From Raw Polyphonic Audio to Locating Recurring Themes

Thomas von Schroeter[1], Shyamala Doraisamy[2] and Stefan M Rüger[3]

[1] T H Huxley School of Environment, Earth Sciences and Engineering
Imperial College of Science, Technology and Medicine
Prince Consort Road, London SW7 2BZ, England
ts9@ic.ac.uk

[2] Department of Multimedia
Faculty of Computer Science and Information Technology
University Putra Malaysia, 43400 UPM Serdang, Selangor D.E., Malaysia
shyamala@fsktm.upm.edu.my

[3] Department of Computing
Imperial College of Science, Technology and Medicine
180 Queen's Gate, London SW7 2BZ, England
s.rueger@ic.ac.uk

Abstract. We present research studies of two related strands in content-based music retrieval: the automatic transcription of raw audio from a single polyphonic instrument with discrete pitch (eg piano) and the location of recurring themes from a Humdrum score.

1 Introduction

In the age of digitalisation the production, recording and storage of music is easier than ever before. This calls for intelligent, content-based retrieval methods, and it would seem by the sheer volume of audio data that these methods need to be fully automated.

Designing and searching a truly musical database depends on 1) the chosen encoding of the musical data and 2) the method for comparison of musical sequences. Three different levels of encoding are generally considered: a) unstructured raw audio files based on digitised samples of sound waves, b) semi-structured such as MIDI (Selfridge-Field 1997) and c) one of many highly structured formats such as Humdrum (Huron 1997), Plaine and Easie (Howard 1997), DARMS (Selfridge-Field 1997) etc. The former is just the way performances are stored, whereas the latter contains musical features that describe music at a more appropriate level for content-based retrieval. It seems desirable to have access to all levels of encoding for a music piece, so that a particular performance can be archived and played in raw audio, but retrieved using features of a higher-level encoding.

It has been widely acknowledged that achieving automatic conversion between any of the aforementioned levels is extremely difficult if this is to pass the critical assessment of an experienced musician (even the audio playback from MIDI is hard when real-instrument sound is required). However, for the purposes of music retrieval, simple musical representations have proven to be successful, eg the n-gram encoding of successive pitch intervals as text strings where each letter stands for an interval or interval class (Downie and Nelson 2000). Indeed,

other retrieval systems such as the New Zealand Digital library MELDEX system (McNab, Smith, Bainbridge and Witten 1997; Bainbridge 1998), the joint ThemeFinder project of Stanford University and Ohio State University (Kornstädt 1998), or various "query by humming" approaches (Ghias, Logan, Chamberlin and Smith 1995; Blackburn and DeRoure 1998) use simple encoding schemes, eg as simple as a sequence of pitch directions (up, down, rest).

All these systems work more or less successfully on databases of folksongs or other monophonic music pieces, where a representation in terms of pitch and duration is relatively straightforward. In contrast to this, most Western-style music is essentially polyphonic. Here, the transcription from raw audio to a higher-level encoding is much more challenging. Section 2 surveys several approaches to the task of transcribing Western-style music that uses the diatonic scale and introduces some new ones. We do not address the issue of instrument identification at all; instead we limit our analysis to a single polyphonic keyboard string instrument with discrete pitch such as piano or harpsichord.

The second part of this article is concerned with comparison of musical sequences of polyphonic music. Humans, normally, will find it not difficult recognizing similarities between slightly modified or decorated musical sequences. However, one major computational difficulty lies in modelling the human perception of musical similarity; this and related problems have been described in (Selfridge-Field 1998). One example for the importance of similarity matching is the identification of the recurrence of a theme in a given piece of musical score. The theme here refers to the main melody or musical idea that forms the basis of the composition. Composers usually repeat this theme throughout the composition and upon repetition, this theme is usually modified to add variety to their composition or is repeated using some method defined by the form of the composition.

In Section 3, we discuss the problem of locating recurring themes in polyphonic music using the Humdrum score format (Huron 1997) — thereby addressing the similarity problem in a way that is more amenable to evaluation. Standard musical sequence matching algorithms use simple pitch-and-duration-based distance measures to compute matches or similarities (for a review see (Crawford, Iliopoulos and Raman 1998)) and our algorithm, a modification of (Mongeau and Sankoff 1990), is no exception. We used Bach's Fugues for testing. The data set was encoded in Humdrum format where the various voices are clearly separated.

2 Experiments on polyphonic music transcription

We consider transcription algorithms which convert raw audio into a list of fundamental frequencies over, and possibly varying in, time. We believe that for the purposes of retrieval this is sufficient to capture some essential (if crude) details of a performance, while avoiding the more involved interpretation problems usually associated with transcription, such as approximating the relative durations of neighbouring notes in terms of the fractions expressible by a conventional score, and introducing heuristics about how to group notes to parts.

Conceptually we divide the task into two subtasks: *time-frequency spectral analysis* and *fundamental line extraction*. Most research to date seems to have followed such a two-step approach, with some notable exceptions, for instance (Walmsley, Godsill and Rayner 1999).

2.1 Time-frequency spectral analysis

Of the many algorithms that have been used or proposed for time-frequency analysis of musical and speech signals, we implemented and tested the following three:

• short-time *least mean squared (LMS) filtering* (Choi 1997) extended to a sum of sinusoids with exponentially spaced frequencies, based on singular value decomposition;
• a decimated version of the *constant-Q spectrogram* due to Brown (Brown 1991; Brown and Puckette 1993); and
• a decimated version of the *Phase Vocoder* (Flanagan and Golden 1966; Puckette and Brown 1998).

None of them gave satisfactory results for polyphonic signals:

Tests of the *LMS approach* with synthesised signals consisting of no more than 2-3 sinusoids showed significant bias in the amplitude estimation when the actual frequencies did not exactly coincide with grid frequencies.

The *constant-Q spectrogram* and the *Phase Vocoder* were tested in detail with synthesised and acoustic piano signals with one and two parts. We were rarely able to see more than 2 or 3 partials (see Fig. 1 and 2 (a)); higher partials were usually too weak to be detected against the noise background. Moreover, the *frequency resolution* of spectrogram methods is limited to one sinusoid per channel. Thus spectral lines belonging to different tones which happen to fall into the same channel cannot be resolved. The channels must therefore have passbands of at most a semitone in width, with transition bandwidths in the region of a quarter tone. In order to achieve these filter specifications, very long filters are required, resulting in poor time resolution. Furthermore, *spectral leakage* leads to substantial amounts of energy in bands neighbouring a strong component. This is not a problem for the detection of a single component since the frequency estimates will be very close for both bands; however, for the same reason, Phase Vocoder estimates are subject to considerable bias when a neighbouring band contains energy from a *different* component. This effect has been studied quantitatively in (Puckette and Brown 1998).

Thus we eventually decided to abandon Fourier methods altogether in favour of *auto-regressive (AR)* estimators. We believe that we are the first to have applied AR methods to musical signals. A number of estimation schemes based on auto-regressive models have been published; initially we implemented and tested four of them with synthesised signals (von Schroeter 2000). Marple's *MODCOVAR* algorithm (Marple 1987) turned out the most accurate.

Its comparative advantage over Fourier methods is illustrated by the larger number of partials which it detects, typically 5 or 6 per note for the same signals in which Fourier and Phase Vocoder methods detect only 3 or 4 in all (see Fig. 2). In high quality piano recordings recently made available to us by Eric Scheirer at the MIT Media Lab, up to 14 partials are detected in a *monophonic* passage! We measured their anharmonicity and found it in good agreement with Fletcher's model (1964).

We therefore used Marple's algorithm as the basis for all further experiments. Its output is a list of poles in the complex plane for each frame. Poles are accepted or rejected according to their distance from the unit circle, which gives a measure of their relative weakness. The angle of the pole locations with the real axis gives the digital frequency ω which can easily be converted to pitch k (in semitones above some reference frequency ω_0) using the relation

$$\omega = \omega_0 \cdot 2^{k/12} \ .$$

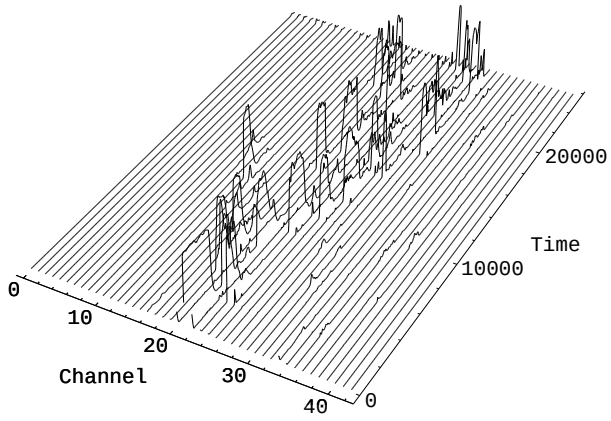

Figure 1: *Constant-Q spectrogram of a musical signal (bars 2-3 of a recording of Bach's Fugue in C Major from part 1 of the Well-Tempered Clavier, sampled at 5kHz and low-passed to 2.5kHz), obtained with a filterbank of Kaiser windows with transition bandwidth of 1/4 tone. The powers (vertical axis) are normalised according to* $\hat{p} = p/(1 + \|p\|)$*, where* p *denotes the spectrogram powers in each channel and* $\|p\|$ *their vector 1-norm). The reference frequency is 220 Hz for channel 0.*

2.2 Fundamental line extraction

2.2.1 Prior work

In contrast to music *synthesis* for creative purposes, the transcription problem seems to have received comparatively little attention, even though the first attempts in this area date from the late 1970's (Moorer 1977). The methods which have so far been proposed essentially fall into one or more of four categories which can be labelled as follows:

• *Correlation methods.* These methods are motivated by the use of correlations to find similarities between signals. Separately in each time frame, they compute either multiple autocorrelations of the spectrum (Tanguiane 1993), or convolutions of the spectrum with the spectral pattern of a single tone (Brown 1992). In both cases, tone hypotheses will appear as peaks in the resulting sequences. Neither method seems to have been tested with polyphonic *acoustic* signals.

• *Tone data bases.* This approach consists in the use of pre-recorded training data, reflecting the individual acoustical properties of the instrument when playing single notes, to aid detection of chords in the piece to be analysed (Rossi, Girolami and Leca 1997). The approach has been tested on acoustic piano signals with detection rates of 98% for scales and 92% for 4-part polyphony, albeit under somewhat idealized conditions.

• *Bayesian Networks* (Walmsley, Godsill and Rayner 1999). This is the most recent and perhaps the most principled approach to date as it makes explicit use of a mathematical tone model; however it also seems computationally very expensive. The interdependencies of parameters are modelled as a probabilistic network with a priori probabilities reflecting prior knowledge. Parameters are then estimated using Markov chain Monte-Carlo methods.

• *Context enlargement.* This approach complements the recorded and digitised signal by higher-level information in order to reduce the search space of tones compatible with the measured spectrum. Such additional information can be given in the form of AI-style rules, for instance rules governing the formation and rejection of note hypotheses based on signal shapes (Fernandez-Cid

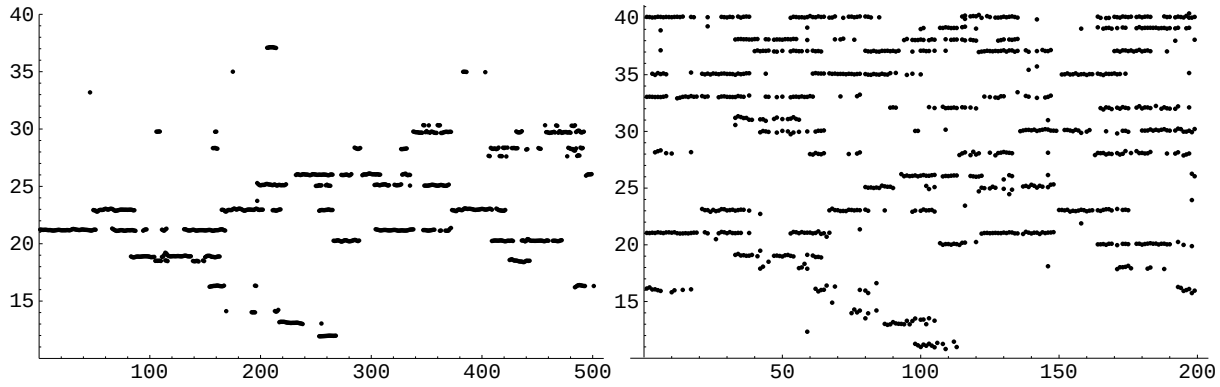

Figure 2: *(a) Left: Phase Vocoder time-pitch spectrum for the musical signal of Fig. 1 with 50 samples step size between frames. For each frame, dots indicate the precise pitch of components found in the channels in Fig. 1 satisfying* $\hat{p}_i > 0.05$*. (b) Right: AR spectrum obtained from the same signal by Marple's algorithm using 40 poles, 250 samples per frame (without overlaps), and a pole acceptance width of 0.01 to both sides of the unit circle.*

and Casajus-Quiros 1998) or assumptions about a particular musical style, see (Tanguiane 1993) and references there.

2.2.2 Towards a topological approach to transcription

Based on what little prior expertise was available, and guided by experimental results, we developed a suite of transcription algorithms, starting from a generalisation of the correlation approach, but finding ourselves naturally led to applying concepts with increasingly topological content. Here we sketch the beginning and the current stage of this development; typical results are shown in Fig. 3.

(a) Correlation peaks with a tone pattern. This approach is a modification of the one due to Brown (Brown 1992) to allow for continuous analysis frequencies, where the tone pattern is realized as a list of intervals with equal width on the pitch scale. Instead of computing the correlation with the tone pattern, we simply count the number of components covered by it. For each frame, this is a step function of the pattern offset. The algorithm extracts the mid point k of the first pitch interval in which the highest value of this function occurs, and discards from the frame all other pitches covered by the tone pattern centred at k, taking k as the fundamental pitch of a note hypothesis. This process is repeated until the remaining spectrum is empty (or so sparse that it does not give rise to a further note hypothesis).

Experimental results even with monophonic signals show that the convolution peaks found by this simple scheme are often below the spectrum, falsely suggesting a tone with missing fundamental. When we restricted the search of fundamental pitches to the range of pitches occurring in each frame (up to a tolerance), our algorithm detected about 90% of the notes in a 2-part acoustic piano signal, but also some spurious components (see Fig. 3(a)).

(b) Connectivity patterns in pitch and time. Closer scrutiny of Fig. 2 (b) reveals that in general the partials of a note have asynchronous onsets and vary in their decay time. Hence a purely frame-based algorithm is likely to fail. This led us to model tones as *two-dimensional subsets of the time-pitch spectrum* whose points are connected by two kinds of relations, namely

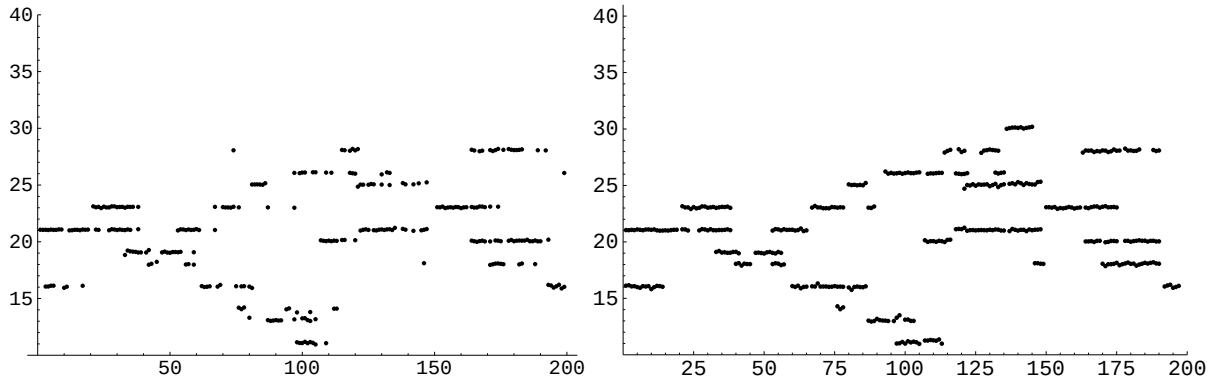

Figure 3: *Transcription results for the spectrum shown in Fig. 2(b) using (a) the restricted correlation peak heuristic (left), and (b) connectivity patterns (right).*

connectivity in time as continuation of a spectral line across neighbouring frames, and connectivity in pitch as a simultaneous pattern relation between points in the same frame. There is a two-level hierarchy of such combined time-pitch relations:

(P) two points can belong to a single tone pattern;
(T) they can belong to a single tone pattern with specified fundamental.

(T) implies (P). Thus (P) can be used to break down the input spectrum into connected components such that each tone pattern belongs to exactly one of these components; no prior knowledge of the fundamentals is necessary for this step. Within the (P) components we then list the maximal (T) components based at each of the lines, where delays of the fundamentals within the (T) components are tolerated up to a tunable threshold. This list is what we call a *covering table*; the partial ordering of its entries by inclusion reflects a partial ordering of possible tone hypotheses by their "explanatory power", although not necessarily in any probabilistic sense.

In each covering table we admit as chord hypotheses any minimal combination of its entries which covers the entire component. It can be shown that except in rather artificial circumstances, these *minimal covering sets* are unique and identical with the set of maximal elements with respect to inclusion, both up to multiply attributed lines. Thus we form this set, discard elements with too little, inharmonic, or disconnected "essential support" (defined as the set of points not shared with any other element of the covering table), and for all remaining components we extrapolate the fundamental to all frames intersecting the essential support in which it is not detected. The result is a point spectrum in which each point indicates an instantaneous fundamental pitch of a note hypothesis; thus the output format is richer in pitch details than the ordinary MIDI format and would in principle also accommodate pitch-variable instruments.

Preliminary experimental results for this method show a crucial dependence on the choices of parameters; with appropriate settings, detection was reliable for monophonic signals. For polyphonic signals with up to 6 visible partials per note, detection of correct notes is as good as with the convolution peak heuristic, but more of their decay phase is captured, and spurious components can almost completely be eliminated. However, multiple detections still occur (i.e. notes which are struck only once but detected more than once). See Fig. 3(b). Results for the high quality piano signals referred to earlier were badly affected by anharmonicity in combination with the implicit bias caused by alignment of tone patterns with the fundamental. Such signals would seem to require a combination of connectivity and clustering methods; these will be the

object of further study.

3 Locating recurrent themes

3.1 Related work

Early work in the area of comparing two musical sequences includes a system by Dillon and Hunter (1982) where the system was designed to identify variants of Anglo-American folk songs. For every song, 5 variants were generated based on the initial phrase and each variant was designed to capture one aspect of the melody in a form suitable for variant matching operations. Therefore, given a melody, the type of variant being seeked is generated from the query tune and this is matched with the database of songs indexed by the incipit and its variants using Boolean matching techniques. However, the idea of stating the variant before it is sought seems to defeat the purpose of automatically identifying the recurring themes.

Another system described in (Blackburn and DeRoure 1998) compares two musical patterns based on the contours, and one of their objectives is to retrieve songs through query by humming. The song database is indexed by sub-contours (pitch directions). A particular song is encoded as a long sequence of pitch directions (up, down, rest). Sub-contours based on the key length defined are obtained iteratively as segments from the long sequence. To query, part of a song to be retrieved is sung and this sub-contour is used to search the database of sub-contours. A near match set is obtained using a tree search. The concept of obtaining sub-contours can be used in breaking up a fugue into smaller sections. One problem is that the key-length has to be specified.

3.2 A basic comparison algorithm

In locating recurring themes, an algorithm is preferred which takes a more general approach where no specific type of modification is emphasised. The algorithm of Mongeau and Sankoff (1990) was chosen as a baseline and is detailed below.

Let $a = (a_1, a_2, \ldots, a_A)$ be a sequence of a certain number A of notes, each of which is encoded as a pair of pitch and duration and $b = (b_1, b_2, \ldots, b_B)$ be another sequence of B notes. We compute the dissimilarity $d_{A,B}$ of the two sequences a and b recursively as follows:
Boundary conditions

$$\begin{aligned} d_{0,0} &= 0 \\ d_{i,0} &= d_{i-1,0} + w(a_i, 0), i \geq 1 \\ d_{0,j} &= d_{0,j-1} + w(0, b_j), j \geq 1 \end{aligned}$$

General step, $i = 1, \ldots, A$ and $j = 1, \ldots, B$

$$d_{i,j} = \min \begin{cases} d_{i-1,j} & + & w(a_i, 0) & \text{(deletion)} \\ d_{i-1,j-1} & + & w(a_i, b_j) & \text{(replacement)} \\ d_{i,j-1} & + & w(0, b_i) & \text{(insertion)} \\ d_{i-1,j-k} & + & w(a_i, b_{j-k+1}, \ldots, b_j), 2 \leq k \leq \min(j, F) & \text{(fragmentation)} \\ d_{i-k,j-1} & + & w(a_{i-k+1}, \ldots, a_i, b_j), 2 \leq k \leq \min(i, C) & \text{(consolidation)} \end{cases}$$

The underlying idea is the one of the edit distance, and dynamic programming is used to obtain the series of transformations with the minimum distance. $w(a_i, b_j)$ is the distance score or weight associated with the ith note of sequence a and the jth note of sequence b. This score is a weighted sum

$$w(a_i, b_j) = w_{\text{interval}}(a_i, b_j) + k_1 w_{\text{length}}(a_i, b_j)$$

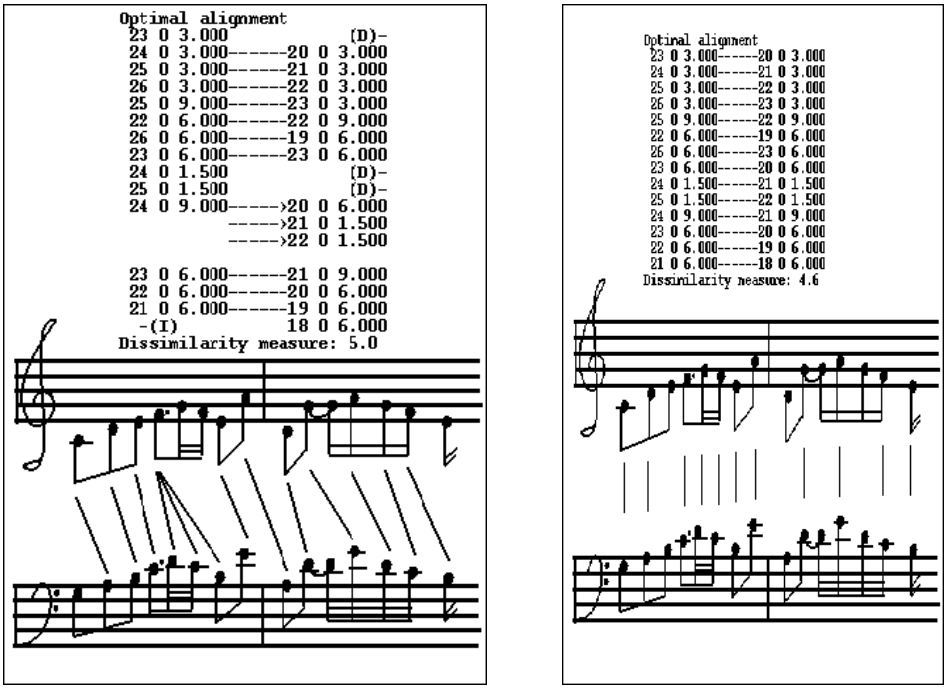

Figure 4: *The effect of different pitch weights on the alignment of sequences.*

of pitch and duration scores, $w_{\text{interval}}(a_i, b_j)$ and $w_{\text{length}}(a_i, b_j)$, the former being the weight assigned for a particular difference in pitch and the latter is the weight assigned for the difference in duration. The factor k_1 can be varied to reflect the relative contribution of pitch and duration. $w(a_i, 0)$, the weight for deletion, is the length of the deleted note a_i times k_1, since this can be viewed as a note a_i replaced by a note of length zero. Here, the pitch weightings would be zero and the weight contribution would only be based on duration weightings. Similarly, $w(0, b_j)$ is the length of the inserted note b_j times k_1.

For a fragmentation, w_{interval} is the sum of the interval weights between each note fragment and the original, and w_{length} is the difference between the total length of the replacing notes and the length of the replaced one; similarly in the case of consolidation. The constant F can be obtained by considering, where it would cost less to insert a number of terms than to fragment more than F elements. Therefore, it is not necessary to consider fragmentations of a_i into more than F elements (or, similarly, to consider consolidations of more than C elements into b_j).

Parameter and weight values are discussed in detail in (Doraisamy 1995). It should be noted that parameter and weight values affect the optimal alignment of the sequences under the algorithm. As an example, consider Fig. 4, where two sequences are compared which are a 4th apart. Weight measures that are sensitive to musical differences and the consonance of intervals were used in the left hand side comparison. However, different and perhaps less intuitive weight values yield a more appropriate optimal alignment in the right hand side. This also illustrates the difficulty this algorithm faces in dealing with two transposed melody lines.

Our experiments with Mongeau and Sankoff's algorithm used the first few notes of Bach's Fugue I of The Well-Tempered Clavier, Book I, with a number of variations such as key change, skipping notes, augmentation and diminution. We found good overall dissimilarity measure except for the following variations: 1) changing the rhythm of a melody line and 2) transposing a melody line into a different key. Both, unfortunately, are quite common variations employed by composers.

3.3 Suggested enhancements

In order to be able to identify transposition, augmentation and diminution we suggest a change in input format which incorporates the generation of a melodic and rhythmic contour.

3.3.1 Melodic contour

From the experimental results, one limitation identified is that the algorithm requires the pitch to be encoded based on the distance from the tonic. This poses a problem when sequences are automatically extracted from a music score, especially where sequences are extracted from modulated portions of the score. For such sequences, the pitch of that sequence is encoded based on its new tonic. Thus, pitches in the sequence would be considered not to belong to the original scale and this would cause weights based on semitone differences to be used, which happen to be much higher, resulting therefore in a high dissimilarity score!

For sequences that have been extracted from a modulated portion of the score pitches can be re-encoded as distances from the new Tonic. This means that some preprocessing would be required where one has to analyse the score where modulation had taken place accordingly. However, this defeats the purpose of automatically extracting sequences for comparison of a score.

If the data was encoded as pitch offset (the distance and direction each note moves from the note that precedes it) instead of absolute pitches (the note itself), then the algorithm would compare *melodic contours* (the patterns of the melody) instead.

3.3.2 Rhythmic contour

In the case of changing the rhythm, ie the duration lengths of the notes, the dissimilarity measure turns out to be high. If the duration was encoded as rhythmic ratio, one would arrive at a *rhythmic contour* which is invariant under the actual rhythmic value.

3.4 Implementation of a theme locator system

A system to locate a recurring theme was implemented in the following steps:
Extraction. We extracted the pitch and duration values from the kern representation of Humdrum. The theme is extracted from this simplified sequence. For now, the theme is taken as the first subject of the fugue. This was taken to be the voice with the first entry and ends when the answer begins on any other voices. The next voice (column) that comes in with the melody note is the first voice to be extracted as a sequence for comparison. This process continues until we obtain sequences for all the voices.
Contours. The absolute pitch and duration values are used to obtain rhythmic and melodic contours which are used as input to the comparison algorithm.
Comparison. In extraction, the theme and the voices were separated. Each voice is now one long sequence, and the theme is trying to be located in each of these long sequences. For particular long sequences, shorter sequences are extracted for comparison to be made whether that particular sequence extracted contains a recurrence of the theme.
Analysis. The obtained dissimilarities are compared against a threshold, and if below a certain threshold, the theme is deemed to recur at this position in the music piece.

The system developed is able to detect themes varied with three common methods of modification which are transposition, augmentation/diminution and addition or skipping of notes.

4 Conclusions and future work

We believe that our work has important implications for both spectral analysis and fundamental tracking as subtasks of polyphonic transcription. As for spectral analysis, we have shown that auto-regressive estimators are superior to spectrogram and Phase Vocoder methods in their capacity to resolve a sufficient number of partials. With respect to fundamental tracking, we believe that a synthesis of topological and clustering concepts can lead to a better model of a tone, lends itself to straightforward implementations in terms of standard graph searching algorithms, and thus offers considerable promise for more reliable note detection.

Although the resulting output of such a polyphonic analysis is somewhat richer in format than ordinary MIDI, it does not contain the details and the quality of a high-level encoding such as Humdrum. We are currently investigating how this gap can be closed with an automatic procedure. One major challenge seems to be the separation of voices from the audio recording. Once these challenges have been overcome, a traditional approach based on monophonic melody comparisons as outlined in Section 3 could be used to locate recurring themes or, more generally, to compare musical sequences. One would hope that, for the purposes of music retrieval and theme location, these challenges do not have to be mastered at a level to satisfy an experienced musician.

Acknowledgements: This work is partially supported by the EPSRC, UK.

References

Bainbridge, D. (1998). Meldex: A web-based melodic index search service. *Computing in Musicology 11*, 223–230.

Blackburn, S. and D. DeRoure (1998). A tool for content-based navigation of music. In *ACM Multimedia 98 - Electronic Proceedings*.

Brown, J. C. (1991). Calculation of a constant Q spectral transform. *J. Acoust. Soc. Am. 89*, 425–434.

Brown, J. C. (1992). Musical fundamental frequency tracking using a pattern recognition method. *J. Acoust. Soc. Am. 92*, 1394–1402.

Brown, J. C. and M. S. Puckette (1993). A high resolution fundamental frequency determination based on phase changes of the Fourier transform. *J. Acoust. Soc. Am. 94*, 662–667.

Choi, A. (1997). Real-Time Fundamental Frequency Estimation by Least-Squares Fitting. *IEEE Transactions on Speech and Audio Processing 5*, 201–205.

Crawford, T., C. S. Iliopoulos and R. Raman (1998). String matching techniques for musical similarity and melodic recognition. *Computing in Musicology 11*, 73–100.

Dillon, M. and M. Hunter (1982). Automated identification of melodic variants in folk music. *Computers and the Humanities 16*, 107–117.

Doraisamy, S. (1995). *Locating recurrent themes in musical sequences*. MSc Thesis, University Malaysia Sarawak.

Downie, S. and M. Nelson (2000). Evaluation of a simple and effective music information retrieval method. In *Proceedings of the 23rd International ACM SIGIR Conference*.

Fernandez-Cid, P. and F. J. Casajus-Quiros (1998). Multi-pitch estimation for polyphonic musical signals. In *Proc. ICASSP*, Volume 6, pp. 3565–3568.

Flanagan, J. L. and R. M. Golden (1966). Phase vocoder. *Bell Syst. Tech. J. 45*, 1493–1509.

Fletcher, H. (1964). Normal Vibration Frequencies of a Stiff Piano String. *J. Acoust. Soc. Am. 36*, 203–209.

Ghias, A., J. Logan, D. Chamberlin and B. C. Smith (1995). Query by humming — musical information retrieval in an audio database. In *ACM Multimedia 95 - Electronic Proceedings*.

Howard, J. (1997). Plaine and Easie Code: a code for music bibliography. In *(Selfridge-Field 1997)*.

Huron, D. B. (1997). Humdrum and Kern: selective feature encoding. In *(Selfridge-Field 1997)*, pp. 375–401.

Kornstädt, A. (1998). Themefinder: A web-based melodic search tool. *Computing in Musicology 11*, 231–236.

Marple, S. L. (1987). *Digital spectral analysis with applications*. Prentice-Hall (Englewood Cliffs, New Jersey).

McNab, R. J., L. A. Smith, D. Bainbridge and I. H. Witten (1997). The New Zealand Digital Library Melody index. *D-Lib Magazine*.

Mongeau, M. and D. Sankoff (1990). Comparison of musical sequences. *Computers and the Humanities 24*, 161–175.

Moorer, J. A. (1977). On the Transcription of Musical Sounds by Computer. *Computer Music Journal*, 32.

Puckette, M. S. and J. C. Brown (1998). Accuracy of Frequency Estimates Using the Phase Vocoder. *IEEE Trans. Speech and Audio Processing 6*, 166–176.

Rossi, L., G. Girolami and M. Leca (1997). Identification of polyphonic piano signals. *Acustica 83*, 1077–1084.

von Schroeter, T. (2000). Auto-regressive spectral line analysis of piano tones. Technical report.

Selfridge-Field, E. (Ed) (1997). *Beyond MIDI: the handbook of musical codes*. MIT Press, Cambridge, MA.

Selfridge-Field, E. (1998). Conceptual and representational issues in melodic comparison. *Computing in Musicology 11*, 3–64.

Tanguiane, A. (1993). *Artificial perception and music recognition*. Number 746 in Lecture notes in artificial intelligence. Springer-Verlag, Berlin/London.

Walmsley, P. J., S. J. Godsill and P. J. W. Rayner (1999). Polyphonic pitch tracking using joint Bayesian estimation of multiple frame parameters. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics, New Paltz (NY), 17th-20th October*.