Automatic Key Recognition

MUS-17

Leon Wittwer

July 5, 2017

Introduction

The key is an important characteristic of musical pieces and the annotation by hand is a challenging task. Especially nowadays, where large digital collections of music are available, a system that automatically annotates the key of songs is really useful because it is way too much effort to annotate all of this songs by humans.
As key recognition is a part of music information retrieval (MIR), having better key recognition systems enhances MIR systems. Also automated mixing is a field that profits from key recognition systems, because blending two songs together is much better when the DJ (or software in this case) chooses songs with same or related keys.
A connected field to key recognition is chord recognition, because improvement in one of the fields would also bring improvement to the other field. This is because knowing the key restricts the chords that are most likely used and on the other way with knowing the chords it is possible to estimate the key based on the known chords. There are also approaches that does both tasks at the same time. Also in the field of music perception the key plays an important role, because the key is such an important musical characteristic. Therefore key recognition systems can be used to support studies on understanding how humans perceive music.
A typical key recognition system consists of two main parts: First a feature extraction stage is utilized and then some key classification based on the extracted features is applied. The features used are mostly pitch class distributions.

This paper gives an overview on the topic of automatic key recognition and summarizes the findings of the thesis [1].

Challenges

There are many points that does make automatic key recognition a non trivial task for a computer, so a few challenges are listed and explained here. The first challenge for automatic key recognition systems is that it can even be a challenging task for humans. So when the piece has changes of the key (modulations) and many notes that are outside of the theoretical note distribution of the key it has to be paid special attention to the order and timing of notes.
Also tuning variations (e.g. mistuned instruments) in the recorded audio has to be detected and the pitch class distribution has to be adjusted, otherwise it would just be shifted or get wrong pitch classes.
Another point is that humans perceive absolute pitches on a nearly logarithmic scale. That means the frequency difference between two higher notes is bigger than the difference between to low notes. So the pitch class generation algorithm has to consider this, also the recording quality has to be high enough.
Every fundamental frequency also creates partials from itself and therefor it does sometimes occur that the partials are counted into the pitch class distributions. Sometimes this is wanted (when e.g. A4 and A5 are played) but when the partials are counted into the distribution then the notes are counted at least double, which is not right.
At last it is also hard to notice modulations of the key. Especially between similar keys it is

hard to determine if just a few notes are being played that are outside of the theoretical key distribution or if these belong to a different key. But the systems shown here does not consider modulations.

Theory

According to the Oxford Dictionary of Music the key is "the pitch relationships that establish a single pitch-class as a tonal center or tonic (or key note), with respect to which the remaining pitches have subordinate functions". So a key consists of two main parts: The tonic and the mode. The tonic is one of the twelve pitch-classes. An example for a pitch-class would be: $[A0, A1, ..., A7]$. This is the pitch-class which pitches are expected to occur most in the piece. But the other important part of the key is the mode. The mode, which can be major or minor for every pitch-class, defines how often the other pitches, which are not in the tonic pitch-class, are likely to occur.

History

When looking up key recognition there are two research fields that has to be considered: The first one is symbolic key detection and the second one is audio key detection. Symbolic key detection like the name suggests uses only symbolic description of music, like scores and MIDI files. So the most of the challenges that was listed earlier are not affecting the symbolic key detection. Compared to that the audio key detection analyzes audio files and does therefore have the added difficulty of analyzing audio files to just get an overview of which notes are played. But that's also why audio key detection is more usable, because for the most available music the audio files are easy to get but to get a score or MIDI file of this same song can be pretty hard.
Symbolic key detection can be seen as the predecessor of audio key detection, because it emerged earlier. So the first notable approach in symbolic key detection was already made in 1971, while in audio key detection the first approach was just made in 1991.
So while audio key detection is more applicable, easy approaches in this field also use the findings that was made in the field of symbolic key detection.

Symbolic Key Detection

The first approach in symbolic key detection was made in 1971 by Longuet-Higgins and Steedman. They used a shape matching approach on the harmonic network, which can be seen in figure 1. They worked out that the two different modes has different shapes in the harmonic network, the tonic could then be determined on the position in the shape. The different shapes can be seen in figure 2. This is an easy approach but also not a very good one, because it is easy to produce melodies so that the system detects the wrong key.
Another approach is based on theoretical key profiles (through studies on music perception). These profiles are the information how often the notes in a pitch class (that is not the tonic) are occuring in the piece, arranged in the relative distance of the pitch-class to the tonic. These profiles are also incorporated in audio key detection systems. The examples for this are shown in figure 3 and in figure 4. The first profile in figure 3 was proposed by Krumhansl and Schmuckler in 1990 while the second profile was proposed by Temperley 9 years later and he made slight changes to the profile to help distinguish between keys. Krumhansl and Schmuckler proposed to calculate the total duration of each pitch class within the complete piece and then compare this to the key profiles and thereby find the key. This also works already pretty good, even without incorporating temporal information between the notes.
There are also approaches like the one from Madsen and Widmer in 2007 which built a system that is trained with an annotated corpus and thereby learns not only a pitch-class distribu-

tion profile but also incorporates temporal information into the model. With training date the success of the system is highly dependent on the corpus, so when the corpus is trained on one kind of music it is specialized for this same kind of music.

Audio Key Detection

Audio Key detection systems can be grouped into the following four categories:
Pattern matching approaches and score transcription methods as first, template-based methods second, then geometric models and the last group are models that are based on chord progressions hidden markov models. Especially the pattern matching part benefits from the findings of symbolic key detection systems. The first systems was proposed already in 1991 and until nowadays this is a more or less active research field, which is part of music information retrieval (MIR), and therefore a yearly evaluation of approaches is done by MIREX [5].

Audio Key Detection

In this section the four categories of audio key detection systems are explained.

Pattern matching approaches and score transcription methods

One of the first models for audio key detection was proposed by Leman in 1991. This approach was pattern matching based. It predetermined templates from self organizing maps and only compared the extracted tone centers of the songs with the predetermined templates.
Another approach that is simple when symbolic key detection models already exist is to use a score transcription system and then analyze the score. Izmirli and Bilgen used in 1994 a partial score transcription system combined with an pattern matching approach.

Template-based methods

Template based models first extract a pitch-class distribution feature from the audio file and then compare this feature to pitch-class templates to know which pitch-class fits the found distribution best. In the pitch-class distribution feature the relative strength of each pitch class within the audio file is represented.
An example for this is the approach from Van de Par et al. from 2006. They used The previously introduced key profile from Krumhansl as template and they created three different distributions from the audio that they compared to the templates. They found it is a better way to create three different distributions with different weighting factors than to use just one.

Geometric models

This methods are based on the geometric models that were made to model the coherence between the keys in a spacial way.
An example for that would be the system by Chuan and Chew from 2005, which uses the spiral array model (shown in figure 5, which is just the harmonic network wrapped into a tube with assuming octave equivalence).

Methods that are based on chord progressions or hidden markov models

As chord recognition is a related field there were proposed some methods that uses the recognition of chords an incorporated the progression to recognize the key.
An example of an HMM based system is the approach from Lee and Slaney from 2007, which is interesting because it performs chord recognition and key detection simultaneously. Therefore it applies 24 separate HMM's with 24 states each. Each HMM was trained for one of the 24 possible keys and each state should represent a single type of chord.

System and findings of the thesis

In the thesis a system with multiple components was implemented to test different parts and to determine the best parts that can be used.
The system consists of feature extraction and key classification, where the feature extraction consists of frequency analysis, pitch class extraction and pitch class aggregation.
The system parts are evaluated not only using classical songs but also using a popular corpus which mainly consists of Beatles songs and MIDI generated classical songs. First the systems are evaluated on each of the sets alone and then they are evaluated using the combined set of these single sets.

Feature Extraction

The frequency analysis was done with the FFT and parameters that were changed was the sampling rate, the windows size and the windows overlap. It was found that the optimal parameters are a sampling rate of 22050Hz, which is surprisingly not the highest one possible, with a window size of roughly $1/3s$ (8192) and a big window overlap of 0.8.
For the pitch class extraction three extension were proposed and tested with which combinations the system works best. These extensions are peak detection (PD), which only consideres high peaks in the signal (by comparing to the average value), spectral flatness measure (SFM), which also tries to detect regions with peaks and regions that are flat and low frequency clarification (LFC), which addresses the problem of low frequency resolution. The evaluation here resulted in the fact that the SFM extension should not be used to get the best results. With the extension the results in each category are worse than without it. Apparently one extension to emphasize the peaks is enough.
Another part that is tested is the pitch class aggregation, which is normally only the arithmetic mean of all pitch classes of all windows. Opposed to that a periodic cleanup can be used, where in every 1/2/4 seconds window the pitch class distributions with the smallest values are not considered. The resulting pitch-class distributions are then added together and normalized. Best results are shown when using periodic cleanup with a period of four seconds.

Key Classification

The output of the previous section is then put into the key classification stage, where four different approaches are tested: Neural networks are applied as classificator, the k-nearest neighbor algorithm, support vector machines and naive bayes classifier.
First it is needed to mention that all parts in the feature extraction section were evaluated using the KNN classifier, so they are tuned for this specific case.
But nevertheless the results are pretty interesting:
On each of the single datasets the KNN classifier outperforms the other approaches, maybe due to the fact that it was used for tuning of the first stage. But on the combined dataset, the naive bayes classifier performs best, followed by the neural network and then followed by the KNN classifier. The margin between these three systems is very small, so they can be seen to perform similar. Interesting is that due to the better performance of the neural network and naive bayes classifier in the combined set it seems that this both approaches are better to generalize how to detect the key when trained with more data and data from diverse genres.

Datasets

It is also noteworthy that in the earlier days the key recognition systems were mainly tested on datasets that only incorporated classical music. Today there is at least one dataset for key recognition available, that incorporates electronic dance music, namely the GiantSteps Key Set. The yearly Music Information Retrieval Evaluation eXchange (MIREX) also incorpo-

rates key recognition and since 2015 the approaches are also evaluated on the GiantSteps dataset. Especially for DJ mixing software this dataset gives more practical conclusions than evaluating the approaches solely on classical datasets. So nowadays MIREX key recognition is evaluated on both datasets (classical MIREX 05, GiantSteps) and the best results are 83% correct detected keys for the dance music [3] and 62% for the classical music [2]. At least for the dance music the results are pretty good. On the GiantSteps website also evaluation results of available DJ software is shown, where the best product is Pioneer Rekordbox v3.2.2 with only 72% correct recognized keys [4].
A consideration is that more diverse datasets of all genres would be needed to train better systems and also evaluate the performance of the systems encompassing.

Conclusion

There has been a lot of research on automatic key recognition and it exist some pretty good approaches, but none of them is completely reliable. The systems are getting better but there is still a gap to the performance of human experts. Nowadays they give mostly good results, but not in every case and so there is still some work to do.

References

[1] Campbell, Spencer. *Automatic Key Detection of Musical Excerpts from Audio*. Music Technology Area, Department of Music Research, Schulich School of Music, McGill University, Montreal, Canada (Aug. 2010), (127 pages total).

[2] Link: MIREX 16 Key Recognition results on MIREX05. Accessed on July 5, 2017. `http://nema.lis.illinois.edu/nema_out/mirex2016/results/akd/mrx_05/summary.html`

[3] Link: MIREX 16 Key Recognition results on GiantSteps. Accessed on July 5, 2017. `http://nema.lis.illinois.edu/nema_out/mirex2016/results/akd/gsteps/summary.html`

[4] Link: GiantSteps evaluation of available algorithms and products (DJ software). Accessed on July 5, 2017. `http://www.cp.jku.at/datasets/giantsteps/`

[5] Link: The main MIREX Website. Accessed on July 5, 2017. `http://www.music-ir.org/mirex/wiki/MIREX_HOME`

Appendix

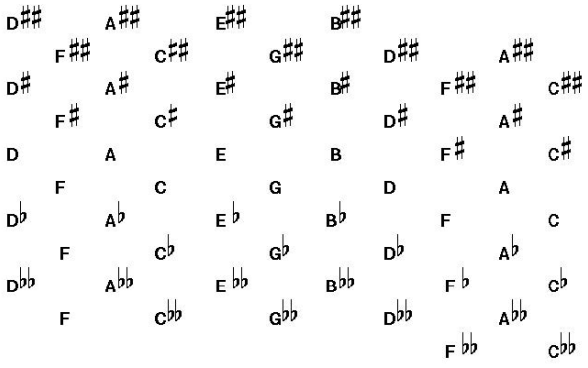

Figure 1: The Harmonic Network (Tonnetz): Closer pitch classes have stronger harmonic relations

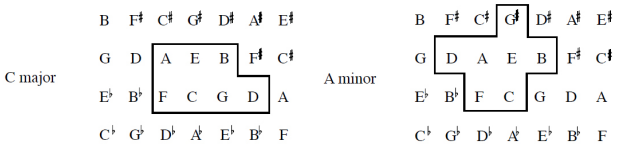

Figure 2: A visualization of the shape matching approach: Especially the different shapes for the different modes can be seen.

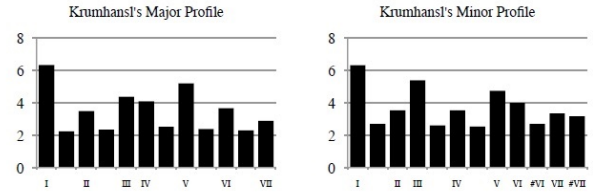

Figure 3: Krumhansl's key profiles.

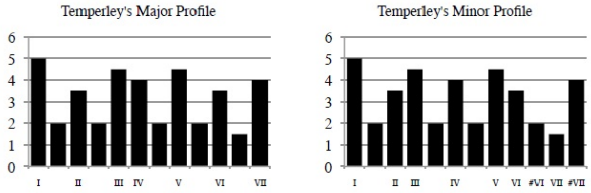

Figure 4: Temperley's key profiles.

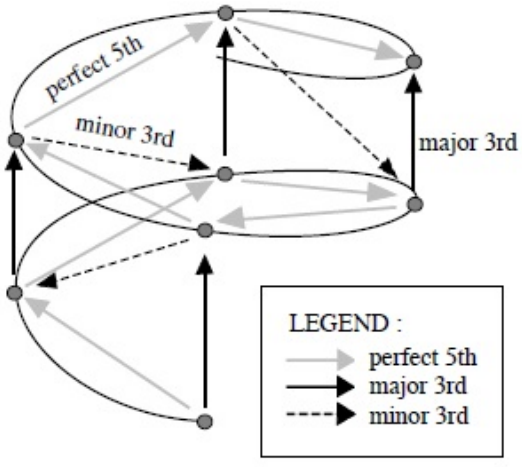

Figure 5: The spiral array model.