text{key}}$ does not use key-finding, but instead uses ground-truth key annotations [10]. All systems are implemented in the functional programming language Haskell and compiled using the Glasgow Haskell Compiler, version 7.4.1.

We evaluate the quality of an automatic chord transcription by comparing it to a transcription of the same piece made by a human expert. We evaluate our system on 179 songs from 12 Beatles albums, 20 Queen songs, and 18 Zweieck songs [10]. [11] The chord vocabulary for the MIREX evaluation is limited to 24 major and minor chords augmented with a "no chord" label, to be used for silence or non-harmonic passages, for instance. In accordance with MIREX, we also use these 25 classes. The translation from the three chord classes of HARMTRACE to major and minor chords is trivial: chords of the major and dominant class are classified as major, and chords of the minor class are classified as minor.

Typically in MIREX, the *relative correct overlap* (RCO) is used as a measure of transcription accuracy. The RCO is defined as the total duration of correctly overlapping chords divided by the total duration of the song. Both the ground-truth and the automatic chord transcription consist of a chord label and an accompanying onset and offset time-stamp. We approximate the RCO by sampling both the ground-truth and the automatic annotations every 10ms and dividing the number of correctly annotated samples by the total number of samples.

[8] The specifications with numbers 20, 21, and 22 were removed.

[9] Allowing dominant seventh chords at the *IV* and *I* scale degree to function respectively as sub-dominant and tonic, to be precise.

[10] `http://hackage.haskell.org/package/HarmTrace-2.0`

[11] `http://isophonics.net/content/reference-annotations`

	SIMPLE	GROUP	MPTREE	MPTREE$^{\text{key}}$
RCO	0.688	0.736	0.739	0.741
Running time	$5m1s$	$5m9s$	$10m23s$	$7m37s$

Table 3. The relative correct overlap and the running times for the four evaluated chord transcription systems.

5. RESULTS

We have compared the MPTREE, MPTREE$^{\text{key}}$, GROUP, and the baseline SIMPLE system on 217 songs of the Beatles, Queen, and Zweieck. All runs were performed on the same Intel Core i7-2600 Processor running at 3.40GHz. The measured differences in RCO and running times are displayed in Table 3.

We tested whether the differences in RCO are statistically significant by performing a non-parametric Friedman test [12] with a significance level of $\alpha = 0.05$. The Friedman ANOVA is chosen because the underlying distribution of the RCO data is unknown, and, in contrast to a regular ANOVA, the Friedman does not assume a specific distribution of variance. To determine which pairs of measurements differ significantly, a post-hoc Tukey HSD test is conducted. Within the MIREX challenge the same statistical procedure is followed. There are significant differences between the four systems, $\chi^2(3, N = 217) = 339$, $p < 0.0001$. Not all pairwise differences between systems are statistically significant; the difference between GROUP and MPTREE, and between MPTREE and MPTREE$^{\text{key}}$ are not significant. All other pairwise differences (including the difference between MPTREE$^{\text{key}}$ and GROUP) are statistically significant.

Considering the differences between the MPTREE$^{\text{key}}$ system and the SIMPLE and GROUP systems, we can conclude that using the HARMTRACE harmony model for chord candidate selection improves chord transcription performance, if correct key information is available. This difference in performance cannot be attributed to the merging function described in Section 3.5 alone. However, clearly a lot of the performance gain must be attributed to this merging function. Hence, we can conclude that forcing chords not to change often and mainly at strong metrical positions improves transcription performance. Although the difference between MPTREE an MPTREE$^{\text{key}}$ is not statistically significant, the errors in the key-finding do have an effect on the transcription performance, since the difference between GROUP and MPTREE is not, but the difference between GROUP and MPTREE$^{\text{key}}$ is statistically significant.

The running times as shown in Table 3 exclude the time

[12] All statistical tests were performed in Matlab 2011a.

taken by the Vamp feature extraction plugins. The results show a trade-off between transcription performance and computation time. However, the running times are acceptable, less than 3 seconds per song on average.

6. DISCUSSION

In this paper we aim at bridging the gap between bottom-up audio feature extraction and top-down symbolic music analysis. We demonstrate in a proof-of-concept how automatic chord transcription can be improved by using domain-specific knowledge about the metrical structure and tonal harmony. For feature extraction we rely on the NNLS chroma and the Bar and Beat Tracker Vamp plugin. We show that preferring harmonically valid combinations of chords yields better chord transcriptions than just picking the best matching chord at each beat, even after smoothing the chord changes with a merging function. This result is good, especially if we consider that we have only connected the different technologies without extensively tuning their parameters.

It is difficult to compare the results of this paper with current the state-of-the-art in an absolute manner; this must be done in a next iteration of the MIREX challenge. The dataset used in this paper closely resembles the one used in MIREX in 2010 and 2011. However, although the same ground-truth is used, many different remastered editions of the Beatles and Queen songs exist, and some editions are known to deviate from these ground-truth annotations. We used the LabROSA script to improve the alignment between our Beatles corpus and the ground-truth, [13] but it is hard to tell whether the results in Section 4 have been influenced by remastering artifacts. However, if this is the case, the results of all compared systems are affected equally.

In the 2011 edition of MIREX, all systems were evaluated as described in Section 4, yielding RCO values between 0.126 and 0.829, and a deliberately over-fitted result yielding an RCO of 0.976. Clearly, a system with a model trained in this manner will very likely perform poorly on unseen data. All algorithms that returned an RCO above 0.740 were HMM-based machine learning approaches, and it is unclear how much they have over-fitted on the used dataset. The chances that the HARMTRACE harmony model is over-fitting the used dataset are very low. After all, candidate chord sequences are not selected based on how often they occur in a training sample, but only based on whether they follow the general rules of tonal harmony. Another benefit of the knowledge-based approach presented in this article, is that we can analyse why certain chord sequences are preferred over others and reason about whether these choices are justified. An HMM remains a black box, which does not provide insights into the choices made.

Nevertheless, there is still room for improvement. Perhaps that using different signal processing parameters, or different plugins improve the results. Moreover, we expect that carefully adjusting the parameters and tailoring the modules to maximise their interoperability will result in an increase of performance. Also, Mauch et al. [9] successfully improved chord transcription performance by averaging the chroma vectors of segments that were classified as having very similar harmonies. Such a technique could possibly improve the results in the MPTREE system as well. We have shown that connecting state-of-the-art low-level feature extraction methods to high-level symbolic knowledge systems offers new capabilities to boost the analysis and retrieval of musical audio. We also expect similar combinations to be able to improve other common MIR related tasks, such as cover-song finding, music transcription, and structural analysis.

Acknowledgements We thank Anja Volk and Remco C. Veltkamp for providing comments on earlier drafts of this article. W. Bas de Haas is supported by the Netherlands Organization for Scientific Research, NWO-VIDI grant 276-35-001, and José Pedro Magalhães by EPSRC grant number EP/J010995/1.

7. REFERENCES

[1] J.P. Bello and J. Pickens. A robust mid-level representation for harmonic content in music signals. In *ISMIR Proceedings*, pages 304–311, 2005.

[2] M.E.P. Davies and M.D. Plumbley. Context-dependent beat tracking of musical audio. *Audio, Speech, and Language Processing, IEEE Transactions on*, 15(3):1009–1020, 2007.

[3] T. Fujishima. Realtime chord recognition of musical sound: A system using common Lisp music. In *ISMIR Proceedings*, pages 464–467, 1999.

[4] W.B. de Haas. *Music information retrieval based on tonal harmony*. PhD thesis, Utrecht University, 2012.

[5] W.B. de Haas, J.P. Magalhães, F. Wiering, and R.C. Veltkamp. HarmTrace: Improving harmonic similarity estimation using functional harmony analysis. In *ISMIR Proceedings*, 2011.

[6] W.B. de Haas and F. Wiering. Hooked on music information retrieval. *Empirical Musicology Review*, 5(4):176–185, 2010.

[7] C.L. Krumhansl. *Cognitive Foundations of Musical Pitch*. Oxford University Press, USA, 2001.

[8] J. P. Magalhães and W. B. de Haas. Functional Modelling of Musical Harmony—an Experience Report. In *Proceedings of the International Conference on Functional Programming*, pages 156–162, 2011.

[9] M. Mauch. *Automatic chord transcription from audio using computational models of musical context*. PhD thesis, Queen Mary University of London, 2010.

[10] M. Mauch, C. Cannam, M. Davies, S. Dixon, C. Harte, S. Kolozali, D. Tidhar, and M. Sandler. Omras2 metadata project 2009. In *ISMIR Proceedings*, 2009.

[11] M. Mauch and S. Dixon. Approximate note transcription for the improved identification of difficult chords. In *ISMIR Proceedings*, pages 135–140, 2010.

[12] Y. Ni, M. McVicar, R. Santos-Rodriguez, and T. De Bie. An end-to-end machine learning system for harmonic analysis of music. *Audio, Speech, and Language Processing, IEEE Transactions on*, 20(6):1771–1783, 2012.

[13] M. Rohrmeier. Towards a generative syntax of tonal harmony. *Journal of Mathematics and Music*, 5(1):35–53, 2011.

[14] D. Temperley. *The Cognition of Basic Musical Structures*. Cambridge, MA, MIT Press, 2001.

[15] G.H. Wakefield. Mathematical representation of joint time-chroma distributions. In *Conference on Advanced Signal Processing Algorithms, Architectures, and Implementations*, pages 637–645, 1999.

[13] `http://labrosa.ee.columbia.edu/matlab/beatles_fprint/`