Pattern Recognition in Music

SAMBA/07/02

Line Eikvil
Ragnar Bang Huseby

February 2002

NR Norsk Regnesentral
ANVENDT DATAFORSKNING

NR-notat/NR Note

Tittel/Title:
Pattern Recognition in Music

Dato/Date: February
År/Year: 2002
Notat nr:
Note no: SAMBA/07/02

Forfatter/Author:
Line Eikvil and Ragnar Bang Huseby.

Sammendrag/Abstract:

This report gives a brief overview of different applications, problems and methods related to pattern recognition in music. Many of the applications of musical pattern recognition are connected to music information retrieval. This area covers fields like information retrieval, signal processing, pattern recognition, artificial intelligence, databases, computer music and music cognition. The report focuses on problems and methods related to signal processing and pattern recognition.

Automatic music transcription and content-based music retrieval are the problems that have received the most attention within this area. For music transcription the current state-of-the-art is that monophonic transcription for well-defined musical instruments has to a large degree been solved as a research problem, while transcription of polyphonic music remains a research issue for the general case. Content-based retrieval based on audio queries is somewhat dependent on the transcription, although a full transcription may not be necessary to find similarity.

Other problems like genre classification, music summarization and musical instrument recognition are also treated in the report. These are also problems that are related to music retrieval in that these techniques can be used for organizing the music databases and to present the results to users. Less research has been done in these areas.

Emneord/Keywords: Audio, music, pattern recognition, content-based retrieval

Tilgjengelighet/Availability: Open

Prosjektnr./Project no.: GB-BAMG 1033

Satsningsfelt/Research field: Pattern recognition

Antall sider/No. of pages: 37

Norsk Regnesentral / Norwegian Computing Center
Gaustadalléen 23, Postboks 114 Blindern, 0314 Oslo, Norway
Telefon 22 85 25 00, telefax 22 69 76 60

Pattern Recognition in Music

Line Eikvil and Ragnar Bang Huseby

February 28, 2002

Contents

Chapter 1

Introduction

In this report we will look at different applications, problems and methods related to pattern recognition in music. Many of the applications of musical pattern recognition are connected to music information retrieval. This area covers fields like information retrieval, signal processing, pattern recognition, artificial intelligence, databases, computer music and music cognition. This report will focus on problems and methods related to signal processing and pattern recognition.

In Chapter 2 we will present different application areas and problems where musical pattern recognition is needed. Chapter 3 briefly describes different methods from signal processing and pattern recognition which have been used to solve the different problems. Finally, Chapter 4 lists some existing systems based on musical pattern recognition.

Chapter 2

Applications

2.1 Content-based retrieval

In content-based music information retrieval (MIR) the primary task is to find exact or approximate occurrences of a musical query pattern within a music database. MIR has many applications, and in the future, one can imagine a widespread use of MIR systems in commercial music industry, music radio and TV stations, music libraries, music stores, for musicologists, audio engineers, choreographers, disc-jockeys and even for one's personal use. Some user could possibly require all musical documents with the same key another user would obtain all documents of the same tempo. Another user might need to know the number of times the violin had a solo part in a given composition.

By humming a short excerpt of a melody into a microphone, a CD-player can be requested to play a particular piece of music or MPEG files can be down-loaded from the Internet. This application is discussed more closely in section 2.1.1. A query could also be input via a keyboard. MIR techniques may also be used for solving judicial plagiarism cases [14]. In Section 2.1.2, we present some applications concerning similarity queries against a database of digital music. A summary of the methodology is given in Section 3.8.

2.1.1 Query by humming

Several systems allow the user to perform queries by humming or singing. The challenge of this task is that people do not sing accurately, especially if they are inexperienced or unaccompanied; even skilled musicians have difficulty in maintaining the correct pitch for the duration of a song. Thus, a MIR system needs to be resilient to the humming being out of tune, out of time or out of key. Also, it is not known which segment of the song that will be hummed a priori.

Examples of systems are MELDEX [51, 54], "Search By Humming" [8, 64], Tuneserver [60, 68], Melodiscov [61], Semex [40, 44], and SoundCompass [35]. These systems are briefly described in Chapter 4.

2.1.2 Query by similarity

There are systems capable of performing similarity queries against a large archive of digital music. Users are able to search for songs which sound similar to a given query song, thereby aiding the navigation and discovery of new music in such archives. For instance, while listening to a song from the database, the user can request finding similar songs. An example system was developed at the University of California at Berkeley by Welsh et al. [73] and it works on an online MP3 archive.

2.2 Automatic music transcription

For decades people have been trying to design automatic transcription systems that extract musical scores from raw audio recordings. Automatic music transcription comes in two flavours: polyphonic and monophonic. In monophonic music, there is only one instrument playing, while for polyphonic music there are usually many instruments playing at the same time.

Polyphonic music is the most common form, especially for western music, but it is much more difficult to transcribe automatically. Hence, automatic transcription has only succeeded in monophonic and very simple polyphonic cases, not in the general polyphonic case [77]. In contrast, monophonic music transcription is simpler. If there is only one instrument playing, it is a matter of finding the pitch of the instrument at all points, and of finding where the notes change.

When working on transcription systems, whether polyphonic or monophonic, most researchers start with absolute pitch detection, and work from there. Automatic absolute pitch detection can however be a difficult problem, even for a monophonic signal. Research into automatic absolute pitch detection has lead to many different methods, each with its related difficulties. The problems of pitch tracking and automatic music transcription will be treated in Chapter 3.

2.3 Genre classification

Musical genres are categorical descriptions that are used to characterize music. They are commonly used to structure the increasing amount of digital music where categorization is useful for instance for music information retrieval [70]. Genre categorization has traditionally been performed manually, and humans are remarkably good at genre classifications from just very short segments of music. Although the division of music into genres is somewhat subjective and arbitrary there are perceptual criteria related to the texture, instrumentation and rhythmic structure of music that can be used to characterize a particular genre. In Chapter 3 different approaches for music genre classification will be presented.

2.4 Music summarization

Music summarization or thumb nailing refers to the process of creating a short summary of a large audio file in such a way that the summary best captures the essential elements of the original sound file [69]. It is similar to the concept of key frames in video and can be useful for music retrieval, where you want to present a list of choices which can quickly be checked by the user. Hence, potential applications are multimedia indexing, multimedia data searching, content-based music retrieval, and online music distribution. To date music summarization has not been a focused subject, but a few methods have been suggested. These will be presented in Chapter 3.

2.5 Musical instrument recognition

Automatic musical instrument recognition is a sub problem in music indexing, retrieval and automatic transcription. It is closely related to computational auditory scene analysis, where the goal is to identify different sound sources. However, musical instrument recognition has not received as much interest as for instance speaker recognition, and the implemented musical instrument recognition systems still have limited practical usability [21]. Some methods that have been used for this task will be presented in Chapter 4.

Chapter 3

Methods

In this chapter we will take a look at some of the different methods and problems encountered in the analysis of music signals. In the first sections, the representation and the features that are basis for the further analysis are introduced. Then some of the more fundamental problems like pitch tracking and matching are treated. Finally, more application oriented problems and methods are presented, including music transcription, genre classification, musical instrument recognition etc.

3.1 Audio features

The basis of any algorithm for audio signal analysis is short-time feature vector extraction, where the audio file is broken into small segments in time and for each of these segments a feature vector is calculated. Features describing the audio signal can typically be divided into two categories, physical and perceptual features. The physical features are based on statistical or mathematical properties of the signals, while the perceptual features are based on the way humans hear sound. The physical features are often related to the perceptual features.

Pitch is an important perceptual feature, that gives information about the sound. It is closely related to the fundamental frequency, but while frequency is an absolute, numerical quantity, pitch is not. Techniques for pitch determination will be discussed in Section 3.4.

Timbre is defined as that quality of sound which allows the distinction of different instruments or voices sounding the same pitch. Most of this is due to the spectral distribution of the signal, and spectral features can be used to extract information corresponding to timbre.

Rhythm generally means that the sound contains individual events that repeat themselves in a predictable manner. To extract rhythmic information from sound, repetitive events in energy level, pitch or spectrum distributions can be identified. However, the rhythm may be more complex and change with time. Also, not only music may have rhythm, but also speech, e.g. the reading of a poem.

The physical features can in general be divided into two main groups: those derived from time domain characteristics and those based on the frequency domain. Many of these features have been identified from studies within speech processing, and different features may be suitable for different problems. In the following some of the basic features will be presented. More specific features are also treated in Section 3.4 which looks at the problem of pitch determination or pitch tracking.

Energy is one of the most straight-forward features and is a measure of how much signal there is at any one time. It is used to discover silence and to determine the dynamic range in the audio signal. It is computed by windowing the signal, squaring the samples within the window and computing the average. The distribution of energy over time has been used to distinguish between speech and music [25]. Speech tends to consist of periods of high energy followed by periods of low energy, while music tends to have a more consistent energy distribution.

Zero-crossing rate is a measure of how often the signal crosses zero per time unit. This can be used to give some information on the spectral content of the signal. A large advantage of this feature, compared to spectral features, is that it is very fast to compute and can easily be calculated in real-time.

Fundamental frequency, or F0, of a signal is the lowest frequency at which a signal repeats. F0 detectors are therefore often used to detect the periodicity, and to determine if the signal is periodic or not.

Spectral features describe the distribution of frequencies in the signal. A common spectral transform is the Fourier transform. In audio signal analysis the Short Time Fourier Transform (STFT) is much used. STFT is an attempt to fix the lack of time resolution in the classic Fourier transform, where the input data is broken into many small sequential pieces called frames or windows, and the Fourier transform is applied to each of these frames in succession. This produces a time-dependent representation, showing the changes in the harmonic spectrum as the signal progresses. To reduce frame boundary effects a windowing function is used. The Fourier transform is the most common spectral transform, and it is useful for many applications, but can be less effective in time location and accurate modelling of human frequency perception.

Cepstral coefficients are found from the Fourier transform to the log-magnitude Fourier spectrum and have been much used for speech related tasks, but they also have properties that can be helpful in music analysis [13]. The variability of the lower cepstral coefficients is primarily due to variations in the characteristics of the sound source. For speech recognition, these variations are considered as noise and are usually de-emphasized by cepstral weighting, but when analysing music, the differentiation of the generating source (strings, drums, vocals etc) can be useful.

3.2 Music representation

Music can be represented in computers in two different ways [77]. One way is based on acoustic signals, recording the audio intensity as a function of time, sampled at a certain frequency, and often compressed to save space. Another way is based on musical scores, with one entry per note, keeping track of the pitch, duration (start time and end time), strength etc. for each note. Examples of this representation include MIDI and Humdrum, with MIDI being the most popular format. Score-based representations are much more structured and easier to handle than raw audio data. On the other hand, they have limited expressive power and are not as rich as what people would like to hear in music recordings.

3.3 High level features

In this section, we discuss feature extraction for music information retrieval in the case where the sound event information is encoded. That is, the pitch, onset, duration of every note in the music source are known. We consider both monophonic music where no new note begins until the current note has finished sounding, and polyphonic music where a note may begin before a previous note finishes. Homophonic music lies somewhere between these two, here notes with different pitches may be played simultaneously but they must start and finish at the same time.

3.3.1 Monophonic feature selection

Basic approaches

Most of the current work in MIR has been done with monophonic sources. Obviously, the two most important descriptors of a note are duration and pitch. In a simple approach, pitch is extracted and duration is ignored. The opposite method consists of extracting duration and ignoring pitch.

There are several reasons for taking only one attribute (at a time) into account. The main one is to facilitate the modelling of music and the modelling of distortions in the pattern. In such a case, it is often only the rhythmic pattern of the melody that has been changed. Therefore, to retrieve pieces of music from a database, without any a priori knowledge of the style they have been performed in, it would be advantageous to use only the pitch information.

Most MIR researchers (e.g. [28]) favour relative measures of pitch and duration because a change in tempo or transposition across keys does not significantly alter the music information expressed. Relative pitch has three standard expressions: exact interval, rough contour and simple contour. *Exact interval* is the signed magnitude between two contiguous pitches. Simple contour keeps the sign and discards the magnitude. Rough contour keeps the sign and groups the magnitude into a number of bins. Relative duration has similar expressions: exact ratio, rough contour and simple contour.

Lemström et al. [44] introduced a measure combining pitch interval and note duration in a single value called the interval slope. This measure is equal to the ratio of the sizes of pitch intervals to note durations. In order to obtain invariance under different tempo, they also considered the proportions of consecutive interval slopes. However, pitch and duration are most commonly treated as independent features.

N-grams

An *n-gram* is an n-tuple of things, e.g. an n-length combination of letters. The term is frequently used in analysis of text. In this context, an *n-gram* is an n-length combination of intervals (or ratios). In this way, $n + 1$ notes are turned into a single term. A special case of an n-gram is a *unigram*. A *unigram* consists of just a single interval (or ratio).

Unigrams are sufficient for retrieval systems that use string matching to compare melodic similarity, or systems that build ordered sequences of intervals (phrases) at retrieval time. Other systems may require larger basic features. An n-gram is then constructed from an input sequence of unigrams.

There are several methods for extracting n-grams. A simple approach is to use sliding windows [58], that is, the sequence of unigrams

$$\{a_1, a_2, \dots\}$$

is converted to the sequence

$$\{(a_1, a_2, \dots, a_n), (a_2, a_3, \dots, a_{n+1}), \dots\}.$$

There is a trade-off between unigram type and n-gram size. If exact magnitude unigrams are used as input, n is kept small. If contour unigrams are used, n is larger.

Another method consists of detecting repeating patterns corresponding to key melodies [71]. Such patterns may be easily recalled by people once they hear a part of the song or the name of the song.

An alternative method consists of segmenting a melody into musically relevant passages [58]. Weights are assigned to every potential boundary location, expressed in terms of relationships among pitch intervals, duration ratios, and explicitly delimited rests. Boundary markers are then placed where local maxima occur. The sequence of notes between two consecutive markers becomes an n-gram. It is also possible to use string matching for n-gram extraction [58].

Statistical features

Descriptive statistical measures can be used in MIR [58]. Such measures could be the *relative frequencies* of various pitch unigrams or pitch n-grams. Duration measures could be used in a similar manner. The length of the source could also be a relevant feature.

In some applications, the key is an important attribute. The key can be extracted by examining a sequence of note pitches and doing a probabilistic best fit into a known key [65, 38].

3.3.2 Polyphonic feature selection

Most research on MIR has been based on monophonic music. However, since most real music is polyphonic, it is necessary to develop methodology for extraction of patterns from polyphonic sources. The source usually consists of multiple tracks and channels, each representing a separate instrument.

Monophonic reduction

By *monophonic reduction* a polyphonic source is reduced to a monophonic source. This is done by selecting at most one note at every time step. Lemström et al [39] consider unrestricted search for monophonic patterns within polyphonic sources. The problem is to find all locations in a polyphonic musical source that contain the given monophonic query pattern. To find a matching pattern, any note of each chord can be selected. Since an exhaustive evaluation of all possible melody lines that the source contains would be very slow, faster solutions are needed. They propose algorithms for searching with and without transposition invariance.

Uitdenbogerd [72] propose several approaches for pulling out an entire monophonic note sequence equal to the length of the polyphonic source.

1. Combine all channels and keep the note with the highest pitch from all simultaneous note events.
2. Keep the note with the highest pitch from each channel, then select the channel with the highest first-order predictive entropy, that is, the most complex sequence of notes.
3. Use heuristics to split each channel into parts, then choose the part with the highest entropy.
4. Keep only the channel with the highest average pitch, then keep only the notes with the highest pitch.

The underlying idea is that, although many instruments may be playing simultaneously, only some of the notes are perceived as part of the melody. In their experiments the first approach was the most successful. However, in many cases, e.g. in choral music, the highest voice does not necessarily contain the melody; it is even possible that the melody is distributed across several distinct voices.

Instead of extracting a melodic line, the source can be split into a number of monophonic sequences. Each monophonic sequence can then be searched independently, and combining the results yields a score for the piece as a whole.

Homophonic reduction

Homophonic reduction consists of selecting every note at a given time step. In this way we obtain a sequence of sets of notes instead of a sequence of single notes. Such sets are called homophonic slices. Other names like syncs and chords are also used in the literature.

It is possible to construct a transposition invariant sequence from the homophonic slices [39]. This is done by taking the difference between all possible note combinations in two contiguous homophonic slices. However, intervals formed in this way do not always reveal the true "contour" of the piece. This is caused by ornamentation, passing tones, and other extended variations. Therefore Pickens [58],suggests that each set in the sequence is extended to allow for differences of note combinations from non-contiguous homophonic slices.

Of course, duplicates of differences may occur. Instead of discarding duplicates, this information could be useful for incorporating the strength of the intervals. Also, intervals could be weightened. For instance, slices that are not located "on beat" could be down weighted. In order to emphasize harmonic context, intervals within the same slice can be included.

Statistical features

As with monophonic music it is possible to extract descriptive statistical measures for polyphonic music. Many of the measures that are applied to monophonic music can be extended to polyphonic music in a trivial way. Also there are more features possible for polyphonic music. Examples of features are the number of notes per second, the number of chords per second, the pitch of the average note, the pitch of the lowest/highest note, and so on.

3.4 Pitch tracking

The general concept of pitch is that it is the frequency that most closely matches the tone we hear. Determining the pitch is then equivalent to finding which note has been played. However, performing this conversion in a computer is a difficult task because some intricacies of human hearing are still not understood, and our perception of pitch covers an extremely wide range of frequencies.

In monophonic music the note being played has a pitch that is related to the fundamental frequency of the quasi-periodic signal that is the musical tone. In polyphonic music, there are many pitches acting at once. Pitch determination has also been important in speech recognition, since some languages such as Chinese rely on pitch as well as phonemes to convey information.

The objective of a pitch tracker is to identify and track the fundamental frequency of a waveform over time. Many algorithms exist, and some of these are inspired by image processing algorithms, since a time-varying spectrum has three dimensions. The first methods for this started to appear

30 years ago, and many different algorithms have been developed over the time. But while improvements to the common algorithms have been made, few new techniques have been identified.

The algorithms may be categorized dependent on the domain in which they are applied:

- Time domain (based on a sampled waveform)
- Frequency domain (amplitude or phase spectrum)
- Cepstral domain (second order amplitude spectrum)

3.4.1 Time domain methods

A sound that has pitch has a waveform that is made up of repeating segments or pitch periods. This is the observation on which time domain pitch trackers are based. They attempt to find the repeating structure of the waveform. In the following a few of these techniques are briefly described.

Autocorrelation: Autocorrelation is one of the oldest of the classical pitch trackers. The goal of the autocorrelation routines is to find the similarity between the signal and a shifted version of itself. The signal peaks where the impulses occur. Therefore, tracking the frequency of these peaks can give the pitch of the signal. The technique is most efficient at mid to low frequencies. Thus, it has been popular in speech recognition applications where the pitch range is limited. Depending on the frame length, autocorrelation can be computationally expensive involving many multiply-add operations. The autocorrelation can also be subject to aliasing (picking an integer multiple of the actual pitch).

Maximum Likelihood: Maximum Likelihood is a modification of autocorrelation that increases the accuracy of the pitch and decreases the chances of aliasing. The computational complexity is higher than that of auto-correlation.

Zero Crossings: This is a simple technique that consists of counting the number of times that the signal crosses the 0 level reference. The technique is inexpensive but is not very accurate, and when dealing with highly noisy signals or harmonic signals where the partials are stronger than the fundamental, the method has poor results.

Gold-Rabiner: Gold-Rabiner is one of the best known pitch tracking algorithms. It determines frequency by examining the structure of the waveform [50]. It uses six independent pitch estimators, each working on a different measurement obtained from local maxima and minima of the signal. The final pitch estimate is chosen on the basis of a voting procedure among the six estimators. When the voting procedure is unable to agree on a pitch estimate, the input is assumed to be silence, or an unvoiced sound. The algorithm was originally designed for speech applications.

AMDF: The average magnitude difference function (AMDF) is another time-domain algorithm that is very similar to autocorrelation. The AMDF pitch detector forms a function which is the complement of the autocorrelation function, in that it measures the difference between the waveform and a lagged version of itself.

Super Resolution Pitch Determination: This method uses the idea that the correlation of two adjacent segments is very high when they are spaced apart by a fundamental period or a multiple of it. The method quantifies the degree of similarity between two adjacent and non-overlapping intervals with infinite time resolution by linear interpolation.

3.4.2 Frequency Domain methods

The second group of methods operates in the frequency domain, locating sinusoidal peaks in the frequency transform of the input signal. Frequency domain methods call for the signal to be frequency transformed, then the frequency domain representation is inspected for the first harmonic, the greatest common divisor of all harmonics, or other such indications of the period. Windowing of the signal is recommended to avoid spectral smearing, and depending on the type of window, a minimum number of periods of the signal must be analysed to enable accurate location of harmonic peaks. Most successful analysis methods for general single voice music signals are based on frequency domain analysis.

3.4.3 Cepstrum Analysis

The term cepstrum is formed by reversing the first four letters of spectrum. The idea is to take the Fourier transform to the log-magnitude Fourier spectrum. Thus, if the original spectrum belongs to a harmonic signal, it is going to be periodic in the frequency representation, and taking the FFT again it will show a peak corresponding to the period in frequency, thus we can isolate the fundamental period. The output of these methods can be viewed as a sequence of frequency estimations for successive pitches in the input. The cepstrum approach in pitch tracking often takes more computation time than autocorrelation or Fourier transformation based methods. Besides, it has been reported that the method does not perform well enough for pitch tracking on signals from singing or humming.

3.5 Segmentation

There are different segmentation problems related to the analysis of digital music. In this section we do not consider the low-level segmentation problems like note segmentation, but rather more high-level segmentation problems like that of distinguishing speech from music and segmenting vocal parts within a music piece.

3.5.1 Speech/music segmentation

Automatic discrimination of speech and music is an important tool in many multimedia applications, like speech recognition from radio broadcasts, low bit-rate audio coding, and content-based audio and video retrieval. Several systems for real-time discrimination of speech and music signals have been proposed. Most of these systems are based on acoustic features that attempt to capture the temporal and spectral structures of the audio signals. These features include, among others, zero-crossings, energy, amplitude, cepstral coefficients and perceptual features like timbre and rhythm.

Scheirer and Slaney [63] evaluate 13 different features intended to measure conceptually distinct properties of speech and musical signals and combine them in a classification framework. Features based on knowledge of the speech production such as variances and time averages of spectral parameters are extracted. Characteristics from music are also used, exploiting the fact that music has a rhythm that follows all the frequency bands synchronously. Hence, a score for synchronous events in the different bands over a time interval is calculated. Different classifiers, including a Gaussian mixture model and KNN were tested, but little difference between the results are reported. For the most successful feature combinations a frame-by-frame error rate of 5.8% is reported. Averaging results over larger windows, results in an error rate of 1.4% for integrated segments.

A different approach is suggested by William and Ellis [75], who propose the use of features based on the phonetic posterior probabilities generated in a speech recognition system. These features are specifically developed to represent phonetic variety in speech, and not to characterize other types of audio. However, as they are precisely tuned to the characteristics of speech, they behave very differently for other types of signals.

Chow and Gu [12] describe a two-stage algorithm for discrimination between speech and music. Their objective is to make the segmentation method more robust to singing. In the first stage of the segmentation process they want to identify segments containing singing, as singing can be more difficult to discriminate from speech than other forms of music. Different features are tested and 4 Hz modulation energy is identified as the feature most suited to distinguish between speech and music, while features like MFCC and zero-crossing were less successful in this.

3.5.2 Vocal segmentation

In [5] an approach to segment the vocal line in popular music is presented. They see this as a first step on the way to transcribe lyrics using speech recognition. The approach assumes that the audio signal consists of music only, and that the problem is to locate the singing within the music. This problem is not directly related to that of distinguishing between music and speech, but the work is based on ideas from this.

A neural network trained to discriminate between phonetic classes of spoken English is used to generate a feature vector which is used as a basis for the segmentation. This feature vector will contain the posterior probability of each possible phonetic class for each frame. The singing, which is closer to speech than the instrumental parts, is then assumed to evoke a distinctive pattern of response in this feature vector, while the instrumental parts will not. Different types of features are derived from the basic feature vector and introduced to a hidden Markov model to perform the segmentation. A HMM framework with two states “singing” and “not singing” is used to find the final labelling of the stream. Distributions for the two states are found from manual segmentation, by fitting a single multidimensional Gaussian to the training data. Transition probabilities are also estimated from the manually segmented training set. The approach showed a successful segmentation rate of 80% on the frame level.

3.6 Matching

Matching is one of the primary tools for pattern recognition in musical data. Matching consists of comparing a query pattern with patterns in a database by using a similarity measure. Often the pattern is a pitch sequence. The pattern may also be a sequence of spectral vectors. Given the query pattern the task is to find the most similar pattern in the database. If the similarity measure is a distance measure this corresponds to finding the pattern in the database with the shortest distance to the query pattern. It is probably impossible to define a universal distance measure for music comparison and retrieval due to the diverse musical cultures, styles, etc.

3.6.1 Similarity measures

One of the simplest distance measures is the *Euclidean distance*. Given two vectors of the same dimension, the *Euclidean distance* is defined as the square root of the sum of the squares of the component wise differences. The Euclidean distance is used in [73]. In order to prevent one feature from weighting the distance more than others they normalise all features such that each feature value is between zero and one. If k is the number of similar songs to return for a given query, the similarity query is performed by a k-nearest-neighbour search in the feature space.

Another simple distance measure is the *city block distance*. It is defined by taking the sum of the absolute values of the component wise difference. The *average city block distance* can be defined by dividing the *city block distance* by the number of components.

Often the representation of a melody is based on musical scores keeping track of the pitch and duration for each note. In such cases a tune can be represented by pitch as a function of time. Ó Maidín [57] compares two tunes by computing the area between two graphs of the two functions. This method is not transposition invariant. Another drawback is that it assumes that both tunes have the same tempo.

Francu et al. [24] define a transposition invariant distance between two monophonic tunes. In order to define this measure the time and the pitch scales are quantised such that the two tunes can be regarded as two pitch sequences. Given a pitch sequence, a new pitch sequence is formed by adding a constant pitch offset. Another pitch sequence can be formed by a time shift. For each possible pitch offset and time shift of a pitch sequence we can compute the *average city block distance*, and by taking the minimum over all the pitch offsets and time shift, we obtain a transposition invariant distance.

It is possible to modify this measure in order to allow for different tempos. This can be done by rescaling the time scale of the query with various factors, then performing the minimisation above for each factor, and finally determining the minimum taken over all the factors. If the tunes consist of multiple channels, the distance can be computed for each pair of channels. Then the minimum distance taken over all pairs defines a distance measure.

Several n-gram measures have been proposed in the literature. For a given n-gram the absolute difference between number of occurrences in the two pitch sequences in question can be computed. The Ukkonen measure is obtained by taking the sum of these absolute differences over all n-grams occurring in either of the two. Another measure considered by Uitdenbogerd et al. [72] is the number of n-grams in common between the two pitch sequences. A version of this measure has also been proposed by Tseng [71].

3.6.2 Matching based on edit operations

In western music the number of various pitches and durations is quite small. Therefore a (monophonic) tune can be regarded as a discrete linear string over a finite alphabet. This motivates the use of matching algorithms originally designed for keyword matching in texts.

The basic method for measuring the distance between two string patterns p and q consists of calculating local transformations. Usually the considered local transformations are:

- insertion of a symbol in q,
- deletion of a symbol in p, and
- substitution of a symbol in p by a symbol in q.

By a composition of local transformations, p can be transformed into q. The composition is not unique since a replacement can be obtained by a composition of one insertion and one deletion. The *edit distance* between p and q is defined as the minimum number of local transformations required to transform p into q. This measure can be determined by dynamic programming. When the three kinds of transformations mentioned above are used the edit distance is called *Levenshtein distance*.

Another kind of edit distance is the *longest common sub string*. When applying this measure pieces are ranked according to the length of the longest contiguous sequence that is identical to a sequence in the query. It is also possible to use the *longest common subsequence*. This method differs from the previous in that there is no penalty for gaps of any size between the matching symbols.

The definition of an edit distance between two strings p and q can also be based on the concept of alignment. This done by splitting p and q into equally many subsequence's. Thus q can be obtained from p by transforming each subsequence in p to the corresponding subsequence in q. A cost can be assigned to each transformation step. Minimising the total cost over the set of possible alignments yields a distance measure. A different cost function yields a different distance measure. The cost could for instance depend on the magnitude of the difference between the components.

For the purpose of music information retrieval a transposition invariant edit distance is useful. One suggestion is defined by letting the cost be zero if the difference between two consecutive components is the same for the two sequences, and one otherwise.

Some distance measures are defined by partial alignment. In this case subsets of sequences are matched. The distance is then a sum of all matching errors plus a penalty term for the number of non-matching points weighted by β. By minimising the distance measure over the possible pairs of matching subsets we obtain another distance measure. Such a distance measure was used by Yang [77] in order to compare sequences of spectral vectors.

In that case the distance measure alone does not yield a robust method for music comparison. Yang [77] examines the graph of the optimal matching pair of subsets, fits a straight line through the points and removes outliers. The number of remaining matching points is taken as an indicator of how well two tunes match.

3.6.3 Hidden Markov Models

Spotting a query melody occurring in raw audio is similar to keyword word spotting performed in speech processing. The successful use of Hidden Markov Models in word spotting applications suggests that such tools might make a successful transition to melody recognition.

A hidden Markov model (HMM) consists of an underlying stochastic process that is not observable (hidden), but can be observed through another stochastic process that produces a sequence of observation vectors. The underlying process is a Markov chain, which means that only the current state affects the choice of the next state. HMM tools is an example of methodology that maintains a number of guesses about the content of a recording and qualifies each guess with a likelihood of correctness.

Durey et al. [20] use HMMs to represent each note for which data is available. These note level HMMs are then concatenated together to form an HMM representing possible melodies. The observation vectors are either computed from raw data using Fast Fourier Transform or single pitch estimates obtained by using a autocorrelation method. Based on the HMM an algorithm can produce a ranked list of most likely occurrences of the melody in a database of songs.

3.7 Automatic music transcription

The aim of automatic music transcription is to analyse an audio signal to identify the notes that are being played, and to produce a written transcript of the music. In order to define a note-event, three parameters are essential: pitch, onset and duration. Hence, an important part of music transcription is pitch tracking. (An overview of pitch tracking methods is given in Section 3.4). For music transcription pitch tracking is usually based on a Fourier-type analysis, although time-domain pitch detection methods are also used.

In the late 70's a number of researchers were working on the music transcription problem, and since then several methods for both monophonic and polyphonic transcription have been developed. Monophonic transcription is not trivial, but has to a large degree been solved as a research problem, while polyphonic transcription is still a research issue for the general case. In the following some of the attempts will be briefly described.

3.7.1 Monophonic

Piszczalski and Galler [59] (1986) developed a system for transcription of monophonic music all the way to common music notation. Their system used DFT to convert the acoustic signal to the frequency domain, after which the fundamental frequencies were detected. A pattern matching approach was then used to find the start and end of notes, and a score was generated. The system was limited to instruments with a strong fundamental, and had some problems with determining the correct length of notes and pauses, but otherwise it performed reasonably well. Piszczalski and Galler restricted input to recorders and flutes playing at consistent tempo. These instruments are relatively easy to track because they have a strong fundamental frequency and weak harmonics.

Askenfelt [2] (1979) describes the use of a real-time hardware pitch tracker to notate folk songs from tape recordings. People listened to output synthesised from the pitch track and used a music editor to correct errors. However, it is not clear how successful the system was: Askenfelt reported that the weakest points in the transcription process was the pitch detection and the assignment of note values.

Kuhn [36] (1990) described a system that transcribes singing by displaying the evolution of pitch as a thick horizontal line on a musical staff to show users the notes they are producing. No attempt was made to identify the boundary between one note and the next. The only way to create a musical score was for users to tap the computers keyboard at the beginning of each note.

McNab [50] (1996) presents a scheme for transcribing melodies from acoustic input, typically sung by the user. It tracks the pitch of the input using the Gold-Rabiner algorithm (see Section 3.4). A post-processing step where the output are filtered to remove different types of errors is then performed. After the filtering, note segmentation is performed. Two different methods are used. The first is a simple amplitude based segmentation, which requires the user to separate each note by singing *da* or *ta*. The consonant will then cause a drop in amplitude at each note boundary. The alternative method performs segmentation based on pitch.

Monophonic transcription is not trivial, but has to a large degree been solved for well-defined instruments with strong fundamentals, however transcription of a singing voice is more difficult. In the latter case, accuracy at the beginnings and ends of notes and transitions between frequencies, can be a problem. Determining the boundaries between notes is not easy, particularly not for vocal input, although users can help by singing *da* or *ta*. Furthermore, most people are not good singers, which introduces another source of variability that must be addressed for a transcription device to be useful.

3.7.2 Polyphonic

The problem of polyphonic pitch detection is not solved for the general case and many researchers are working on this. One approach to the problem is to separate the auditory stream. If a reliable method for separation existed, then one could simply separate the polyphonic music into monophonic lines and use monophonic techniques to do the transcription.

Moorer at Stanford University [55] (1977) developed an early polyphonic system, where he managed to transcribe guitar and violin duets into common music notation. His system worked directly on the time domain data, using a series of complex filtering functions to determine the partials. The partials were then grouped together and a score was printed. This system generated very accurate scores, but could not resolve notes sharing common partials, and reported problems in finding the beginnings and ends of notes. Further improvement of this system was made by Maher by relaxing the interval constraints.

Kashino et al. [33] (1995) were the first to use human auditory separation rules. They applied psychoacoustic processing principles in the framework of a Bayesian probability network, where bottom-up signal analysis was integrated with temporal and musical predictions. An extended version of their system recognised most of the notes in a three voice acoustic performance involving violin, flute and piano.

Martin [47] (1996) proposes a blackboard architecture for transcribing Bach's four voice piano chorales. The name "blackboard systems" stems from the metaphor of experts working together around a blackboard to solve a problem. Martin's blackboard architecture combines top-down and bottom-up processing with a representation that is natural for the musical domain, exploiting information about piano music. Knowledge about the auditory physiology, physical sound production and musical practice are also integrated in the system. The system is somewhat limited, fails to detect octaves, and assumes that all notes in a chord are struck simultaneously and that the sounded notes do not modulate in pitch.

Klapuri et al. [34] (2001) use an approach where processing takes place in two parallel lines; one for the rhythmic analysis and one for the harmony and melody. The system does not utilize musical knowledge, but simply looks at the input signal and finds the musical note for each segment at a time. It detects beginnings of discrete events in the acoustic signal from the logarithmic amplitude envelopes and distinct frequency bands, and combines the result across channels. Onsets are first detected one by one, then the musical meter is estimated in several steps. Multipitch estimation is performed, using an iterative approach. The system is tested on database of CD-recordings and synthesized MIDI-songs of different types. The performance is comparable to that of trained musicians in chord identification tasks, but it drops radically for real-world musical recordings.

One of the fundamental difficulties for automatic transcription systems, is the problem of detecting octaves [48]. The theory of simple Fourier series dictates that if two periodic signals are related by an octave interval, the note of the relative higher pitch will share all of its partials with the note of lower pitch. Without making strong assumptions about the strengths of the various partials, it will not be possible to detect the higher-pitched note. Hence, it is necessary to use musical knowledge to resolve the potential ambiguities. It is therefore also a need to formalize musical knowledge and statistics for musical material.

3.8 Music retrieval

In music information retrieval (MIR) the task is to find occurrences of a musical query pattern in a database. In order to apply MIR techniques the music must be represented in a digital form. One popular representation is the Musical Instrument Digital Interface (MIDI), and there exist software programs that converts singing or playing (monophonic music) to MIDI, e.g. Autoscore [74]. From the MIDI representation it is possible to extract a sequence of symbols taken from an alphabet corresponding to attributes of the music, such as the duration and the pitch of a note. The latter is more convenient for MIR purposes. The problem of converting raw audio to symbolic representation has received much attention, see Section 3.4.

Having obtained a representation of a melody it is necessary to extract features from the representation. Such features should contain the information most relevant to the problem in question. If the representation is symbolic, the methods of Section 3.3 can be applied, while techniques for extracting features from raw audio is described in Section 3.1.

When the feature vector of the query melody has been generated, the feature vector is compared with corresponding feature vectors in the database. This comparison is usually done my matching techniques, see Section 3.6. The goal is to search in the database for the feature vector that is closest or close to the feature vector representing the query.

Using computers for musical retrieval was proposed as early as in 1967 [45]. The ideas were at a very general level, such as: transcription by hand had to be avoided; and there had to be an effective input language for the music and economic means for printing the music. Automatic music information retrieval systems have been practically unavailable, mainly due to the lack of standardized means of music input. In 1995 the first real experiments of MIR were presented by Ghias et al. [28]. One of the first working retrieval system was MELDEX [51, 54]. For more information about this and other systems for music retrieval, see Section 4.2.

3.9 Music summarization

Automatic music summarization or thumb nailing aims at extracting a concise gist from music, so that interaction with large multimedia databases can be made simpler and more efficient [13]. This is a problem which has only been addressed by a few researchers, as this has been less important for music transcription and retrieval.

In one of the few papers treating this problem, Chu and Logan [13] present two different approaches for obtaining music summaries. One approach is based on clustering and the other is based on hidden Markov models. For both approaches, cepstral coefficients are used as audio features.

The clustering approach solves the problem in two steps. First, a given song is divided into fixed length segments. These segments are grouped into clusters based on their cross-entropy measures. In the second step the clusters are sorted by their frequencies of occurrence. The longest example of the most frequent episode is then chosen as the key-phrase or summary. The limitations of this clustering approach are reported to be the fixed resolution resulting from the initial set of clusters, which can result in unnatural segmentation.

In the HMM approach, the Markov model is used to reflect the structure of the data, and to integrate the segmentation process and the pattern discovery. One HMM is trained for each song, where they want each state to correspond to a group of similar segments in the song. In an evaluation of the two methods, they are reported to perform better than chance, with the clustering approach achieving the best results.

In the MARSYAS system for musical analysis, a third method for music summarization is included [69]. This is a segmentation-based method, which identifies short segments around segmentation boundaries and concatenate these to form a summary.

3.10 Genre classification

Music genre classification is related to general audio categorization, which is a field that has received quite much attention. We will here only review what has been done related to musical genre classification. Previous work in music classification has been done by looking at musical symbols, and not the audio signals. The computing musicology community has presented several tools for instance for retrieval based on symbolic representations of scores, and has tried to extract content by first transcribing the music into symbolic notation and then using music theory to characterize it. In the following we will however concentrate on methods based on the audio signal.

Current systems try to analyse structure and content directly from features calculated from the audio signal. Matityaho and Furst [49] (1995) proposed to use a multi-layer neural network classifier to separate classical and pop music. The audio features used are the average amplitude of

Fourier transform coefficients within different sub bands. The sub band is defined by dividing the cochlea into several equal sized bands and choosing corresponding resonance frequencies along the cochlea at these positions. The neural network considers a window of successive frames simultaneously, and the final decision is made after the output of the neural network is integrated over a short period.

Lambrou et al. [37] (1998) attempted to classify music into rock, piano, and jazz. They collected eight first-order and second-order statistical features in the temporal domain as well as three different transform domains: adaptive splitting wavelet transform, logarithmic splitting wavelet transform, and uniform splitting wavelet transform. For features from each domain, four different classifiers were examined. These were a minimum distance classifier, a K-nearest neighbour distance classifier, a least squares minimum distance classifier (LSMDC), and a quadrature classifier. An accuracy of 91.67% was achieved under several combinations of feature set and classifiers. The LSMDC was the best classifier for most feature types.

In a paper from 2000 Tzanetaikis et al. [70] present a method for genre classification where musical surface features derived from the time domain signal, the Fourier transform and rhythm features derived from the wavelet transform are used. In addition Mel-Frequency cepstral coefficients are used for speech and classical music. A statistical classifier is applied used for the genre classification.

In a prototype system from 2001 called MUGEC [17] feature extraction for genre classification is performed in the visual domain. This is obtained by using the spectrograms from the Fourier transform and the Mel-Frequency Cepstral Coefficients to go from the audio to the visual domain. The result is two images; one containing the spectrogram, and one containing the MFC coefficients. From these images an approach using recursive filters are used to extract the final features. Different classification techniques were tested, including KNN, Gaussian and SVM, to classify music into the categories rock, classical and jazz. Both this approach, and the one presented in [70] report on problems with the rock and jazz genres, and reasonable success with classical music.

3.11 Musical instrument recognition

Musical instrument recognition can be seen as a subtask of the more general sound source recognition problem. One of the most studied problems of this field is speaker recognition, but also problems like recognition of vehicles from sound have been much studied for instance for military purposes. Musical instrument recognition has received less attention, but also here various attempts have been made to construct automatic recognition systems. Researchers have used different approaches and scopes, each giving different performances. Most systems have operated on isolated notes, often taken from the same, single source, and having notes over a very small pitch range. The most recent systems have operated on solo music taken from commercial recordings. Attempts on polyphonic recognition has also been performed, although only for a limited number of instruments.

The first attempts in musical instrument recognition operated with a very limited number of instruments and note ranges. De Poli and Prandoni [16] used mel-frequency cepstrum coefficients calculated from isolated tones as inputs to a Kohonen self-organizing map, in order to construct timbre spaces. Kaminsky and Materka used features derived from an rms-energy envelope and used a neural network or a k-nearest neighbour classifier to classify guitar, piano, marimba and accordion tones over a one-octave band [32].

Brown [9] and Martin [48] have built classifiers that can handle data with samples played by several different instruments of a particular instrument class, recorded in noisy environments. In the approach described by Brown pattern recognition with a cluster-based Gaussian mixture model was used to identify a statistically significant number of short sounds from commercial recordings representing four musical classes; oboe, flute, saxophone, and clarinet (all woodwinds). Features explored included cepstral coefficients, constant Q coefficients, spectral centroid, autocorrelation coefficients, and moments of the time wave. The most successful features for classification gave correct results in the range of 79%–84%.

Martin has used a statistical pattern recognition technique to classify musical instrument tones within a taxonomic hierarchy. Perceptually salient acoustic features related to the physical properties of source excitation and resonance structure, were measured from the output of an auditory model (the log-lag corellogram) for isolated tones over the full pitch ranges of 15 orchestral instruments, including examples of string, woodwind and brass. Instrument families were identified with 90 % accuracy and individual instruments with a success rate of 70 %.

Eronen and Klapuri [21] present a system for pitch independent musical instrument recognition, where they use a wide set of features covering both spectral and temporal properties of sounds. For classification a hierarchical model similar to that of Martin was used. The methods were tested on strings, woodwinds and brass, yielding correct identification of instrument family with 94 % accuracy, and 80 % accuracy for individual instruments.

The current state-of-the-art in artificial sound source recognition is still very limited in its practical applicability [21]. Under laboratory conditions, the systems are able to successfully recognize a wider set of sound sources. However, if the conditions become more realistic, i.e. the material is noisy, recorded in different locations with different setups, or there are interfering sounds, the systems can only handle a small number of sound sources. The main challenge for the future is to build systems that can recognize wider sets of sound sources with increased generality and in realistic conditions [48]. Only in limited contexts, such as discriminating between four woodwind instruments, have computer systems performed comparable to human subjects.

Chapter 4

Systems

There exists several systems based on pattern recognition in music. The commercial interest has primarily been within automatic music transcription and music information retrieval. In this chapter some of these systems are presented.

4.1 Automatic Music Transcription

Quite a lot of systems exist for transcribing music. Most of these will only transcribe monophonic music, but there are also some that make an attempt at transcribing polyphonic music. However, no software currently exists that can separate instruments, and true polyphonic transcription can not be performed. The systems claiming to perform polyphonic transcription, will therefore do so only in a restricted domain. In the following three such systems are briefly described.

AmazingMIDI [3]

AmazingMIDI by Tetsuya Araki is a system that recognizes single-instrument polyphonic music, converting WAV files into MIDI files. It is assumed that every sound in the file is played with the same tone colour. As a result, even if the music contains several different instruments, all detected notes are written down as single-instrument music. The quality of the music recognition depends on the tone colour of the music. The system is said to be suitable for analysing the precise-frequency attenuating sounds like piano or guitar, while e.g. the drum sound becomes only noise.

AKoff Music Composer [1]

AKoff Music Composer by AKoff Sound Labs is a software designed for recognition of polyphonic music from audio source and its conversion to MIDI score. Recognition is performed from pre-recorded WAVE files or directly from audio input in real-time, by tracking note dynamics and pitch bends, using different harmonic models to improve recognition of appropriate instruments.

The system should recognize polyphonic music with one instrument or voice, but will not handle many instruments playing at the same time (especially not with drums). Note dynamics and frequencies are determined and translated into MIDI events, while the types of sounding instruments can not be recognized. Moreover, human voice and instruments have various timbres and complicated harmonic components, therefore recognition accuracy depends on concrete instrument or singing style. Also the recognition is influenced by the quality of the recordings such as background noises and recording level.

Intelliscore [31]

Intelliscore by Innovative Music Systems, Inc. is available both in a monophonic and polyphonic edition and will convert live or recorded polyphonic music into MIDI data. IntelliScore can recognize music containing several different instruments, it can however not differentiate one instrument from another. As a result, it writes all detected notes to the same MIDI track. The system will only be able to recognize music from instruments that have a strong pitch, and cannot recognize e.g. drums and percussion.

IntelliScore employs three different recognition algorithms based on psycho-acoustic physics, using 95 instrument filters. Some instruments and forms of music are recognized better than others. Recognition is best on audio files that are recorded at a good volume, are not too fast, and contain only a few instruments and minimal drums and percussion.

4.2 Music Information Retrieval Systems

This section presents some systems for content-based retrieval of music. This field is quite new and only quite recently has digital music become widely available e.g. on the Internet. Hence, this is not yet a mature field with many commercial systems, and most of the systems are academic. In the following we will describe briefly some of the more well-known systems.

MARSYAS [46]

MARSYAS (Princeton University) is a software framework for rapid protoyping of computer audio research. The motivation has been research in content-based audio retrieval. A significant number of Audio Information Retrieval related tools have been integrated in the framework. The framework is designed to be extensible and new features, classifiers and analysis techniques can be added to the system.

MELDEX [51, 54]

MELDEX is the New Zealand Digital Library's MELody inDEX, which retrieves music on the basis of a few notes that are sung, hummed, or otherwise entered. This is one of the very first systems performing content-based retrieval of music.

Rodger McNab et al. developed this system to search a database of folk songs from a sung query. The audio is transcribed into a melodic contour using pitch-tracking techniques and this is used to search the database. Pitch tracking and note segmentation are performed adapting to the user's gradually changing tuning in an adaptive mode. The melodic contour uses a notation which specifies whether the pitch of a note is above (U), below (D) or the same (S) as the previous note. Using approximate string matching the query melody is matched to the best result from the melodies in the database. The matching is based on a variant of dynamic programming.

A large database of 9,600 folk tunes was used to test the system and, using exact matching of rhythm and pitch, tunes were identified with only a very few mismatches. However, the system only matched the beginning of tunes. The task of searching was further simplified due to the folk songs being single track and monophonic.

MuscleFish [56, 76]

Muscle Fish has developed a variety of algorithms for analysing the content of an audio signal and producing metadata describing the signal. With this data, it is possible to query, search and classify the audio signal or collections of audio signals. The focus in this system is at the "sound" level - acoustic and perceptual properties. Musclefish works directly with the waveform and groups sounds into different classes. This system takes the acoustical features of different sounds (such as pitch and loudness) and represents these as N-vectors. By analysing these sounds it is possible to classify audio samples, search for similar sounds, search for transitions in long pieces or convert monophonic melodies to MIDI.

Search By Humming [8, 64]

This tool allows for a database of MIDI files to be searched using a pitch contour. The system uses musical pitch contours for retrieval, and the query contour can be typed in or another MIDI file can be used to find similar files. The tool would typically form the 'back-end' of a search by humming system, where the user hums a query which is converted into a contour by a pitch tracker.

A contour describes a series of relative pitch transitions, an abstraction of a sequence of notes. A note in a piece of music is classified in one of three ways: it is either a repetition of the previous note (R); higher than previous note (U); or lower than the previous note (D). Thus, the piece can be converted into a string with a three letter alphabet (U, D, R). With respect to a search by humming system, the use of contours eliminates input errors due to the user singing out of key, out of time or out of tune. As long as the pitch direction is correct then the contour should be found. The drawback is that all rhythmic information is lost; if this could be used in conjunction with the pitch contour then the number of incorrect matches would be decreased.

Tuneserver [60, 68]

Tuneserver is developed by Lutz Prechelt and Rainer Typke at the University of Karlsruhe and is a software system which recognizes a musical tune whistled by the user, finds it in a database, and returns its name, composer, and other information. It is based upon the work of Denys Parsons, a collection containing a directory of tunes that describes the opening bars of about 15,000 different melodies (classical tunes, musical theme, popular tunes and national anthems) in terms of movements up or down the scale. A Java applet is used to pitch track a whistled query. Tune recognition is then based on the encoding used by Parsons, that focuses on the direction of the melody, ignoring the size of intervals as well as rhythm.

HumDrum [30]

Humdrum is a set of tools used in the research of music, developed by David Huron at Ohio State University. It is a general-purpose software system intended to assist music researchers, with quite broad capabilities. It consists of two parts: (1) a representation syntax that is similar to tables in a spread-sheet, and (2) a set of utilities or tools that manipulate Humdrum data in various ways. The tools carry out operations such as displaying, performing, searching, counting, editing, transforming, extracting, linking, classifying, labelling and comparing. Users can also write their own programs using the Humdrum tools, or they can add new tools that augment the functioning of Humdrum.

Themefinder [67]

Themefinder has been developed through a joint project of the Center for Computer Assisted Research in the Humanities (CCARH) at Stanford University and the Cognitive and Systematic Musicology Laboratory at the Ohio State University. It is a Web-based melodic search tool with a database consisting of 2000 handpicked monophonic incipits from classical and folksong repertoires. It is implemented as a web interface to the Humdrum toolkit described above. A full complement of meta-data is also used, that includes the composer, genre, key and meter. The monophonic representations may be searched by using one of several music representations: exact pitch, pitch class, musical interval, semitone interval, scale degree, gross contour or refined contour. our. In addition, the search may be limited by entering the name of the composer and a time signature. Only exact matching is available.

Melodiscov [61]

The Melodiscov system is developed by Rolland et al. Melodiscov stands for Melody Discovery and is a melodic search system that takes an audio query. It is being developed as a WYHIWYG system (What You Hum Is What You Get). They claim that the audio query is transcribed with over 90% accuracy even when singing lyrics: many of the transcription components of similar work do not support singing of words. The search component uses several levels of representation (called descriptions) of the music that are used to describe the properties that are derived from the raw sequence of notes.

There are three levels of description: individual, local and global. Information at the individual level concerns a single note (e.g. specifying the duration and pitch). Local descriptions contain information on a segment, or musical phrase (e.g. that the passage consists of notes that are ascending in pitch). General information about the entire piece, such as the key or the average pitch, is given at the global level. Each description may be assigned a weight for similarity matching, so that differences in important areas of the music effect the matching process more. These weights can be specified by the user, so allowing different viewpoints on what makes one piece of music similar to another.

Semex [40, 44]

SEMEX is an MIR prototype developed at the University of Helsinki, Department of Computer Science. It allows, e.g., retrieving based on approximate matching, query-by-humming (the pitch tracking module it uses was developed at the Helsinki University of Technology, in the Laboratory of Acoustics and Audio Signal Processing), different reductions of the alphabet to enable feasible error tolerance, and queries targeted at polyphonic databases. It does not currently support the consideration of context in music, but the concept has been modelled. The queries are implemented by using the fast bit-parallel implementation of the dynamic programming approach.

SoundCompass [35]

SoundCompass is a retrieval system developed at the NTT Laboratories in Japan. It accepts hummed tunes as queries, using similarity retrieval as a hummed tune may contain errors. The result of a query is a list of song names ranked according to the closeness of the match, where the goal is to make the correct song appear first on the list.

The processing of the musical information is based on beats rather than notes to make it robust against queries generated from erroneous input. Acoustic information is transcribed and converted into relative intervals and is used for making feature vectors. The database currently holds over 10,000 songs, and the retrieval time is at most one second, which is achieved through the use of indices for retrieval. The SoundCompass system has been in practical use as a part of a song-selection system for Karaoke in Tokyo since the end of 2000.

Chapter 5

Summary and conclusions

In this report we have given a brief overview of different applications, problems and methods related to pattern recognition in music. Most of the applications in this area are connected to music information retrieval. This is a field covering very different technologies, ranging from signal processing and information retrieval to computer music and music cognition. We have focused on the problems related to signal processing and pattern recognition.

Music can be represented in computers in two different ways based on acoustic signals, recording the audio intensity as a function of time, or based on musical scores, with one entry per note, keeping track of pitch, duration (start time and end time), strength etc. for each note. Features describing the audio signal can typically be divided into two categories, physical and perceptual features. The physical features are based on statistical or mathematical properties of the signals, while the perceptual features are based on the way humans hear sound. The physical features are often related to the perceptual features.

Automatic transcription and content-based music retrieval are the problems that have received the most attention. The aim of automatic music transcription is to analyse an audio signal to identify the notes that are being played, and to produce a written transcript of the music. In content-based music information retrieval, the primary task is to find exact or approximate occurrences of a musical query pattern within a music database.

Automatic music transcription comes in two flavours: polyphonic and monophonic. In monophonic music, there is only one instrument playing, while for polyphonic music there are usually many instruments playing at the same time. Monophonic transcription is not trivial, but has to a large degree been solved for well-defined instruments with strong fundamentals, however transcription of a singing voice is more difficult. In the latter case determining the boundaries between notes is particularly difficult. Furthermore, the fact that most people are not good singers, introduces another source of variability. True polyphonic transcription is still a research issue for the general case.

In music information retrieval, most of the current work has been done with monophonic sources.

Recently, however, some work has also been done with polyphonic music although many problems are far from being solved. There is a great deal to be done on melody extraction, query presentation and matching techniques.

There exists systems both for music transcription and music information retrieval. For transcription several commercial systems have been developed. Although some of these systems claim to be polyphonic, none of them are able to handle true polyphonic music and will therefore do so only in a restricted domain. In the field of content-based music retrieval, the existing systems are mainly academic.

Problems like genre classification, music summarization and musical instrument recognition have also been briefly reviewed in this report. These are also problems related to music retrieval in that these techniques can be useful for organizing the music databases and to present results to the user. These areas have however received much less attention than that of music transcription and music information retrieval.

In general music analysis is still a research area where very many problems remain to be solved, although some problems like monophonic transcription has been solved to a certain degree. Research in many different areas are probably necessary to get further in the music understanding. Klapuri [34] sees a need to formalize musical knowledge and statistics for musical material, to be able to utilize more information in the analysis. While Gerhard [25] suggests that research from areas like linguistics, psychoacoustics and image processing might contribute to the field.

Bibliography

[1] AKoff Music Composer, Wav-to-Midi Converter, Akoff Sound Labs, http://www.akoff.com/

[2] Askenfelt A. *Automatic notation of played music: the VISA project*, Fontes Artis Musicae, Vol. XXVI, 1979/2, pp. 109-118.

[3] AmazingMIDI, Polyphonic Music Transcriber, Araki Software, http://www.pluto.dti.ne.jp/ araki/amazingmidi/

[4] N. Bell, *A guitar to musical instrument digital interface converter using signal processing and pattern recognition techniques*, Master's thesis. University of East Anglia, School of Information Systems, 1997.

[5] A.L Berenzweig, D.P.W. Ellis, *Locating Singing voice segments within music signals* Proc. IEEE Workshop on Apps. of Sig. Proc. to Acous. and Audio, Mohonk NY, October 2001.

[6] S.G. Blackburn. *Search by Humming*, University of Southampton, 1997. PhD. Thesis, September 2000.

[7] S.G. Blackburn. *Content Based Retrieval and Navigation of Music*, Master's thesis, University of Southampton, 1999.

[8] D.C. De Roure and S.G. Blackburn. *Content Based Retrieval and Navigation of Music Using Melodic Pitch Contours*, Multimedia Systems, 8 p.190-200. 2000.

[9] Brown, J.C. (1997). *Computer identification of musical instruments using pattern recognition* 1997 Conference of the Society for Music Perception and Cognition, Cambridge, MA.

[10] J.C.C. Chen and A.L.P. Chen, *Query by rhythm - an appoach for song retrieval in music databases*, In Proceedings of the Eighth International Workshop on Research Issues in Data Engineering (RIDE'98).

[11] T-C. Chou, A.L.P. Chen, and C-C. Liu, *Music databases: Indexing and implementation*, In Proceedings of the IEEE International Workshop on Multimedia Data Base Management Systems, 1996.

[12] W. Chou and L. Gu, *Robust Singing Detection in Speech/Music Discriminator Design*, International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001), Salt Lake City, Utah, USA, May 2001, pp.865-868.

[13] Stephen Chu and Beth Logan, *Music Summary Using Key Phrases.*, In Proc.Int.Conf on Audio, Speech and Signal Processing, ICASSP, 2000.

[14] C. Cronin, *Concepts of melodic similarity in music-copyright infringement suits*, Computing in Musicology, 11, 185-209, 1998.

[15] R. Dannenberg, J. Foote, G. Tzanetakis and C. Weare, *Panel: New directions in Music Information Retrieval* In. Proc. Int. Computer Music Conf. (ICMC), Habana, Cuba, 2001

[16] G. DePoli, P. Prandoni, *Sonological Models for Timbre Characterization*, Journal of New Music Research, Vol.26, n. 2, 1997.

[17] H. Deshpande, U. Nam, R. Singh. *MUGEC: Automatic Music Genre Classification* Technical project report (CS 329), Stanford University, June 2001.

[18] S. Doraisamy and S. M Rüger, *An Approach Towards A Polyphonic Music Retrieval System*, Proc of the 2nd Annual International Symposium on Music Information Retrieval, ISMIR 2001 (Indiana University, Bloomington, USA, 15-17 Oct 2001).

[19] M.J. Dovey, *A technique for "regular expression" style searching in polyphonic music*, Proc of the 2nd Annual International Symposium on Music Information Retrieval, ISMIR 2001 (Indiana University, Bloomington, USA, 15-17 Oct 2001).

[20] A.S. Durey and M.A. Clements, *Melody Spotting Using Hidden Markov Models*, Proceedings, International Symposium on Music Information Retrieval, October 15-17, 2001, Bloomington, IN.

[21] A. Eronen and A. Klapuri, *Musical instrument recognition using cepstral coefficients and temporal features*. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2000, pp. 753-756.

[22] J. T, Foote, *Content-Based Retrieval of Music and Audio.*, In Multimedia Storage and Archiving Systems II, Proceedings of SPIE, pp 138–147.

[23] J. T, Foote, *An Overview of Audio Information Retrieval.*, ACM Multimedia Systems, Vol. 7, No. 1, pp 2–11, 1999.

[24] C. Francu and C.G. Nevill-Manning, *Distance metrics and indexing strategies for a digital library of popular music*, IEEE International Conference on Multimedia and Expo (II), 2000

[25] D. Gerhard, *Computer Music Analysis*, Simon Fraser University School of Computing Science Technical Report CMPT TR 97-13. 1997.

[26] D. Gerhard, *Automatic Interval Naming Using Relative Pitch* Bridges 98. Winfield, KS. 37-47.

[27] D. Gerhard, *Audio Signal Classification: An Overview* Canadian Artificial Intelligence. Winter 2000. 4-6.

[28] A. Ghias, J. Logan, D. Chamberlin, and B.C. Smith, *Query by humming - musical information retrieval in an audio database*, ACM Multimedia 1995.

[29] M. Hawley, *“The personal orchestra”*, Computing Systems 2(2), pp 289–329.

[30] D. Huron, Music Research Using Humdrum. A User’s Guide. http://www.music-cog.ohio-state.edu/Humdrum/guide01.html

[31] Intelliscore, Innovative Music Systems, Inc, http://www.intelliscore.net/

[32] I. Kaminskyj and A. Materka, *Automatic Source Identification of Monophonic Musical Instrument Sounds*, IEEE International Conference on Neural Networks ICNN95, Perth, Western Australia, Nov/Dec 1995, Vol. 1, pp. 189-194.

[33] K. Kashino, K. Nakadai, T. Kinoshita, H.Tanaka. *Organization of Hierarchical Perceptual Sounds: Music Scene Analysis with Autonomous Processing Modules and a Quantitative Information Integration Mechanism.* Proceedings of the 14th Int. Joint Conf. on Artificial Intelligence (IJCAI-95), Vol.1, pp.158-164 (Aug. 1995).

[34] . A. Klapuri, A. Eronen, J. Seppanen, T Virtanen. *Automatic transcription of musical recordings,* Consistent and Reliable Acoustic Cues Workshop, CRAC-01, Aalborg, Denmark, September 2001.

[35] N. Kosugi, Y. Nishihara, T. Sakata, M. Yamamuro, and K. Kushima, *A Practical Query-By-Humming System for a Large Music Database*, In Proc. ACM Multimedia 2000.

[36] W. B. Kuhn, *A real-time pitch recognition algorithm for music applications*, Comput. Music J., pp. 60–71, Fall 1990.

[37] Lambrou, T., P. Kudumakis, R. Speller, M. Sandler, and A. Linney. (1998) *Classification of audio signals using statistical features on time and wavelet transform domains*, In International Conference on Acoustics, Speech and Signal Processing (ICASSP-98), vol. 6, (Seattle WA), pp. 3621-3624.

[38] M. Leman, *Tone context by pattern integration over time*. In D. Baggi, editor, *Readings in Computer-Generated Music*, Los Alamitos: IEEE Computer Society Press, 1992.

[39] K. Lemström and J. Tarhio, *Searching Monophonic Patterns within Polyphonic Sources*, Proc. Content-Based Multimedia Information Access (RIAO’2000), pp. 1261-1279 (vol 2).

[40] K. Lemström and J. Tarhio, *SEMEX - An Efficient Music Retrieval Prototype*, In: Proc. 15th annual Symposium on Login in Computer Science (LICS’2000), pp. 157-167, Santa Barbara, USA, June 26-29, 2000.

[41] K. Lemström and P. Laine, *Musical Information Retrieval Using Musical Parameters*, International Computer Music Conference, Ann Arbour, 1998.

[42] K. Lemström, *String Matching Techniques for Music Retrieval*, Ph.D Thesis, Report A-2000-04, University of Helsinki, 2000.

[43] K. Lemström, A. Haapaniemi, E. Ukkonen *Retrieving Music - To Index or not to Index*. The Sixth ACM International Multimedia Conference, 1998.

[44] K. Lemström, P. Laine, S. Perttu *Using Relative Interval Slope in Music Information Retrieval*, In Proceedings of the International Computer Music Conference (ICMC), pp 317–320, Beijing, 1999.

[45] H.B. Lincoln, *Some criteria and techniques for developing computerized thematic indices*, in Heckman, editor, *Electronische Datenverarbeitung in der Musikwissenschaft*, Regensburg 1967.

[46] MARSYAS, MusicAl Research SYstem for Analysis and Synthesis, Princeton University, http://www.cs.princeton.edu/ gtzan/marsyas.html

[47] . K.D. Martin. *A Blackboard System for Automatic Transcription of Simple Polyphonic Music*, M.I.T. Media Lab Perceptual Computing Technical Report No. 385, July 1996.

[48] K.D. Martin and Y.E. Kim, Y. (1998). *Musical instrument identification: a pattern-recognition approach.* Presented at the 136th Meeting of the Acoustical Society of America, Norfolk, VA, October, 1998.

[49] B. Matityaho and M. Furst. *4pMU7. Classification of music type by a multilayer neural network.* ASA 127th Meeting M.I.T. 1994 June 6-10.

[50] R. J. McNab, L. A. Smith and i. h. Witten, *Signal Processing for Melody Transcription*, Proc Australasian Computer Science Conference, PP. 301-307, 1996.

[51] R. J. McNab, L. A. Smith, D. Bainbridge, and I.H. Witten, *The New Zealand digital library MELody inDEX*, D-Lib Magazin, May 1997.

[52] R.J. McNab, L.A. Smith, I.H. Witten, C.L. Henderson and S.J. Cunningham, *Towards the Digital Music Library: Tune Retrieval from Acoustic Input*. ACM Digital Libraries 96, March 1996

[53] R.J. McNab, L.A. Smith, I.H. Witten and C.L. Henderson, *Tune Retrieval in the Multimedia Library*. Multimedia-Tools and Applications, Volume 10, 113-132, 2000.

[54] MELDEX, The New Zealand Digital Library's MELody inDEX, http://www.nzdl.org/musiclib

[55] J.A. Moorer, *On the Transcription of Musical Sound by Computer.* Computer Music Journal, 1 (4). 1977.

[56] Muscle Fish, http://www.musclefish.com

[57] D. Ó Maidín, *A geometrical algorithm for melodic difference*, Computing in Musicology 11, pp 65-72, 1998.

[58] J. Pickens, *A Survey of Feature Selection Techniques for Music Information Retrieval*, Technical report, Center for Intelligent Information Retrieval, Department of Computer Science, University of Massachusetts, 2001.

[59] M. Piszczalski, M. and B.A. Galler. *Automatic Music Transcription*. Computer Music Journal, 1 (4). 1977.

[60] L. Prechelt and R. Typke, *An interface for melody input.* ACM Transactions on Computer-Human Interaction, Volume 8 , Issue 2 (2001), pp. 133-149.

[61] Rolland, P.Y., Raskinis, G., Ganascia, J.G. 1999. *Musical Content-Based Retrieval : an Overview of the Melodiscov Approach and System.* Seventh ACM International Multimedia Conference, Orlando, November 1999. Pages 81-84.

[62] P. Salosaari and K. Järvelin, *MUSIR - a retrieval model for music,* Technical Report RN-1998-1, University of Tampere, Department of Information Studies, July 1998.

[63] E. Scheirer, and M. Slaney. *Construction and evaluation of a robust multifeature speech music discriminator.* Proc. 1997 IEEE ICASSP, Munich, April 1997.

[64] Search By Humming, Steven Blackburn, University of Southampton, http://audio.ecs.soton.ac.uk/sbh/

[65] I. Smulevich and E.J. Coyle, *The use of recursive median filters for establishing the tonal context in music*. In Proceedings of the IEEE Workshop on Nonlinear Signal and Image Processing, Mackinac Island, MI, 1997.

[66] A. Sterian, G.H. Wakefield, *Music Transcription Systems: From Sound to Symbol* Proceedings AAAI-2000, Workshop on Artificial Intelleigence and Music, Austion, Texas, July 2000.

[67] Themefinder, Stanford University, http://www.ccarh.org/themefinder/

[68] TuneSever, University of Karlsruhe, http://name-this-tune.com/

[69] G. Tzanetakis, P. Cook. *Audio Information Retrieval (AIR) Tools.*, Proceedings International Symposium for Audio Information Retrieval (ISMIR 2000) Plymouth, USA, Octomber 2000.

[70] G. Tzanetakis, G. Essi, P. Cook, *Automatic Musical Genre Classification of Audio Signals* Proc of the 2nd Annual International Symposium on Music Information Retrieval, ISMIR 2001 (Indiana University, Bloomington, USA, 15-17 Oct 2001).

[71] Y-H. Tseng, *Content-Based Retrieval for Music Collections*. SIGIR. ACM, 1999.

[72] A. Uitdenbogerd and J. Zobel, *Melodic Matching Techniques for Large Music Databases*. Proc. ACM Multimedia 1999.

[73] M. Welsh, B. Borisov, J. Hill, R. von Behren, and A. Woo, *Querying Large Collections of Music for Similarity*. Technical Report UCB/CSD-00-1096, University of California at Berkeley.

[74] http://www.wildcat.com/.

[75] G. Williams and D. Ellis. *Speech/music discrimination based on posterior probability features*, Proc. Eurospeech-99, Budapest.

[76] E. Wold, T. Blum, D. Keislar, and J. Wheaton. *Classification, search, and retrieval of audio* in IEEE Multimedia , 3(3), 1996.

[77] C. Yang. *Music Database Retrieval Based on Spectral Similarity* In International Symposium on Music Information Retrieval, 2001.