**Machine Annotation of   
Traditional Irish Music**

**PhD**

**Bryan Duggan MSc, BSc.**

*School of Computing*

*Dublin Institute of Technology*

*Kevin St. Dublin 8, Ireland.*

**Supervisors:  
Prof. Brendan O’ Shea  
Dr. Mikel Gainza**

**Prof. Padraic Cunningham**

**Dublin Institute of Technology, School of Computing**

**December 2008**

**Declaration**

I herby certify that this dissertation which I now submit for assessment by the School of Computing, Dublin Institute of Technology on the programme of study leading to the award of Doctor of Philosophy is entirely my own work and has not been submitted for assessment for any academic purpose other than in particular fulfilment for the stated above.

Signed

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Bryan Duggan

Date

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Abstract**

**Keywords: Musical creativity, musical style modelling, traditional Irish music, transcription, signal processing, Machine learning**

**Acknowledgements**

**Table of Contents**

[1 Introduction 1](#_Toc204958161)

[1.1 Research aims 3](#_Toc204958162)

[1.2 Original Contribution 3](#_Toc204958163)

[1.3 Use cases 3](#_Toc204958164)

[1.4 Organisation 5](#_Toc204958165)

[2 Traditional Irish Music 6](#_Toc204958166)

[2.1 Tune Types 8](#_Toc204958167)

[2.1.1 Reel 8](#_Toc204958168)

[2.1.2 Jig 8](#_Toc204958169)

[2.1.3 Hornpipe 8](#_Toc204958170)

[2.1.4 Slides & slip jig 8](#_Toc204958171)

[2.1.5 Polka 8](#_Toc204958172)

[2.1.6 Mazurka 8](#_Toc204958173)

[2.1.7 Scottische 8](#_Toc204958174)

[2.1.8 Waltz 8](#_Toc204958175)

[2.1.9 Air 8](#_Toc204958176)

[2.2 Instruments 8](#_Toc204958177)

[2.2.1 Flute 8](#_Toc204958178)

[2.2.2 Uilleannn Pipes 9](#_Toc204958179)

[2.2.3 Harp 9](#_Toc204958180)

[2.2.4 Accordian & concertina 9](#_Toc204958181)

[2.2.5 Banjo 9](#_Toc204958182)

[2.2.6 Guitar 9](#_Toc204958183)

[2.2.7 Bouzuki 9](#_Toc204958184)

[2.2.8 Piano 9](#_Toc204958185)

[2.2.9 Bodhrán 9](#_Toc204958186)

[2.2.10 Lilting 9](#_Toc204958187)

[2.3 Collections 10](#_Toc204958188)

[2.4 Musical Creativity 11](#_Toc204958189)

[2.5 Style in Traditional Flute Music 15](#_Toc204958190)

[2.5.1 Ornamentation 15](#_Toc204958191)

[2.5.2 Breathing 18](#_Toc204958192)

[2.5.3 Regional Styles 19](#_Toc204958193)

[2.6 Traditional Music Sessions 21](#_Toc204958194)

[2.7 Conclusions 21](#_Toc204958195)

[3 Characteristics of Digital Music 23](#_Toc204958196)

[3.1 Pitch 23](#_Toc204958197)

[3.2 Note onset detection 25](#_Toc204958198)

[3.3 Loudness 28](#_Toc204958199)

[3.4 Chroma 28](#_Toc204958200)

[3.5 Timbre 28](#_Toc204958201)

[3.6 Mel-Filtered Cepstral Coefficients 28](#_Toc204958202)

[3.7 Wavelet analysis 28](#_Toc204958203)

[3.8 Spectral Centroid 28](#_Toc204958204)

[3.9 Ornamentation 28](#_Toc204958205)

[4 Melodic Similarity 29](#_Toc204958206)

[4.1 Melodic contour (Parson’s code) 29](#_Toc204958207)

[4.2 n-grams 29](#_Toc204958208)

[4.3 Geometric distance 30](#_Toc204958209)

[4.4 Implication-realisation 30](#_Toc204958210)

[4.5 Transportation Distance 30](#_Toc204958211)

[4.6 Edit Distance 32](#_Toc204958212)

[5 Content Based Music Information Retrieval 38](#_Toc204958213)

[5.1 Searching symbolic representations 38](#_Toc204958214)

[5.2 Searching audio data 42](#_Toc204958215)

[5.3 Hybrid approaches 42](#_Toc204958216)

[5.4 Conclusions 46](#_Toc204958217)

[6 Machine Annotation of Traditional Tunes (MATT2) 49](#_Toc204958218)

[6.1 System design 49](#_Toc204958219)

[6.1.1 Onset detection 50](#_Toc204958220)

[6.1.2 Breath detection 52](#_Toc204958221)

[6.1.3 Pitch spelling 53](#_Toc204958222)

[6.1.4 Edit distance matching 57](#_Toc204958223)

[6.2 Interface 58](#_Toc204958224)

[6.3 Experiment and results 59](#_Toc204958225)

[6.4 Conclusions 61](#_Toc204958226)

[7 Machine Annotation of Traditional Sets (MATS) 63](#_Toc204958227)

[7.1 Sets of traditional irish dance tunes 63](#_Toc204958228)

[7.2 MACHINE annotation of TRADITIONAL sets algorithm (MATS) 64](#_Toc204958229)

[7.3 Experiment and results 67](#_Toc204958230)

[7.4 Conclusions 70](#_Toc204958231)

[8 Conclusions & Future Work 71](#_Toc204958232)

[8.1 Conclusions 71](#_Toc204958233)

[8.2 Future work 71](#_Toc204958234)

[Appendix A – Publications 72](#_Toc204958235)

[References 73](#_Toc204958236)

**Table of Figures**

Figure 1: The M-Audio Micro Track II digital audio field recorder 2

Figure 2: Wooden flutes (Source: Author) 9

Figure 3: The tune "Contentment is Wealth" in the ABC format. 11

Figure 4: Characteristics required in a human listener for the cognition of musical style (Source: Author based on (Baroni 2006)) 14

Figure 5: An example of a *run* in ABC format (Source: Author) 17

Figure 6: Geographic origin of regional style (Source: Author based on (Keegan 1992)) 20

Figure 7: Waveform plots of a piano (top) and a wooden flute (bottom) playing the notes A to G (Source: Author) 26

Figure 8: the Onset Detection Function (ODF) for a musical phrase calculated using the Onset Detection using Comb Filters implemented by the author in Java (Source: Author) 27

Figure 9: The Themefinder user interface 39

Figure 10: thesession.org user interface 40

Figure 11: Musicians in a session compare tunes using TunePal (Source: Author) 40

Figure 12: Screenshots of TunePal running on a Windows Mobile Smartphone (Source: Author) 41

Figure 13: MELDEX Interface. A user can play a part of melody or record a query for transcription 43

Figure 14: High level diagram of the MATT2 tune annotation system 49

Figure 15: Signal and ODF plots of the first bar of the tune "The Boyne Hunt" 51

Figure 16: Fuzzy histogram of candidate note lengths from a 28 second phrase from the tune "The Hunters Purse" 54

Figure 17: Pseudocode for the fuzzy histogram quaver length calculator 55

Figure 18: Normalisation stages for the A part of the tune “Come West Along the Road” 56

Figure 19: Screenshot of MATT2 58

Figure 20: Top ten edit distances for a recording of the tune "The Golden Keyboard" 60

Figure 21: Waveform of the last phrase from the tune "Jim Coleman’s" and the first phrase from the tune "George Whites Favourite" played in a set 64

Figure 23: Pseudocode for the MATS set annotation algorithm 66

Figure 24: Edit distance profiles for three tunes played in a set 67

Figure 25: Filtered version of first graph in Figure 4. The dynamic threshold and detected troughs are marked 68

**Table of Tables**

Table 1: Possible features that characterise creativity in traditional Irish flute playing (Source: Author) 18

Table 2: Edit distance matrix for the strings “DFGDGBDEGGAB” and “DGGGDGBDEFGAB” with the minimum edit distance position highlighted 34

Table 3: Edit distance for the string “BDEE” in “DGGGDGBDEFGAB”. This string represents the first 13 notes from the tune "Jim Coleman's" in normalised ABC format 35

Table 4: Average edit distances for the closest match and the next closest match for tunes correctly and incorrectly annotated 61

Table 6: Correctly and incorrectly identified tunes 68

Table 7: Human & machine annotated turns 69

Table 8: Annotation accuracy 69

1. Introduction

In common with the folk music of many countries, repertoire in Irish traditional music is primarily acquired orally. Musicians playing Irish music learn tunes by hearing the tune played by fellow musicians in sessions, classes, workshops and from commercial recordings . It is common at workshops such as those held as part of the Willie Clancy Summer School for students to use electronic devices to record their classes. Increasingly students use digital audio field recorders such as the M-Audio Micro Track II, which record high quality audio directly to WAV or MP3 format (Figure 1). In this way, over the years musicians can acquire many hours of field recordings in standard audio formats. Similarly, organisations such as Na Píobairí Uilleann, Comhaltas Ceoltóirí Éireann and the Irish Traditional Music Archive have been acquiring field recordings of traditional music for over sixty years and these organisations now possess many thousands of hours of recordings in a variety of formats and on a variety of different media.



Figure : The M-Audio Micro Track II digital audio field recorder

In order for these archives to be useful, they must be annotated with appropriate metadata, such as tune names, time signatures, key signatures and instruments. Additionally for musicological and ethnographic study, archives could be annotated with stylistic metadata. The main goal of this PhD thesis is to develop algorithms for *automatically* annotating field recordings of monophonic traditional dance music. Several recent papers address the necessity of developing MIR (Music Information Retrieval) systems that are adapted to the specific requirements of ethnic music and also to the needs of musicologists studying ethnic music . This work presents the first attempt to develop a content based music information retrieval system adapted to the specific characteristics of Irish traditional dance music. The algorithms and systems proposed in this work adapt melodic dissimilarity measures to take account of such characteristics as slow onset times in windblown instruments such as the wooden flute and the tin whistle, the use of ornamentation, variation, breathing and the playing of tunes *segue* in sets. The work also takes advantage of the ABC music notation language, which has been developed especially for the transcription of Western traditional music and there exist over 7,000 traditional Irish, Scots and Breton freely available in ABC format from public databases. The ABC language has the advantage of being based on ASCII text and so tunes in ABC can be easily processed and analysed using algorithms for textual information retrieval. Although this work focuses on traditional Irish music, it is hoped that the techniques proposed can be generalised to other genres and instruments.

* 1. Research aims

The overall aim of this research is to develop new algorithms and systems for the annotation of traditional Irish dance music.

* 1. Use cases

This section presents several possible usage scenarios for the outputs of this research.

Maria is taking classes on the wooden flute at the Willie Clancy summer school in Milltown Malbay one year. The classes take place over 6 days from 10am until 1pm each day. Her teacher is flute maker Eamonn Cotter. Each day, Eamonn spends the first half of the class teaching new tunes to the students and the second half of the class discussing technique. As the class is quite advanced, the class is able to learn about two tunes per day. Maria uses a digital audio field recorder to record the classes each day. Eamonn encourages the students to learn the tunes by ear and therefore doesn’t give the students the notes for the tunes. Eamonn has forgotten several of the titles for the tunes. In addition to the tunes he teaches the class Eamonn, records some additional tunes for the students to study in their own time. At the end of the week Maria feels that she has learned so many tunes that she ends up mixing them up. She has about two hours of recordings made from the classes. Mixed in with the recordings of the class, Maria has also recorded random tunes played in pub sessions she has listened into that week. At the end of the week, when she returns home, Maria transfers the MP3’s of the recordings to her computer. She uses MATT2 to analyse the recordings and identify the tunes. These titles get saved in the ID3 tags of the files, so she can import the files into Windows Media Player.

Catherine is a professional flute player who is working on a new CD, with her brother John and piano player Felix. Having played music all her life, she feels she has a repertoire of at least a thousand tunes, but like many traditional musicians she has difficulty recalling the correct titles for much of her repertoire. She wishes to include tunes on the recording that she learned from local musicians when she was growing up. When arranging the sets of tunes for the recordings, she realises that several of the tunes she knows just by the name of the person who played the tunes. Several others have no name at all, even though she senses the tunes are commonly played. She plays a phrase from each of the unknown tunes and uses MATT2 to identify the tunes. Once she has the names of the tunes she looks them up in Breandán Breathnach’s Ceol Rince na hÉireann series of books and uses the bibliographic notes therein to write the CD notes.

Treasa works for the Irish Traditional Music Archive. One of her jobs is to digitise the analogue recordings that the archive receives. The archive is working on a project to make its collection available for streaming on the internet. The archive has just been bequeathed a set of recordings made between 1900 and 1930 by a collector in Chicago. The recordings are on wax cylinders and shellac discs and are in remarkably good condition. Treasa uses equipment in the archive to transfer the recordings to WAV format for inclusion in the public archive. When listening to the recordings Treasa is surprised to hear several unusual settings of common tunes. She uses MATT2 to annotate the WAV files with the tune titles. In one cases MATT2 returns a version of a tune from O’Neills Dance Music of Ireland as the closest match and the same tune as transcribed in the website thesession.org as the second closest match. Treasa feels that this is an example of how the interpretation of tunes can change as a consequence of regional style and the tastes of period. She adds a bibliographic note to the recording marking it as an example of this phenomenon.

* 1. Original Contribution

The development of a novel and useful recording annotation system for traditional Irish music represents an important contribution to the study of music information retrieval and to the traditional music community. In addition, the main specific contributions to knowledge are listed as follows:

1. The use of community generated collections of traditional tunes in ABC format in a content based music information retrieval system.
2. The development of a content based music information retrieval system that supports queries played on traditional instruments.
3. The development of a framework to compensate for interpretative style in queries to a content based music information retrieval system.
4. The development of a novel algorithm based on edit distance profiles to annotate sets of traditional Irish dance tunes.
5. The validation of the proposed algorithms using the “Cranfield Model” of Information Retrieval evaluation.
   1. Organisation

The remaining sections of this document are organised as follows:

1. Traditional Irish Music

Irish traditional music includes several musical *forms*. In the song tradition, both sean nós (“old style” singing in the Irish language) and singing in English exist. The baroque music of Turlough O'Carolan is also considered part of the tradition (Vallely 1999). This project however, is primarily concerned with traditional dance music, as played on the concert flute and tin whistle. The most common forms of dance music are *reels, double jigs* and *hornpipes*. Other tune types include *marches, set dances, polkas, mazurkas, slip jigs, single jigs and reels, flings, highlands, scottisches, barn dances, strathspeys* and *waltzes* (Larson 2003). These forms differ in time signature, tempo and structure. For example a reel is generally played at a lively tempo and is in 4/4 time (although played and transcribed as 8 quavers in a bar) while a waltz is generally played at slower pace and is in 3/4 time. The time signature, tempo and structure of a tune form are determined by the dance it accompanies. Most tunes consist of a common structure of two parts called either the first and second part or the A part and B part. Tunes are typically arranged into sets. A set consists of a number of tunes (commonly two or three) played sequentially. Each tune in a set is usually repeated two or three times (Vallely 1999). Certain common sets were originally put together to accompany set dances (Vallely 1999), while other sets have become popular as a result of being recorded by emigrant Irish musicians in America in the early part of the twentieth century. The origin of many sets of tunes is unknown and musicians often compile new sets “on the fly” in traditional music sessions.

Instruments used to play traditional dance music include the tin whistle, fiddle (violin), uilleann (elbow) pipes, accordion, concertina, harp and the banjo (Wallis and Wilson 2001).

Music is a creative art form and “individual expression” is a defining component of traditional Irish music (Breathnach 1977). Creativity in traditional music takes three forms:

1. The composition of new tunes.

2. The arrangement of tunes into sets.

3. The individual creativity of a musician in interpreting a tune.

This work focuses developing algorithms for content based music information retrieval that specifically address points 2 and 3 above. When a traditional musician plays a tune, it is never played exactly as transcribed, though unlike with jazz for example, traditional musicians never deviate from the structure or framework of the tune. In fact an experienced musician rarely plays the same tune twice, identically. Interestingly, there is no scope in traditional dance music for rubatto (except for micro-tempo artefacts). Instead, a musician will employ the subtleties of ornamentation and variation to interpret the tune (Larson 2003).

Ornamentation plays a key role in the individual interpretation of traditional Irish music (Canainn 1978). Ornamentation has a different meaning in Irish traditional music than its definition in classical music. In classical music, the expression is achieved by adding notes to the melody. By contrast, with the exception of the slide effects, Irish traditional music ornamentation is played on the beat, and alters the onset of the notes in a manner in which, it is argued, only one note will be heard (as opposed to two notes as in classical music) (Larson 2003). The usage of ornamentation is highly personal and large variations exist in the employment of ornamentation from region to region, instrument to instrument and from musician to musician.

* 1. Tune types

This section presents the background and history of the most common forms of traditional dance music. The most common forms of dance music are *reels, double jigs* and *hornpipes*. Other tune types include *marches*, *set dances*, *polkas, mazurkas, slip jigs, single jigs and reels, flings, highlands, scottisches, barn dances, strathspeys* and *waltzes* . These forms differ in time signature, tempo and structure. For example a reel is generally played at a lively tempo and is in 4/4 time (four crochets in a bar, though usually transcribed as eight quavers in a bar), while a waltz is generally played at slower pace and is in 3/4 time. Most tunes consist of a common structure of two parts called the *A* part and *B* part. Tunes are typically played as *sets*. Certain common sets were originally put together to accompany set dances (Vallely 1999), while other sets have become popular as a result of being recorded by emigrant Irish musicians in America in the early part of the twentieth century.

* + 1. Reel
    2. Jig
    3. Hornpipe
    4. Slides & slip jig
    5. Polka
    6. Mazurka
    7. Scottische
    8. Waltz
    9. Air
  1. Instruments

Fiddle

* + 1. Flute

The “Irish flute” is also known as the concert flute (because it is in concert pitch), the timber flute (because it is made from wood), the simple system flute or the fheadóg mhór (big whistle). It has six holes tuned such that the lowest playable pitch (all holes closed) is the D above middle C, and the instrument will play a D scale (D, E, F#, G, A, B, C#) as the holes are uncovered sequentially to shorten the resonant length of the bore. The basic flute is often augmented with the addition of up to eight keys (typically made from silver, mounted on wooden blocks) used to play pitches which are impossible to produce on the basic flute. Figure 2 depicts a 6 keyed wooden flute made from African black wood by Eamonn Cotter, an unkeyed made from African black wood flute made by Eamonn Cotter and an unkeyed bamboo flute made by Patrick Olwell in the key of F.



Figure : Wooden flutes (Source: Author)

Wooden flutes from the 19th Century were originally designed to play classical music, but with the invention of the Boehm system flute in 1832, wooden flutes became unpopular amongst classical musicians and thus came to be acquired by traditional musicians. Since the 1970’s, there has been a renaissance in wooden flute making and now many musicians play modern wooden flutes based on the 19th Century designs (Vallely 1999).

* + 1. Uilleannn Pipes
    2. Harp
    3. Accordian & concertina
    4. Banjo
    5. Guitar
    6. Bouzuki
    7. Piano
    8. Bodhrán
    9. Lilting
  1. Collections

There have been several notable initiatives to catalogue the cannon of Irish traditional music . Around the turn of the twentieth century, Francis O’Neill, the then police chief in Chicago, transcribed and documented a large body of dance tunes from immigrant Irish musicians.

In 1903, he published a book of his collected tunes entitled *The Music of Ireland*. The 1,850 tunes presented in the collection were classified according to tune-type (airs and songs, Carolan compositions, double jigs, slip jigs, reels, hornpipes, long dances, marches and miscellaneous). In 1907, he published *The Dance Music of Ireland – 1001 Gems*. This collection focused entirely on the dance music repertoire and contained many tunes published in his previous collection. Until the publication of Brendan Breathnach’s *Ceol Rince Na hÉireann* series O’Neill’s second book was considered the definitive source for traditional musicians and musicians would often refer to a tune by its reference number in the book .

ABC is a music notation language introduced by Chris Walshaw in 1991 . The format was designed primarily for folk and traditional tunes of Western European origin which can be written on one stave in standard classical notation . ABC files are ASCII text files and so can be edited by any text editor, without the necessity for special software. Each file (known as a *tune book*) can contain multiple tunes. File sizes are typically measured in kilobytes and this facilitates easy transmission by electronic means. The small size of ABC files also makes them an ideal medium for the storage of tunes on a memory constrained mobile device.

is the tune “Contentment is Wealth” in the ABC format. Each tune consists of a header section and a tune body. The header section contains amongst other fields, the title, composer, source, tempo, key signature, geographical origin and transcriber . As tunes can have several titles, the title field can be repeated for a given tune.

X:11

T:Contentment is Wealth

R:jig

M:6/8

K:Edor

GFG Eed|BAB EFG|FAF DdB|AFD D2f|gfe edB|BAB ~d3|BdB DFA|GED E3:|

|:ede Beg|bge gfe|dcd Adf|afd fed|ede Beg|bge gfe|BdB DFA|GED E3:|

Figure 3: The tune "Contentment is Wealth" in the ABC format.

The tune body contains the notation for the tune. The body encoding supports such features as ornaments, bar divisions, sharps, flats, naturals, repeated sections, key changes, guitar chords, lyrics and variations. There is an active and vibrant community supporting the ABC format and a range of tools have been developed for a variety of platforms and purposes.

Between 1997 and 2000, a group of musicians under the leadership of Dan Beimborn and John Chambers, undertook a grass roots project to transcribe three of O’Neill’s books to electronic format using the ABC music notation language. As copyright had expired on O’Neill’s original books, they made their work freely available on the internet .

Many of the tunes from O’Neill’s books are played differently by musicians today, as is normal with a living tradition. Around the same period (the late 1990’s) Henrik Norbeck collected nearly 2000 tunes in ABC format from various sessions and recordings. Again this collection was made freely available on the internet. This collection contains many modern settings of tunes from O’Neill’s books .

* 1. Musical Creativity

(Williamson, Thompson et al. 2006) identify reasons why authors have had difficulty characterising creativity. They suggest that it has been impossible to offer an unambiguous and broadly agreed on definition. Further they propose that creativity is difficult to isolate empirically and finally they suggest that creativity has an entrenched "mythology" especially in the arts world where it is construed as a mysterious, unknowable process. In this section, the problem of defining style in traditional flute playing is divided into two sub-problems. Firstly the concept that style is related musical creativity is proposed. Secondly, this section summarises approaches to the problem of what characterises style in traditional flute playing.

(Götz 1981) relates creativity to “making” and defines creativity as “the process or activity of deliberately concretising insight”. (Boden 1996) is extensively cited by authors seeking to understand creativity. Boden distinguishes two types of creativity. Psychological creativity (P-creativity) occurs when an individual has an idea which is novel to that individual, regardless of how many other individuals have had that same idea. Historical creativity (H-creativity) defines ideas that are novel not only to an individual, but also novel in the history of human endeavour. P-creativity is therefore judged by an individual. H-creativity is judged by society at large. The concept of two levels of creativity is also proposed by (Gardner, 1993b), who distinguished between “little c” and “big C” creativity.

There are examples in traditional music of both P-creativity and H-creativity as defined by (Boden 1996). Individual expression (P-creativity) is in fact a defining component of traditional Irish music (Breathnach 1977). When a traditional musician plays a tune, it is rarely played exactly as transcribed, though unlike with jazz for example, traditional musicians never deviate from the structure or framework of the tune. In fact, experienced musicians rarely play the same tune twice, identically. In the introduction to the revised edition of O’ Neill’s Music Of Ireland (originally published in 1906), Krassen describes a typical scenario:

“*It seems that on this particular occasion Touhey wanted to learn a tune from McFadden. He had McFadden play it for him several times and then tried his own hand at it. Of course McFadden had to play it again, pointing out several "errors." This happened a number of times until Touhey finally gave up, for McFadden was playing the tune a little differently each time through!*”

- (Krassen 1975)

A traditional musician will usually employ subtle variations, ornamentation, timbre and phrasing to interpret a tune (Larson 2003).

H-creativity by definition, more rarely occurs in traditional music. Some examples might include the introduction of the concert flute in the nineteenth century, the development of the ceili band form in the 1920’s, the renaissance of traditional music led by Sean O’ Riada and Ceolteori Cuailann in the 1960’s and the introduction of the Bouzuki in the 1970’s (Wallis and Wilson 2001).

The cognition of individual creativity implies that an individual musician demonstrates a style that can be recognised. (Meyer 1989) defines musical style as:

“*a replication of patterning…that results from a series of choices made within some set of constraints”*.

- (Meyer 1989)

(Keegan 1992) again associates the concept of style with creativity and claims that the technique and creativity of an individual and their musical style are one and the same thing.

(Baroni 2006) suggests that a listener can have different approaches to music which influences their perception of style. A listener’s approach can be:

*“a mere abandon to the flux of sounds where music is lived as an emotional stimulus and a source of immediate pleasure”*.

- (Baroni 2006)

He suggests that in this context a listener has little appreciation of the style of the musician. He continues by proposing that a listener must have an *objective approach*, *a precise knowledge* of the cultural conditions where the music was produced and must have *competence* to distinguish one style of music from another. He suggests that examples are categorised by comparing them to *prototype models* that represent its fundamental characteristics.



Figure : Characteristics required in a human listener for the cognition of musical style (Source: Author based on (Baroni 2006))

(Baroni 2006) describes an experiment carried out to establish the features used by a group of both experts and amateurs in a musical domain to categorise a piece of music. A group of 13 subjects listened to a recording of a fragment of a little known piece of music by the composer Donizetti. The subject group contained musicologists, professional and amateur musicians. Each subject was provided with a tape recorder to record the cognitive paths followed in order to identify the composer. The experiment demonstrated that those subjects who possessed “prototype models” or “stored memories” were able to identify the century and genre, form and instruments in the piece of music, in other words to classify the style. The experiment also demonstrated that those subjects who possessed a “lexicon” of music terminology were better able to classify the piece. The author concludes by explaining that the subjects used:

*“prototype, conceived as a hierarchical organisation of memorised listening experiences, orientated by historical knowledge”*

*-* (Baroni 2006)

to classify the music.

* 1. Style in Traditional Flute Music

There are a number of authoritive sources that describe characteristics that can define an individual musician’s flute style. These include Valley’s, “Timber: The Flute Tutor”, and his PhD thesis, “Flute Routes to 21st Century Ireland” (Vallely 2004), Larson’s “The Essential Guide to Irish Flute and Tin Whistle”, McCormack’s, “Fliúit: Irish Flute Tutorial”, Keegan’s MPhil thesis “Words of Traditional Flute Style” (Keegan 1992). In addition there is Casey’s “Traditional Irish Flute Music from East Galway A Regional study and Documentary Field Collection”. Additionally Tansey’s “The Bardic Apostles of Inishfree” (Tansey 1999), a profile of Sligo musicians contains references to ornaments not described in any of the other literature, (*bark*, *backstitch, run* and *pop*). In personal interviews (Tansey 2006) he has elaborated on the meaning of these terms. Although there are some disagreements in definitions of certain features, the literature generally agrees that flute style can be characterised by features that include use of ornamentation, phrasing (where a musician takes a breath), use of variation, staccato or legato playing (with throating/tounging attacks), the timbre a musician achieves with an instrument, tempo, choice of tune and choice of tune type.

* + 1. Ornamentation

(Larson 2003) defines ornamentation as:

“…*ways of altering or embellishing small pieces or cells of a melody that are between one and three eight-note beats long. These alterations and embellishments are created mainly through the use of special fingered articulations*.”

- (Larson 2003)

Fingered articulations are a defining characteristic of traditional Irish music. The sound of most articulations is very brief. Although generated by inserting additional notes, (Larson 2003) argues that the notes are played at such speed that they are not perceived as having a discernible pitch or duration. There are differing opinions as to the origins of ornamentation in traditional Irish music. (Larson 2003) suggests that ornamentation is derived from the playing of the *píob mór*, a mouth blown bagpipe that predated the development of the modern uilleann pipes. The  *píob mór* had no capacity for momentary interruptions to the flow of air and thus melodies were played as unbroken streams of sound. In order to generate a perceived stop between two notes of the same pitch, a musician would play a third note momentarily between the two notes.

(Tansey 1999) argues that ornamentation developed as an attempt to mimic the sounds of nature. He compares for example the sound of a *cran* to that of a sheep’s “baa” and postulates that the ornament was developed by shepherd’s who played wooden flutes while tending sheep:

“*I put it to you therefore that it had to come from the throats of birds, the wild animals, the ancient chants of our forefathers, the hum of the bees and the mighty rhythms of the galloping hooves of wild horses all moulded together…*”

- (Tansey 1999)

The main components of wooden flute ornamentation are now identified:

A *cut* is defined as an articulation used to separate two notes. A cut is articulated by playing a middle note momentarily at a higher pitch than the second note. The overall length of the two notes does not change when cutting and so the length of the second note must be shortened very slightly to accommodate the cut.

A *tap* (referred to in some sources as a *strike* or a *bounce*) is an articulation also used to separate two notes. A tap is articulated by playing a middle note momentarily at a lower pitch than the second note.

A *long roll* is articulation used to separate three notes. The second note in the sequence is cut and the third note is tapped. Again, the overall length of the three notes does not change. A *short roll* is similar to a long roll, but the first note in the sequence of three is dropped.

Concert flutes are usually pitched in D. As there is no note lower than a low D on the instrument, a tap on the low D is not possible. Instead, to execute a “roll” type ornament on a low D, a musician will play a *cran*. In order to play a cran, the musician replaces the tap with a second cut. The second cut uses a different note, usually higher than that of the first cut. This creates a “bubbling” sound typical of the playing of Matt Molloy. Not all musicians use crans, for example Catherine McEvoy does not play crans at all. Although (Larson 2003) suggests that crans can be done on any note, most other sources suggest that crans are only played on the low and middle D and E. They can be played long or short as with rolls.

With all of the above articulations, the actual pitch of the “extra” notes may vary depending on which finger the musician feels most comfortable lifting at speed (Keegan 1992). Using different fingers to perform the ornamentation also gives the ornament a specific character which can be part of a musician’s unique sound. An interesting example of this can be found Seamus Tansey’s 1975 recording “The King of the Concert Flute” (Tansey 1975).

(Larson 2003) suggests that trills are not common in Irish flute music, however an analysis of the corpus described in section **Error! Reference source not found.** finds this not to be the case. A trill is defined as a rapid alteration of the principal note and the note above it. A trill may begin on either the principal note or on the higher ornamental note. Trills are usually played for short durations in traditional music, with longer duration trills being considered too much of an allusion to classical music.

A *tight triplet* alsocalled a *treble* in (Tansey 1999) is a stepwise rising or falling sequence of three notes played in quick succession in the rhythm of two notes. A specific type of tight triplet mentioned in (Tansey 1999) is a *back stich* which he describes as a treble using the notes BCD. A *run* as described by (Tansey 2006) is a descending sequence of two tight triplets as illustrated in Figure 5. In the note sequence, the first four notes are played without the use of a run while the second sequence of six notes are two tight triplets, in other words a *run* on the four note sequence.

K:D

M:Reel

=cABG (3=cBA (3BAG

Figure : An example of a *run* in ABC format (Source: Author)

Switching between octaves on a wooden flute is achieved using a technique known as *overblowing*. (Hamilton 1990) describes how overblowing can be used as a technique to add variation to a performance by overblowing a phrase meant to be played in the lower octave of the instrument.

Overblowing is also used as a technique in the sounding of a *hard D*. A *hard D* is achieved on a wooden flute by overblowing the D in the lower register to the extent that the note is perceived as a group of harmonics of D that can be impossible to distinguish (Keegan 1992).

* + 1. Breathing

Phrasing in traditional flute music is easily identified as the timings in a performance of a tune where a musician takes a breath.

|  |  |  |
| --- | --- | --- |
| Ornamentation | Single-note | Cut |
| Tap |
| Multi-note | Roll |
| Cran |
| Triplet |
| Run |
| Breathing | Phrasing |  |
| Throating (attacks) |  |
| Overblowing |  |
| Timbre |  |
| Variation |  |  |
| Repertoire | Reels |  |
| Jigs |  |
| Hornpipes |  |
| Polkas |  |
| Slides |  |
| Scotisches |  |
| Strathspeys |  |
| Mazurkas |  |
| Tempo |  |  |

Table : Possible features that characterise creativity in traditional Irish flute playing (Source: Author)

Traditional music scores are not annotated with breath marks and it is up to an individual musician to decide where a breath should be taken. Taking a breath usually means leaving out a note or several notes from the score in a performance. Phrasing is therefore more obvious in music played on the flute than on any other traditional instrument (Keegan 1992). (Keegan 1992) in his interviews establishes that phrasing (and in particular the length of phrases) is a strong indicator of a particular regional and individual style.

In traditional Irish flute playing, *tounging* as used as a note attack by classical flute players is rarely used. Instead a technique called *throathing* is often used (the stop is produced by the throat rather than by the tongue) (Hamilton 1990). This can often result in the note following the attack to be overblown, (sometimes one of the harmonics of the fundamental rather than the fundamental itself is perceived).

On the flute, the timbre achieved by a musician can vary widely between a broad/breathy sound and a sharp/clear sound and naturally, volume also can characterise a style.

Table 1 summarises the possible features elaborated upon in this section.

* + 1. Regional Styles

(Canainn 1978) describes regional style as the common features which distinguish the majority of performances by musicians from a particular area. Until the 1940’s there existed distinct regional styles of flute playing attributed mainly to the isolation of rural communities prior to the advent of mass communication. Similarly the country as a whole was largely preserved from the influence of other cultures due to its geographic position and the isolationist economic policies of the early Irish Free State (Keegan 1992). (Keegan 1992) describes his work in understanding the cognition of regional styles of Irish flute music by conducting a series of interviews with prominent musicians. He reports that four regional styles were identified by his subjects, though his work suggests that the characteristics that distinguished these styles varied somewhat. The regional styles identified in his work are: The West Clare style, the Ballinakill/East Galway style, the Fermanagh/Northern style and the Sligo/Roscommon style. Figure 6 shows a map of Ireland with the locations of the four regions identified by (Keegan 1992).

The West Clare and Ballinakill/East Galway styles he describes as demonstrating much use of ornamentation and accidentals, with the melody played at a relatively slow pace. These styles differ in repertoire and use of breath articulation, with The West Clare style being characterised by the use of throathing to emphasise rhythm. The Ballinakill/East Galway style developed from the playing of the musicians in one of the first ceili bands (The Ballinakill Traditional Players). (Keegan 1992) suggests that the Ballinakill/East Galway sound is more legato, with an emphasis on melody rather than rhythm. This is evident in the repertoire played by musicians in that style, which contains tunes with several parts. The suggests that in the past a substantial group of East Galway musicians have adopted the Boeme system flute or other fully keyed instruments, which are more suitable for the repertoire which involve tunes in unusual keys and with accidentals.



Figure : Geographic origin of regional style (Source: Author based on (Keegan 1992))

The Fermanagh/Northern style he describes as being sparsely-ornamented, but with heavy stress on breath articulation techniques. He states that there exists two styles of phrasing. In some examples, there is an emphasis on natural-phrasing (regular two-bar phrases), while other musicians demonstrate short irregular phrasing, characteristic of the music of North Leitrim (and hence similar to the Sligo-Roscommon style).

There is a strong concentration of flute players in the Leitrim/Sligo/Roscommon area which (Tansey 2006) attributes to the prevalence of coal mining in the region. He argues that the flute was considered good for the development and health of the lungs of coal miners, constantly exposed to high levels of coal dust in their profession. Although (Keegan 1992)’s subjects reported contradictory opinions on many aspects of the Sligo/Roscommon style, they agreed that the style is very rhythmical because of the use of breath articulation and emphasis. They also suggest that the overuse of ornamentation is not characteristic of many musicians of the Sligo/Roscommon style (though he points out several notable exceptions).

* 1. Traditional Music Sessions
  2. Discussion

It is clear from this introduction to the domain of traditional Irish dance music that an MIR system for traditional dance music must deal with many special problems. Firstly and most obviously, the system should support the input of queries played in traditional instruments such as the flute, tin-whistle and the fiddle or lilted queries. Stylistic variation is very common even within the same performance of a tune and therefore any system developed needs to be robust to melodic variations. The use of ornamentation means that transcribed melodies are always augmented when performed. This will skew melodic similarity measures that depend on exact matches. Similarly, ornamentation involves inserting additional notes at higher and lower pitches would mean that any melodic dissimilarity measure that depends on melodic contour alone (section 4.1) would not be appropriate. On the other hand, the nature of the ornamentation present in traditional music is well understood and so section X.X proposes a method of compensation for ornamentation

The collection of tunes into sets played in a *segue* creates segmentation problems. An input query to a CBMIR system for traditional music may consist of multiple melodies played without an interval, in the same time signature and often in the same key. The challenge therefore is in segmenting a query appropriately so that each individual tune in a set can be annotated correctly.

Given the limited range of keys used to play traditional Irish dance music owing to the dominance of concert pitch instruments transposition invariance is not generally a requirement for MIR in traditional music.

Support for queries on traditional instruments or lilted queries

Ornamentation

Variation including scattering

Phrasing (breathing)

Sets

Legato playing (hard to detect onsets)

Repetition & Structure (no reason for short queries or incipit's)

Tempo variation

Any phrase (not just the incipit)

Transcription errors (false positives, false negatives, pitches)

The importance of the problem in the domain of cultural heritage preservation

Transposition?

* 1. Conclusions

1. Characteristics of Music

Introduction blagh blagh blach

* 1. Pitch

Pitch is “the perceived quality of a sound that is chiefly a function of its

fundamental frequency in --the number of oscillations per second ” (Randel 1986). The

graphical representation (e.g., \*, +, w, ½, etc.) where pitch is represented by the vertical

position of a note on the staff is the most familiar. Note names (e.g., A, B, C#, ..., etc.),

scale degrees (e.g., I, II, ...,VII), solfège (e.g., do, ré, ..., ti) and pitch-class numbers (e.g.,

0, 1, 2, 3, ..., 11) are also some of the many methods of representing pitch.

5

The difference between two pitches is called an interval. Intervals can be

represented by the signed difference between two notes as measured in semitones (e.g.,

-8, -7…, -1, 0, +1, ..., +3, etc.) or by its tonal quality as determined by the location of the

two pitches within the syntax of the Western theoretical tradition. For example, the

interval between A and C# is called a Major 3rd while the aurally equivalent distance

between A and D= is a Diminished 4th. Melodies can be considered sets of either pitches

or intervals perceived as being sequentially ordered through time.

The notion of key is included here as a sub-facet of pitch. The melodic fragment

EDCEDC (i.e.,“Three Blind Mice”) in the key of C Major is considered to be musically

equivalent to BAGBAG in the key of G Major. That is to say, that the melodic contours

(i.e., the pattern of intervals) are perceived by the listener to be equivalent, despite the

fact that the absolute pitches of the latter are higher than the former. In our experience,

singers are the most sensitive to the notion of key, for they must find works, or

transpositions of works, in a key that does not extend the absolute pitches of a melody

beyond their particular vocal ranges.

Autocorrelation is one of the oldest of the classical pitch

trackers[7]. Autocorrelation isolates and tracks the peak en­

ergy levels of the signal which is a measure of the pitch.

Referring back to figure 3, we see that the signal s(n) peaks

where the impulses occur. Therefore, tracking the frequency

of this peaks should give us the pitch of the signal.

In order to get the frequency of these peaks we can employ

autocorrelation as defined by:

R(l) =

1

X

k=\Gamma1

h(k)h(l + k) (3)

Unfortunately autocorrelation is subject to aliasing (picking

an integer multiple of the actual pitch) and is computationally

complex. We found our implementation of autocorrelation to

require approximately 45 seconds for 10 seconds of 44KHz,

16­bit audio on a 90MHz pentium workstation.

ffl Maximum Likelihood

Maximum Likelihood[14] is a modification of Autocorrela­

tion that increases the accuracy of the pitch and decreases the

chances of aliasing.

Unfortunately, the computational complexity of this method

makes autocorrelation look blindingly fast. A straight­forward

implementation in Matlab takes approximately one hour to

evaluate 10 seconds of audio on a 90MHz Pentium worksta­

tion. With some optimizations,we improved the performance

to approximately 15 minutes per 10 seconds of audio, but this

is still far too slow for our purposes. Therefore, we discarded

this method. For a detailed explanation of this method, the

reader may refer to [14].

ffl Cepstrum Analysis

Cepstrum analysis is the definitive classical method of pitch

extraction. For an explanation, the reader is directed to Op­

penheim and Schafer's original work in [10] or in a more

compact form in [11]. We found that this method did not

give very accurate results for humming.

The output of these methods can be construed as a sequence of

frequency estimations for successive pitches in the input. We

convert these estimates into a three­step contour representa­

tion by comparing each estimated pitch with the previous one.

In our system adjacent pitches are considered the same if they

are within a quarter­step of each other (on an equal­tempered

musical scale), but this parameter is adjustable.

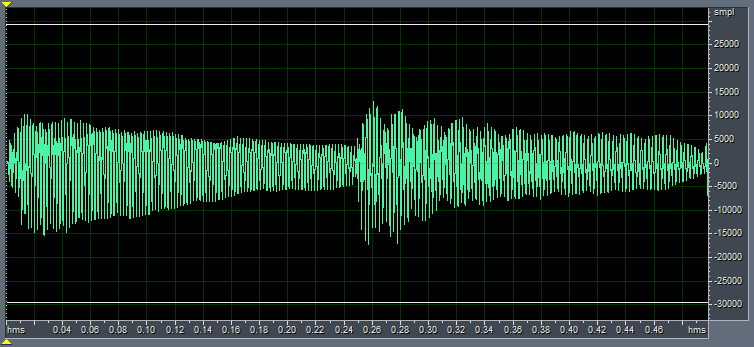
Fourier analysis

Transcription of the singing

melody in polyphonic music (Ryynanen & Klapuri 2006)

* 1. Note onset

To address the first problem, the authors propose an approach they refer to as Onset Detection using Comb Filters (ODCF). ODFC discovers harmonic characteristics of the input signal and is therefore more tolerant to energy changes in an input signal and is also better at detecting onsets in legato playing, where there is no significant change in energy at the onset of a new note.



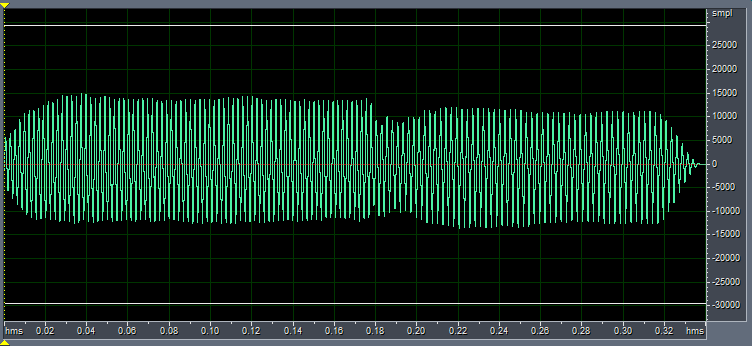


Figure : Waveform plots of a piano (top) and a wooden flute (bottom) playing the notes A to G (Source: Author)

Figure 7 compares waveform plots of a wooden flute playing the notes A to G legato with waveform plots of the same notes being played on a piano. As can be seen from this figure, there is a significant and detectible energy change in the plot from the piano between the offset of the first note and the onset of the second note, whereas with the notes played legato on the wooden flute there is a less detectable energy change from one note to the next.

To generate the Onset Detection Function (ODF), the input signal is first sampled at 44100Khz. The input signal is then segmented into overlapping frames of 2048 samples (approximately 46 milliseconds). Each frame overlaps with the previous frame by 75%. Each frame is then passed through a bank of twelve FIR comb filters.

A FIR comb filter works by summing the input signal with a delayed version of the same input signal. The delay of the filter is calculated as being 1 / frequency being filtered (the length in time of a single period of a waveform at the frequency). This has the effect of amplifying the frequency (or a harmonic thereof) in the input signal that matches the frequency being filtered. Thus, the energy of the input signal is doubled only if the peaks of the signal coincide with the peaks of the FIR comb filter. This will only occur for a given delay and its integer multiples.

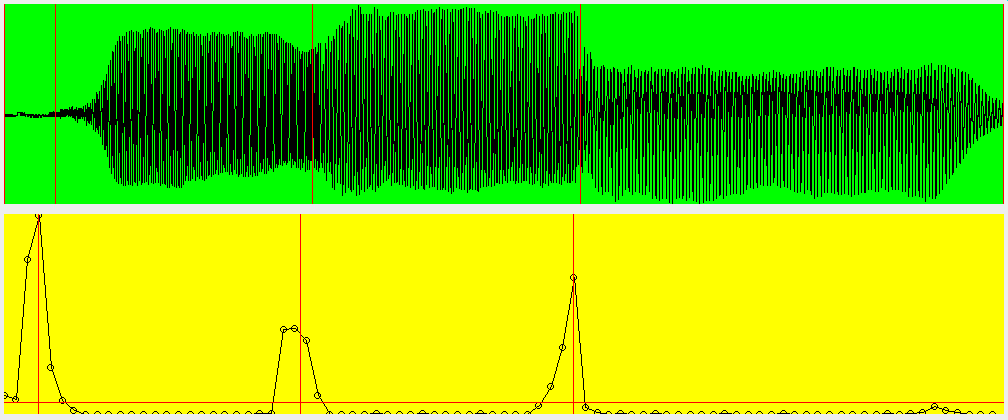


Figure : the Onset Detection Function (ODF) for a musical phrase calculated using the Onset Detection using Comb Filters implemented by the author in Java (Source: Author)

Twelve filters with different delays are used corresponding to the twelve semitones in the key of D3. For each frame of audio examined, the outputs of the audio passed through each of the twelve filters are calculated. A value for the ODF is then calculated as being the sum of the difference between the outputs of each of the twelve filters in successive frames, squared. In the case where the input signal changes from one note to another, this results in a peek in the ODF graph.

Using statistical techniques (average and standard deviation), a threshold is then calculated above which peeks in the ODF are recognised as being candidate note onsets. Figure 8 shows the ODF calculated in this way (using the system developed in Java and described in section **Error! Reference source not found.**) with an input signal of a wooden flute playing the notes D, E and F legato.

As illustrated, the onsets detected correspond to the onsets of each new note. The authors also propose that the ODF is filtered using a low pass filter to smooth the ODF, removing spurious onsets caused by noise in an onset.

(Dixon 2004) describes BeatRoot and the Performance Worm, two systems that the author claims, extract expressive features from a digital recording of a piece of music played by a human musician. They propose that although expression is contained in the physical features of the audio signal, such as amplitudes, frequencies and onset times, it is better understood when viewed from a higher level of abstraction, that is, in terms of musical constructs such as tempo, rhythm, pitch and timbre. These features are not directly measurable and the signal must be analysed to extract them.

BeatRoot models the perception of beats in a piece of music. BeatRoot first analyses the input signal to extract note onsets. Their first attempt to extract note onsets used a time domain algorithm that looked at the energy changes in successive frames. The authors claim that this approach worked well for percussive instruments such as the piano, but admit that the algorithm often detected false onsets and also failed to detect onsets for simultaneously sounding notes.

Their second attempt improves accuracy by separating the signal into frequency bands and looking for onsets in each band. (Gainza, Coyle et al. 2005) algorithm discussed earlier, which uses time domain comb filters seems more promising for detecting onsets in traditional music played legato on wind blown instruments as it is less sensitive to amplitude modulation in the signal. The system then uses an array of agents initialised with a tempo hypothesis. The agent then predicts further beats and is evaluated according to how well the predicted and actual beat times correspond. The system was evaluated against a corpus of Mozart sonatas and popular music and the authors claim a success rate of 90%.

* 1. Loudness
  2. Chroma
  3. Timbre
  4. Mel-Filtered Cepstral Coefficients
  5. Wavelet analysis
  6. Spectral Centroid
  7. Ornamentation

To detect ornamentation, the algorithm use heuristics derived from standard descriptions of traditional ornamentation (summarised in section 2.5.1 of this document). For example, to transcribe a cut on the note G, the algorithm looks for two consecutive G notes, separated by a momentary note at a higher pitch. The authors report a 60% success rate with single note ornament and a 40% success rate at detecting multi-note ornaments.

1. Melodic Similarity

This chapter presents methods for gauging similarity between melodies. These measures may be more accurately described as *dissimilarity* *metrics* as each of the methods presented calculates the distance between melodies. A higher distance implies that the melodies are less similar. describes a *metric* as a function on a set *S*,*d* : *S* × *S* → ℝ+ ⋃ {0} with the following properties:

1. Self-identity: For all *x* ∈ *S*, *d*(*x,x*) = 0
2. Positivity: For all *x≠y* in *S*, *d*(*x,y*) > 0
3. Symmetry: For all *x, y* ∈ *S*, *d*(*x,y*) = *d*(*y, x*)
4. Triangle inequality: For *x, y, z* ∈ *S*, *d*(*x,z*) ≤ *d*(*x, y*)

For measuring melodic dissimilarity in a way that agrees with human perception, a measure should possess the self identity property. This implies that two identical melodies should have a distance of 0. states that positivity is usually, but not always desired. This fact has been explored in section X.X, where the thesis is presented that a melody can be interpreted differently, but perceived as the same by a human listener. This is the premise behind the style compensation algorithms developed in section X.X. states that symmetry, while useful may not correlate with how humans perceive melodic dissimilarity. It is useful to consider these observations when regarding the melodic dissimilarity techniques described in this section.

Included in this chapter also are alternative representation schemes whose aim is to present a simplified representation of a melody so that comparisons can be more easily made.

* 1. Melodic contour (Parson’s code)

Parsons' thesis was that that a simple encoding of tunes that ignores most of the information in the melody can still provide enough information for distinguishing between a large number of tunes. Parsons' code includes only the directions of melodies. Each pair of consecutive notes is coded as “U” (“up”) if the second note is higher than the first note, “R” (“repeat”) if the pitches are equal, and “D” (“down”) otherwise. Rhythm is completely ignored. Figure 9 shows the first two bars from the tune "Banish Misfortune" in ABC format and in music notation, with the corresponding Parsons' code underneath.

=fed cAG|AGd cAG



DDDDDDUDUDDD

Figure : The first two bars from the tune "Banish Misfortune" in ABC format and in music notation, with the corresponding Parsons' code

Note that the first note of any tune is used only as a reference point and does not show up explicitly in the Parsons code at all.

(Downie 1999) represents monophonic melodies as string of note intervals (n-grams). His corpus contains 9354 folksongs. He uses three different encoding schemes to represent the interval set. Each encoding scheme is based on a different representation of the intervals. Set C3 represents all melodies as “a” (no interval) “b” (negative interval) or “c” (positive interval) Set C7 represents negative 1,2,3 as "b", "c","d"; positive 1,2, 3 as "B", "C", "D"; all negatives <=-4 as "d"; all positives >= +4

as "D". Set C15 represents negative 1to 6 as "b" to "g"; positive 1 to 6 as "B" to "G"; all negatives <=-7 as "h"; all positives >= +7 as "H".

* 1. Geometric distance
  2. Implication-realisation
  3. Transportation Distance

propose using transportation distances to measure melodic dissimilarity. First melodies are converted into *weighted point sets* in two-dimensional Euclidian space. The dimensions are the onset time (horizontal) and pitch (vertical) of each note, while the weight is the duration of the note. The Earth Movers Distance (EMD) between two weighted point sets measures the minimum amount of work required to transform one into the other by moving weight . Flow is measured as weight unit multiplied by ground distance. If *A* = {*a1, a2..am*} is a weighted point set such that *ai* = {(*xi, wi*), 1 ≤ *i* ≤ *m*, where *xi* **∈** ℝ and *wi* **∈** ℝ + ⋃ {0}, is the total weight of set *A*. The EMD can be formulated as a linear programming problem .

Given two weighted point sets A and B, *fi,j* is the flow of weight from *ai* to *bi* over the distance *di,j*. If *W* and *U* are the total weights of *A* and *B*, the set of all possible flows of *fi,j* is defined as by the constraints set out in Equation 1.

Equation

Constraint 1 allows moving weight from *A* to *B* and not vice versa. Constraints 2 and 3 limit the amount of weight that can be sent by the elements in *A* to their weights, and the elements in *B* to receive no more weight than the weight they can hold. Constraint 4 means that the total transported weight is the minimum of the total weights of the two sets. The total cost for transforming A to B is the sum of the weights *fi,j* multiplied by the distance *di,j*, normalised by the weight of the lighter set as per Equation 2.

Equation

EMD is a metric as described in the introduction to this chapter, if the ground distance is a metric and if the EMD is applied to two sets with equal weights. In the case of unequal total weights, the EMD does not obey the triangle inequality.

Proportional Transportation distance?? Need to include??

uses the Euclidian distance as the ground distance as per Equation 3.

Equation

In order to recognise augmented or diminished versions of a melody as similar, he proposes stretching the melody with the smaller maximum time coordinate, but leaving the durations (represented as point weights) of the notes unchanged.

He proposes two methods of making the measure transposition invariant. First, he proposes moving one or other of the melodies up or down until a minimum distance is reached, with a corresponding repeated application of the dissimilarity measure and increase in computational complexity. The second method he proposes is to transform one of the melodies so that the weighted average pitch is equal. This second method works to the extent that transposed versions of the same melody appear closer than other melodies from his test corpus of melodies. Time and pitch are also normalised so that transportations in time and pitch are equally expensive. .

This method was evaluated in a number of ways. For example the EMD was used to identify 80,000 incipits from anonymous composers by comparing the incipits against the RISM/A/II (Répertoire International des Sources Musicales International Inventory of Musical Sources)corpus in Plaine & Easie (Howard 1997) format. Using this method 3.9% of unidentified incipits could be identified. This compares favourably with . also reports a segmentation algorithm, where incipits from the corpus and queries are split into segments of between 5 and 16 notes. This is used to match incipits and queries in the case where different length musical sequences are to be matched. The author reports that this technique provided good results at the MIREX 2006 (Music Information Retrieval Evaluation eXchange) competition. The author attributes this to the fact that the distance measure is *continuous* (in that small changes to either of the melodies result in small changes to the distance) and works well with non-quantised data such as hummed queries.

* 1. Edit Distance

Edit distance, also known as *Levenshtein distance* or *evolutionary distance* is a concept from information retrieval and it describes the number of edits (insertions, deletions and substitutions) that have to be made in order to change one string to another. It is the most common measure to expose the dissimilarity between two strings.

The edit distance *ed(x, y)* between strings *x=x1 ... xm* and *y=y1 ... yn*, where *x, y* is the minimum cost of a sequence of editing steps required to convert *x* into *y.*  is the alphabet of possible characters and  is the set of all possible sequences of *ch* . Edit distance can be calculated using *dynamic programming* . Dynamic programming is a method of solving a large problem by regarding the problem as the sum of the solution to its recursively solved subproblems. Dynamic programming is different to recursion however. In order to avoid recalculating the solutions to sub problems, dynamic programming makes use of a technique called *memoisation*, whereby the solutions to subproblems are stored once calculated, to save recalculation.

To compute the edit distance *ed(x,y)* between strings *x* and *y*, a matrix *M1...m+1,1...n+1* is constructed where *Mi,j* is the minimum number of edit operations needed to match *x1...i* to *y1...j*. Each matrix element *Mi,j*  is calculated as per Equation 4, where = 0 if a = b and 1 otherwise. *M1,1* is the edit distance between two empty strings. The algorithm considers the last characters, *xi* and *yj*. If they are equal, then *x*1..i can be converted into *y*1..j at a cost of *Mi*-1*,j*-1. If they are not equal, *xi* can be converted to *yj*  by substitution at a cost of *Mi*-1*,j*-1 + 1, or *xi* can be deleted at a cost of *Mi-*1*,j* + 1 or y*j* can be appended to *x* at a cost of M*i,j*-1 + 1. The minimum edit distance between *x* and *y* is given by the matrix entry at position *Mm*+1*,n*+1.

|  |
| --- |
|  |

Equation

Table 2 is an example of the matrix produced to calculate the edit distance between the strings “DFGDGBDEGGAB” and “DGGGDGBDEFGAB”. The edit distance between these strings given as *Mm+1,n+1* is 3.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | D | G | G | G | D | G | B | D | E | F | G | A | B |
|  | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 |
| D | 1 | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| F | 2 | 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 8 | 9 | 10 | 11 |
| G | 3 | 2 | 1 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 8 | 9 | 10 |
| D | 4 | 3 | 2 | 2 | 2 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
| G | 5 | 4 | 3 | 2 | 2 | 3 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| B | 6 | 5 | 4 | 3 | 3 | 3 | 3 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| D | 7 | 6 | 5 | 4 | 4 | 3 | 4 | 3 | 2 | 3 | 4 | 5 | 6 | 7 |
| E | 8 | 7 | 6 | 5 | 5 | 4 | 4 | 4 | 3 | 2 | 3 | 4 | 5 | 6 |
| G | 9 | 8 | 7 | 6 | 5 | 5 | 4 | 5 | 4 | 3 | 3 | 3 | 4 | 5 |
| G | 10 | 9 | 8 | 7 | 6 | 6 | 5 | 5 | 5 | 4 | 4 | 3 | 4 | 5 |
| A | 11 | 10 | 9 | 8 | 7 | 7 | 6 | 6 | 6 | 5 | 5 | 4 | 3 | 4 |
| B | 12 | 11 | 10 | 9 | 8 | 8 | 7 | 6 | 7 | 6 | 6 | 5 | 4 | 3 |

Table : Edit distance matrix for the strings “DFGDGBDEGGAB” and “DGGGDGBDEFGAB” with the minimum edit distance position highlighted

|  |  |  |
| --- | --- | --- |
|  | if *xi=yi*  else |  |

Equation

An alternative expression of the edit distance equation which gives identical results is given in Equation 5, which is equivalent to Equation 4 because neighbouring cells in *M* differ by at most 1.

The algorithm can be adapted to find the lowest edit distances for *x* in substrings of *y*. This is achieved by setting *M1,j* = 0 for all *j* .*n+1.* In contrast to the edit distance algorithm described above, the last row *Mm+i,j* is then used to give a *sliding window* edit distance for *x* in substrings of *y* as per as Equation 6 (Navarro & Raffinot 2002).

Equation

An example of this variation on the edit distance applied to search for the pattern “BDEE” in “DGGGDGBDEFGAB” is given in Table 3. The minimum edit distance positions are highlighted. Variations on the edit distance algorithm have been applied in domains such as DNA analysis and automated spell checking and are commonly used in MIR systems .

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | D | G | G | G | D | G | B | D | E | F | G | A | B |
|  | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| B | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 1 | 1 | 1 | 1 | 1 | 0 |
| D | 2 | 1 | 2 | 2 | 2 | 1 | 2 | 1 | 0 | 1 | 2 | 2 | 2 | 1 |
| E | 3 | 2 | 2 | 3 | 3 | 2 | 2 | 2 | 1 | 0 | 1 | 2 | 3 | 2 |
| E | 4 | 3 | 3 | 3 | 4 | 3 | 3 | 3 | 2 | 1 | 1 | 2 | 3 | 3 |

Table : Edit distance for the string “BDEE” in “DGGGDGBDEFGAB”. This string represents the first 13 notes from the tune "Jim Coleman's" in normalised ABC format

It is understood from experiments with human listeners that humans perceive transposed melodies to be similar. Interestingly studies in animals have demonstrated that this ability is unique to humans . Hence there have been several attempts to adapt the edit distance algorithm for melodic dissimilarity to provide *transposition invariant* melodic dissimilarity.

for example use intervals between successive pitches to represent a melody for a dissimilarity comparison instead of the absolute values of pitches. Their algorithms can be understood by first considering the alphabet  to be = ℤ or ℝ, the integer or real alphabet. *x'* is the transposed copy of *x,* transposed by *t*, if *x' = (x1 + t) (x2 + t)...(xm + t).* For example if a melody was represented by the string *x* = {*3, 7, 5, 5, 8, 7, 7,5 ,3*} it could be relatively encoded as *x'* = {*4, -2, 0, 3, -1, 0, -2, -2*}. Using this scheme, there is naturally one less element in interval representation of the melody then in the original melody. The crucial property of this representation is that it is transposition invariant. In other words, if *x* and *y* are transpositions of each other, then *x'* = *y'*.

The limitation of this approach becomes apparent when one considers the case of an insertion or a deletion. Consider the two strings *x* = {*1, 2, 3, 4, 5*} and *y* = {*1, 3, 4, 5*}. The edit distance between these strings *ed*(*x, y*) = *1*. When converted to an interval representation these strings become *x'*={*1, 1, 1, 1*} and *y'* = {*2,1, 1*}. The edit distance between these strings *ed*(*x', y'*) = *2*. Hence each insertion and deletion has a double weighting on the calculation of the transposition invariant edit distance of two melodic strings. state that using interval encodings; when intervals are calculated on the fly from absolute sequences, a deletion or insertion transposes the rest of the melody and so as an alternative, they propose instead adapting a cost function for local transformations (insert, delete, replace) that is transposition invariant. A "standard" edit distance cost function considers the insertion, deletion and replacement of each pair of elements in *x* and *y*. In 's proposed transposition invariant edit distance calculation, the cost function is adapted to consider in addition, the previous and current characters in *x* and *y*. Equation 7 provides a transposition invariant method of calculating edit distance which is equivalent to Equation 4 for calculating transposition invariant edit distances.

|  |  |  |
| --- | --- | --- |
|  |  |  |

Equation

describe several algorithms for calculating transposition invariant distances (Hamming distance, longest common subsequence, edit distance) between strings. Their transposition invariant minimum edit distance *edt* is therefore given as per Equation 8.

Equation

Where *T =* {*xi - yj | 1 ≤ i ≤ m, 1 ≤ j ≤ m*}, in other words, the set of all possible values for *t* that would result in an alignment between *x* and *y*. In order to calculate *edt*(*x',y*), they propose using a brute force approach by calculating *ed*(*x + t,y*) for all *T*. This obviously increases the computational complexity of the algorithm over a straightforward edit distance calculation between the two strings. In order to speed up the calculation, they propose using a *sparse dynamic programming* algorithm. Sparse dynamic programming was first introduced by . The main idea behind these techniques is that only elements in a string associated with a match are visited. In order to achieve this, the authors propose calculating an ordered set of matching elements in *x'* and *y* for every value of *t* such that *Mt =* {(*i, j*) *| xi + t = yj*}. Using sparse dynamic programming, the computational complexity of the transposition invariant edit distance algorithm is *O*(*mn* log *n*) compared to *O*(*mn*) for standard edit distance. They also present a measure called "Longest Common Hidden Melody", which is a transposition invariant version of the longest common subsequence measure.

* 1. Hidden Markov Models
  2. Discussion

point out that when comparing music, the transposition invariant version of the edit distance is more useful except when it is known a priori that strings *x* and *y* being compared are in the same key.

Moreover, the style compensation algorithms proposed in Chapter 5 are independent of the metric used. See page 33 of Typke's book also about transposition. Edit distance difficult to use for polyphonic music, but widely used for monophonic comparisons. Also it is important to quantise melodies. Edit distances cannot deal with ornamentation. Chapter X proposes a method of adapting melodies so that they can be effectively compared using edit diatances. S

MIDI – No support for meta data or structure (Hoos 2001)

Contours/intervals – Too many false positives (Schlichte 1990) (Adams, Bartsch, & Wakefield 2003)(Lu, You & Zhang 2001)

But it simplifies the melody so much

that it cannot discriminate music very well, especially

when the music database is large.

current UDR string cannot describe sudden

pitch transitions

EMD – (No ornamentation compensation, cant use for segmentation - TYPKE 2007)

* 1. Conclusions

1. Content Based Music Information Retrieval

The approaches proposed in this work form part of a content based Music Information Retrieval (MIR) system for traditional Irish music. Music Information Retrieval can be defined as “the task of extracting from a large quantity of musical data, the portions of that data with respect to which some musicological statement is true” (Kassler 1966). The term Music Information Retrieval is first mentioned in the literature in (Kassler 1966). In this work the author presents MIR, an assembly like language for formulating musical queries and navigating scores. He suggests that MIR could form part of a “library of the future” although he recognises the limitations of the language proposed.

More recently (Typke, Wiering & Veltkamp 2005; Typke 2007) suggest that there are three main classifications of MIR systems: those for searching symbolic representations of music, those for searching audio data and systems that combine both approaches by first converting audio data to a symbolic representation and then searching for a match in a corpus of symbolically notated music. (Downie 2003) proposes analytic/production systems and locating MIR systems a classification analogous to the first two classifications. This section presents related work in each of the three classifications of system and concludes with an analysis of the suitability of the existing approaches explored to perform MIR for traditional Irish music.

* 1. Searching symbolic representations

Symbolic MIR has its roots in dictionaries of musical themes such as . Monophonic music can be represented as a one-dimensional string of characters, where each character represents a musical note. String can be made up of characters representing pitches, pitch intervals or melody contours. In systems that use this format, standard string matching algorithms such as Knuth-Morris-Pratt, Boyer-Moore, Levenstein (Edit) Distance (Section 4.6), longest common sub-sequence or regular expression searching have been applied (Navarro & Raffinot 2002).

describes Themefinder. Themefinder provides a web-based interface to the Humdrum *thema* command. The *Humdrum Toolkit* is a set of software tools for music researchers that manipulate ASCII data conforming to the Humdrum syntax . The *thema* command allows searching of corpora of musical themes or incipits. There are almost 10,000 themes in the Themefinder collection encoded in the kern music data format (a markup language for musical scores). Queries can match the incipit or any part of a theme in the corpus, but require knowledge of the *Humdrum Syntax* to input. Figure 9 is a screenshot of the interface to Themefinder which illustrates many of the query possibilities of the system.



Figure : The Themefinder user interface

(Downie 1999) uses the encoding scheme for melodies discussed in section 4.1 to build a symbolic MIR system using the SMART Information Retrieval System. The SMART system has the advantage of being an off the shelf textual information retrieval system. He builds n-grams of note interval sequences where n = 4, 5, and 6, by treating melodies as long strings of intervals and taking all substrings of a fixed length. His hypothesis is that there is an equivalency between interval-only melodic n-grams (i.e. “musical words”) and “real words,” intervals and letters. To retrieve matching melodies, he uses the *tf \* idf* (TermFrequency \* Inverse Document Frequency) ranking method from information retrieval and text mining. *tf \* idf* is a statistical measure used to evaluate how important a word is to a document in a corpus. The importance increases proportionally to the number of times a word appears in the document but is offset by the frequency of the word in the corpus. Variations of the *tf \* idf* weighting scheme are used by search engines as a tool in scoring and ranking a document's relevance given a user query. Downie uses melodic strings, extracted from melodies in the corpus as queries to evaluate the system. He artificially creates expansion, compression, repetition and omission errors in the queries to simulate potential error types that might be introduced by human subjects.

*GuidoMIR* is a symbolic MIR system that has a native corpus of melodies in the Guido/XML music notation language. The authors claim that using a symbolic musical score language such as Guido/XML has a number of advantages over MIDI, a format designed for playback. They cite the ability to store meta-data with the melody as the main advantage, but list several others. They also do not use any form of database engine and instead their system in built entirely in Perl and uses a database of flat files. Although their corpus is text based, the authors use a probabilistic matching algorithm based on first order Markov chains to match queries to corpus strings. Their system supports queries based on both pitch and rhythm.

Website thesession.org is not discussed in the literature, but is important because it contains a collection of over 7000 traditional Irish dance tunes in ABC format (section 2.3) entered by the traditional music community, which can be searched using text queries by any of the metadata associated with a tune or melodic queries in the ABC language. The website is significant, because unlike much of the work discussed in this chapter, it is supported by an active community of thousands of musicians who regular contribute tunes, report on traditional music sessions (Section 2.6) and engage in lively discussions. Figure 10 shows an example of the search results generated for the query “broom”.

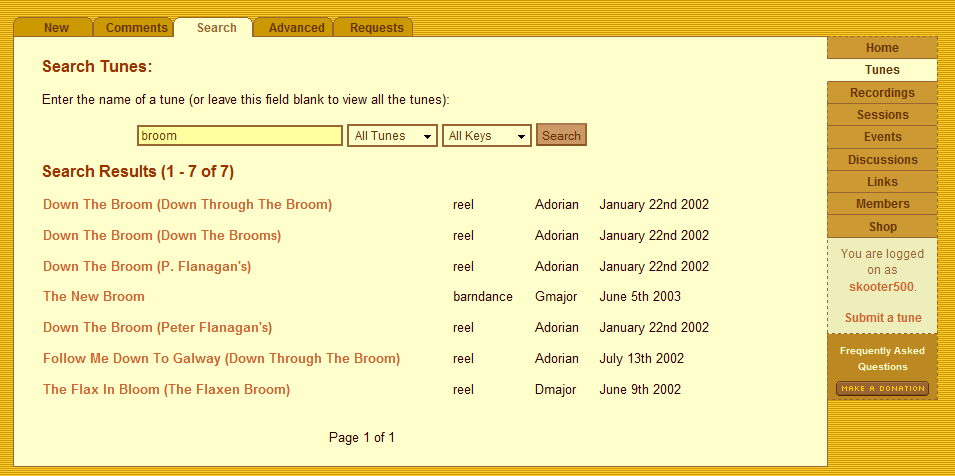


Figure : thesession.org user interface

*TunePal* is an MIR system whose main advantage is that it runs on a mobile device such as PDA or smartphone and so can be used in traditional music sessions and workshops. Figure 11 shows musicians comparing tunes using TunePal at a traditional music session.



Figure : Musicians in a session compare tunes using TunePal (Source: Author)

TunePal has a corpus of approximately 5000 traditional Irish dance melodies in ABC format drawn from transcriptions of O’Neill (O'Neill 1903; Krassen 1975; Chambers 2007) and Henrik Norbeck (Norbeck 2007). The system supports text queries on melodies or any of the meta-data such as tune name, type or composer. For melodic queries, the system requires knowledge of the ABC language. It has an elementary query normalisation algorithm that normalises text queries into the same register and removes ornamentation from corpus strings, but otherwise it requires to exactly match strings from the corpus. TunePal’s main goal is as an *aid memoir* for a musician who wants to play a tune, but can remember the name the tune and not the melody. Hence matching melodies can be easily converted to MIDI and played back at an appropriate tempo. Figure 12 shows screenshots of TunePal running on a Windows Mobile smartphone.

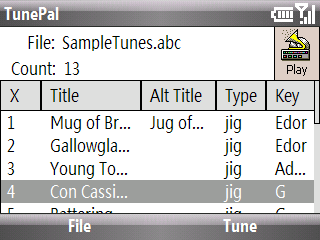
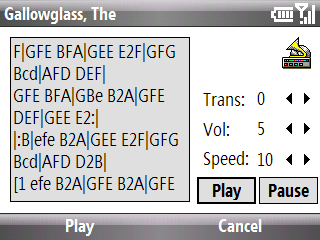
 

Figure : Screenshots of TunePal running on a Windows Mobile Smartphone (Source: Author)

(Lemstrom et al. 2003)describeC-Brahms. C-Brahms uses 9 different algorithms that support monophonic, polyphonic, rhythm invariant, transposition invariant, partial or exact matches for queries against a corpus of polyphonic music in a database of symbolically encoded music, drawn from MIDI files . The algorithms use a number of different techniques, including dynamic programming, bit-parallelism and a two-dimensional geometric representation of music. In the latter algorithms, music is a represented as horizontal line segments in Euclidean two-dimensional space. The horizontal axis represents time and the vertical axis, the pitch values. These algorithms are discussed in detail in section 4.6. C-Brahms has a public user interface available on the web so that the different algorithms can be evaluated.

PROMS ?? Not much detail in the paper so maybe leave it out.

* 1. Searching audio data

*Shazam*

*MusicDNS*

* 1. Hybrid approaches

Most research into hybrid MIR systems has focused on developing query by humming (QBH) interfaces to corpora of symbolically annotated melodies. Query by humming describes music information retrieval systems where audio clips of singing, humming or whistling act as queries. The premise is that if user wants to retrieve a melody from a large collection of music, a natural option is to sing, hum, or whistle a part of the melody into a microphone and let the system retrieve the matching melodies. The QBH task can be divided into two subproblems (Ryynanen & Klapuri 2008):

1. Converting a query into a format which enables searching
2. Matching the query with melodies in the corpus.

The former problem is one of automatically transcribing a query into a sequence of note events, whereas the latter is the problem of measuring

melodic dissimilarity between the query string (which may contain errors) and strings from the corpus.

*Cornell’s Query By Humming* is one of the earliest examples available of a Query-By-Humming system. It has a small corpus of 183 pieces of music in MIDI format stored in a flat file database. Pitch tracking is performed using Matlab, chosen for its built in audio processing facilities. The system transcribes hummed queries into Parsons’ Code (Section x.x) (U, D, S) using a modified autocorrelation algorithm (section X.X) . The corpus is then similarly converted to Parsons’ Code and matched against a query using ’s approximate string matching algorithm. This algorithm matches strings with at most *n* errors. The authors report a success rate of 90% using their techniques for queries of between 10 and 12 characters.

*MELDEX* (McNab et al. 1997; McNab et al. 1996; McPherson & Bainbridge 2001) has a pitch tracking interface that allows users to sing queries. The system depends on the user separating each note by singing *da* or *ta*. The articulation of the consonant is used to detect the onset of each note. As queries were generated by humans, they naturally contained errors. (Downie 1999) has classified the errors into four types: Expansion, Compression, Repetition, and Omission. Figure 13 shows a screen shot of the interface to a typical MELDEX screen.

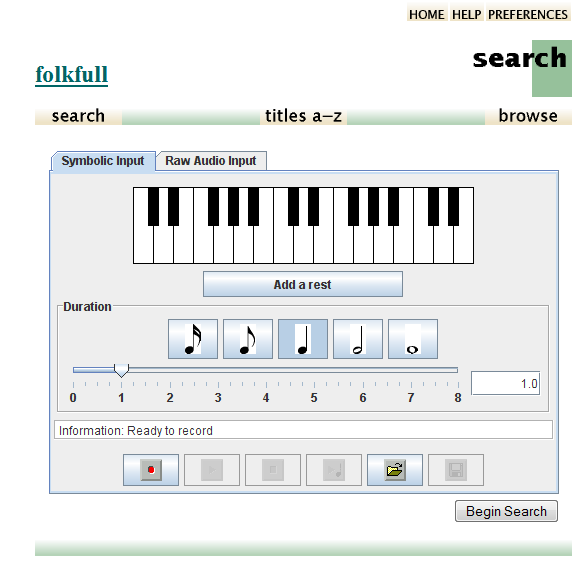


Figure : MELDEX Interface. A user can play a part of melody or record a query for transcription

MELDEX has a database of approximately 10,000 folk songs, compiled from the Essen collection (ref). The system uses the approximate string matching methodology of (Mongeau & Sankoff 1990) . This methodology was designed explicitly for the musicological analysis of melodic strings. Melody contour searches use interval direction method (section x.x). Matching melodies are ranked based on the degree of similarity between query and the items returned. Initially, MELDEX supported queries based on incipit’s , however subsequent improvements facilitated the matching of queries where the match occurs not only in the incipit, but also anywhere within a melody . Reported performance of the system is quite poor, with simple, exact match searches, taking an average of 500 ms to perform and 20 note approximate search pattern, requiring approximately 21 seconds.

*Musipedia* (previously known as *Tuneserver*) (Prechelt & Typke 2001) is a web-based MIR system that supports queries entered by whistling, playing on a virtual piano keyboard, tapping the rhythm on the computer keyboard, or entering the melodic contour. For whistled input, the audio is first sampled and a Fast Fourier Transform is then used estimate pitch. Onsets are noted using a combination of *silence windows* and pitch changes between consecutive frames of audio. The audio is then converted to Parsons’ code (section X.x) and a melodic contour search calculates the weighted edit distances between the query and strings from the corpus. Results are ranked in order of ascending distance from the query. The authors report a success rate of approximately 80% for queries with an average of 16 notes, where the correct melody was within the top 40 matches. The correct melody was returned as the closest match in 44% of queries. The authors ascribe mistakes to transcription errors and queries that were too short to discriminate similar representations of different melodies. The front end to Musipedia is also known as Melodyhound. Interestingly, although Musipedia contains traditional Irish dance tunes as part of its corpus, it does not generate positive results when queries are presented in the form of melodies played on the tin-whistle or wooden flute.

A later implementation of Musipedia supports a pitch and onset time-based search by representing the query into a weighted point set and calculating the Earth Mover's Distance (Section X.x) for each query point set and pre-computed point sets representing segments of melodies from the database. The "query by tapping" method that only takes the rhythm into account uses the same algorithm as the pitch and onset time method, but assumes all pitches to be the same. The system accelerates searches using an indexing technique based on vantage objects (Typke, Veltkamp & Wiering 2004; Typke et al. 2003).

(Lu, You & Zhang 2001) describe a QBH MIR system that represents queries as a triplet consisting of pitch contour, pitch interval, and duration, where pitch contour is U, or D, pitch interval is difference between the frequencies of two consecutive notes and duration represents how long a note is played or hummed. They convert their midi corpus to this format using a heuristic to extract the melody line from the MIDI representation of the audio. To convert audio to a query, they use an energy based onset detection function to determine the onsets of new notes in query audio. They point out the flaw in this method given that humans usually hum melodies *legato* and hence their algorithm misses onsets. Their corpus consists of approximately 1000 melodies in MIDI format. To match a query to a melody, their system first calculates the edit distance between the query and strings from the corpus. Strings whose edit distances are above a threshold are discarded. Strings for further consideration have interval and duration similarity calculated. They describe this as a “hierarchical matching algorithm”. The final measure of similarity is the weighted sum of the three similarities. They observe that people hum the pitch variations more correctly than rhythm and conclude that errors are more likely to involve rhythm than pitch interval. Hence they assign a larger weight to the duration similarity. In 74% of queries, the correct song was listed among the first three matches and that 59% of queries, the corresponding correct song was retrieved as the first match.

Ceolaire ??

*Fast melody Finder* (FMF) (Rho & Hwang 2004) is a web based music information retrieval prototype whose key feature is that it indexes the corpus according to a scheme known as FAI (Frequently Accessed Index) (Rho & Hwang 2004). The principal behind FAI is that a piece of music is often identifiable from a few specific melody segments of the overall melody. In FAI, segments are automatically induced from previous user queries. Each entry in the FAI structure has four variables: Access Count, Age, Repetition and Size. The authors propose an index maintenance system that, for example, supports merging of similar indexes.

Their prototype system has a corpus of 12000 MIDI files, which they pre-process to extract meta data in XML format such as time and key signature. Melodies are represented as pitch (U, D, S) and time contours (L, S, S) (Section X.x). Queries can be input by humming or by drawing the melody on a graphical representation of a 5 line stave. The system presumably incorporates a transcription subsystem, but this is not discussed in the work. Matching is achieved using the Boyer Moore algorithm initially to search for an exact match and if an exact match is not found the system falls back to calculating the edit distance using dynamic programming. Index entries are searched in order of access count. The authors present results which indicate that queries using both pitch and time contours are more accurate that pitch contours alone and also that their indexing scheme increased the performance of the system.

describe a QBH system that uses locality sensitive hashing to speed up retrieval of matching melodies. They use a corpus of 6030 melodies in MIDI format. They use a transcription algorithm described in detail in (Ryynanen & Klapuri 2006). This algorithm uses a frame based pitch salience estimator to measure the strength of different fundamental frequencies in successive frames (section x.x). The algorithm also applies a musicological model to filter note transitions. As an output, the algorithm produces a sequence of notes in the format <pi, bi, ei> where pi is MIDI note number, bi is the onset time and ei is the offset time of the note in seconds. Their system then generates subsequence’s of the transcribed melody the authors call pitch vectors, with different durations. This process is carried out not only on the transcribed melody, but also on each melody from the corpus. The similarity of melodic fragments is measured using the Euclidean distance between pitch vectors. To obtain a sublinear time complexity, the authors employ locality sensitive hashing . LSH is an algorithm for searching approximately nearest neighbours in high dimension spaces. The principal behind LSH is that points whose distances are within the threshold *r* will be hashed to the same bucket. Each query pitch vector is matched against melodic fragments in the database using LSH. The LSH returns the nearest neighbours and their distances to the query as matches. To obtain the final list of retrieved melodies, the candidate melodies are ranked according to their distance to the entire query note sequence. They report a top-3 hit rate of 90% for 427 queries and a performance increase of between 4 and 20 times compared to exact nearest neighbour search.

* 1. Discussion

Interestingly, annotation systems such as that proposed in this work do not seem to form part of the literature. The work proposed seems to fall between two types of MIR systems. It is similar to the systems outlines in section X.X in the sense that the aim of the work is to annotate a digital recording. However the systems in section X.X work entirely in the signals domain. Their aim is to identify a digital recording as being an instance of another digital recording. These systems create hashes of recordings known as *audio fingerprints* in order to decrease computational complexity and minimise memory usage. In these systems two versions of the same piece of music will be annotated differently. In this work, the aim is to make different versions of the same piece of music be annotated identically. This is particularly important if the work is to facilitate the types of queries suggested in section X.

It seems reasonable to understand the aim of a typical QBH system to be to try and find a melody from a corpus that is similar to a hummed query. To understand why the approaches outlines in section X.X are not appropriate for solving the problem addressed by this work, it is necessary to frame the problem differently. The aim of this work is to annotate a melodic query. This means that it is necessary to identify the melodic query as being an *instance* of a melody from a corpus, so that the query can be annotated accurately. The approach therefore should be to maximize similarity between pieces of music played on different instruments, in different tempos and most importantly in different regional and individual styles and ground truth transcriptions of the musical pieces. In order to achieve this it is necessary to first *normalise* both the query and strings from the corpus, where normalisation involves *removal* of musical style. To do this requires a consideration of which parts of a melody are core, which parts are subject to interpretation and also the nature of the interpretation. This question is explored in section x. Removal of musical features should increase accuracy. This is not the case in gross contour representations of melodies such as that described in section X.X used in many MIR systems, which as the literature suggests results in too many mismatches (Schlichte 1990; Adams, MA Bartsch & Wakefield 2003; Lu, You & Zhang 2001). Instead the melody representation scheme should be fine grained enough to minimise the possibility of mismatches.

There should be no arbitrary limits in this system on the length of a query. Queries might conceivably consist of a melody fragment (a fingerprint?), an entire tune, or multiple tunes played segue in a set. For longer phrases, it makes sense to extract the maximum amount of relevant information from a query to use in matching. Section X addresses the problem of how to normalise traditional music to maximise melodic similarity. It also addresses the problem of how to match melody fragments and entire melodies. Chapter X extends this to address the problem of how to annotate multiple segue melodies played in a set.

* 1. Conclusions

1. Machine Annotation of Traditional Tunes (MATT2)

This chapter proposes MATT2, a system for automatically annotating recordings of traditional Irish dance music. First, using a number of DSP (Digital Signal Processing) algorithms, tunes are transcribed to strings of *normalised* ABC music notation language. Once a transcription is made, the system compares it against a corpus of human made tune transcriptions. The ABC language has the advantage of being based on ASCII text and so tunes in ABC can be easily processed and analysed using algorithms for textual information retrieval. Using this approach, a high success rate for both long and short phrases of music is reported.

* 1. System design

MATT2 works on mono, digital audio files in the WAV format recorded at 44KHz. A high level diagram of the sub systems that make up MATT2 are presented in . The subsystems present in MATT2 will now be described.



Figure : High level diagram of the MATT2 tune annotation system

* + 1. Onset detection

The audio file to be annotated is first segmented into candidate note onsets and offsets using an onset detection function adapted from Gainza . The onset detection function ODCF is based on time domain FIR comb filters. ODCF discovers harmonic characteristics of the input signal and is therefore tolerant to energy changes in an input signal not caused by note onsets and is also better at detecting onsets in legato playing typical of windblown traditional instruments such as the flute and the tin whistle.

The input signal is first segmented into overlapping frames of 2048 samples (approximately 46 milliseconds). Each frame overlaps with the previous frame by 75%. Each frame is then passed through a bank of twelve FIR comb filters. A FIR comb filter works by summing the input signal with a delayed version of the same input signal. The delay of the filter is calculated as being the length in time of a single period of a waveform at the frequency.

This has the effect of amplifying the frequency or a harmonic thereof in the input signal that matches the frequency being filtered. Thus, the energy of the input signal is doubled only if the peaks of the signal coincide with the peaks of the FIR comb filter. This will only occur for a given delay and its integer multiples . Twelve filters E(m, D) with different delays are used corresponding to the twelve semitones in the key of D3 as per Equation 9.

|  |  |
| --- | --- |
|  |  |

Equation

For each frame of audio examined, the outputs of the audio passed through each of the twelve filters are calculated. A value for the ODF dE(m) is then calculated as being the sum of the difference between the outputs of each of the twelve filters in successive frames squared, as described in Equation 10.

|  |  |
| --- | --- |
|  |  |

Equation

In the case where the pitch of the input signal changes from one note to another, this will result in a peak in the ODF graph. Using statistical techniques, a threshold is calculated for each 25ms of audio as being the overall average ODF plus the standard deviation of the ODF in that frame (M. Gainza 2006). Peaks above the threshold are recognised by the system as candidate note onsets. A peak in the ODF is defined as a value preceded by four ascending values and followed by four descending values (though MATT2 supports a configurable value for this). Onsets and offsets are considered by the system to be concurrent as the wooden flute is typically played legato and so a candidate note is considered to be a segment bounded by two adjacent onsets.

Figure 3 shows the signal for the first bar of the tune “The Boyne Hunt” with the detected candidate note onsets marked. The second plot in this figure shows the ODF for the signal, with the dynamic threshold and the candidate onsets marked. In this plot, it is significant that the first note contains a dynamic energy change approximately half way through the note which the ODF has correctly ignored.

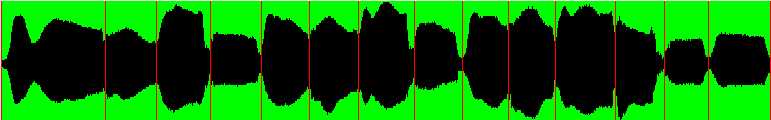
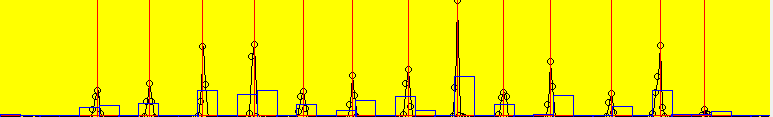
  


Figure : Signal and ODF plots of the first bar of the tune "The Boyne Hunt"

**3.2. Pitch detection**

To establish the perceived pitch of each note, the fundamental frequency (F0) of the note must be derived. The pitch detector subsystem first calculates the highest, nearest power of two nP(m) for the length in samples dS(m) of each segment m of audio bordered by a candidate onset as given in Equation 11.

|  |  |
| --- | --- |
|  |  |

Equation

It then performs a short term FFT (Fast Fourier Transform) on the segment. To determine the perceived pitch the system first calculates peaks in the FFT. In this case, a peak is a value bordered by two ascending/descending values. The algorithm then calculates the interval between the first and second peak. This approach works well for the harmonics of the wooden flute.

* + 1. Breath detection

The flute is a windblown instrument and hence a musician must periodically take breaths as a piece of music is being played (Larson 2003; Hamilton 1990). MATT2 incorporates a breath detector sub system to transcribe a breath in the signal. The breath detector first calculates the average absolute amplitude of the entire signal x(m). It then calculates the average absolute amplitude for each candidate note frame x(n).

A breath is marked if either the perceived pitch detected by the pitch detector is less than 100Hz or the average amplitude of a candidate note n is less than a 10% threshold t of the average amplitude over the entire signal m as in Equation 12. Again, this threshold is configurable. Breaths detected before the transcription of the first pitched note are ignored by the system.

|  |  |
| --- | --- |
|  |  |

Equation

* + 1. Pitch spelling

A wooden flute used to play traditional music has a range of two octaves, though this can be extended by cross fingering techniques (Larson 2003; Hamilton 1990; Vallely n.d.). To tag each candidate note with a pitch spelling pS(n), each calculated note frequency is compared with the frequencies of the notes in the key of D3 Major and D4 Major k1... k16 the two octaves playable on a wooden flute. The nearest match for the frequency f(n) is the assigned the pitch spelling pS(n) (Equation 13).

|  |  |
| --- | --- |
|  |  |

Equation

**3.4. Note quantisation**

At this stage, the system has a set of candidate notes, with corresponding durations and perceived pitches with spellings. To quantise note durations, the quantisation subsystem generates a fuzzy histogram of the intervals between consecutive onsets. As the test matching corpus contains only reels (tunes in 4/4 time) the histogram bin with the highest count is considered to be the length of a quaver note. The fuzzy histogram counts notes within 30% +/- of the the bin width. This 30% is the fuzz referred to in . The algorithm also updates the bin width each time a candidate is added to the bin, so that the bin widths contain the cumulative average lengths of notes counted.

Using the histogram, the duration calculator concludes the length of a quaver qL as being the bin with the highest value. Pseudocode for this algorithm is given in .

Figure 17 shows the duration histogram for a twenty eight second phrase of music from the tune “The Hunters Purse”. Notes whose durations are close to zero have their durations merged with subsequent notes. This has the effect of eliminating consecutive onsets (false positives caused by noisy onsets) and also eliminating ornamentation notes such as those found in rolls, cuts taps and crans typical of traditional Irish music (Larson 2003; Keegan 1992; Hamilton 1990; Vallely n.d.; Tansey 1999).

MATT2 then calculates the maximum bin value in a second fuzzy histogram of the new note durations after consecutive onset and ornamentation elimination. The system uses this value to be the new length of a quaver qL’. The duration calculator then evaluates the nearest multiple qQ of the quaver length qL for each candidate note n as per Equation 14.

Figure : Fuzzy histogram of candidate note lengths from a 28 second phrase from the tune "The Hunters Purse"

|  |  |
| --- | --- |
|  |  |

Equation

In this way notes are quantised as being quavers, crochets, dotted crochets and minims and ornamentation notes are eliminated from the transcription.

*foreach* (note *in* transcribed\_notes)

*begin*

found 🡨 false

*foreach*(bin *in* histogram)

*begin*

bin\_start 🡨 bin.width - fuzz

bin\_end 🡨 bin.width + fuzz

*if* (note.duration >= bin\_start *and* note.duration <= bin\_end)

*begin*

found 🡨 true

bin.count ++

bin.width 🡨 (bin.width +  
 note.duration) / 2

*break*

*end*

*if not* found

*begin*

newNote.count 🡨 1

newNote.width 🡨 note.duration

histogram.add(newNote)

*end*

*end*

quaver\_length 🡨 max(histogram)

Figure : Pseudocode for the fuzzy histogram quaver length calculator

For many of the test recordings used to evaluate MATT2 recorded in imperfect conditions, this approach results in remarkably few transcription errors.

**3.5. ABC Normalisation**

Before edit distance matching against the corpus is carried out, both the transcribed string and strings from the corpus are normalised. This step is necessary as the ABC format supports features such as repeated sections, which need to be expanded so that they can be correctly matched against transcribed phrases. Normalisation of musical strings has the added advantage of minimising the effect of transcription errors on the calculation of the edit distance. Normalisation involves four stages.

Firstly, all whitespace, ornamentation markers and text comments are removed. When ornamentation markers (*~{}*) are removed from ABC transcriptions, this has the effect of quantising the duration of the majority of notes in corpus strings to multiples of the duration of a quaver.

Original:

**d2BG dGBG|~G2Bd efge|d2BG dGBG|1 ABcd edBc:|2 ABcd edBd||**

After Ornamentation removal:

**d2BGdGBG|G2Bdefge|d2BGdGBG|1ABcd edBc:|2ABcdedBd||**

After note expansion:

**ddBGdGBG|GGBdefge|ddBGdGBG|1ABcd edBc:|2ABcdedBd||**

After section expansion:

**ddBGdGBGGGBdefgeddBGdGBGABcdedBc**

**ddBGdGBGGGBdefgeddBGdGBGABcdedBd**

After register normalisation:

**ddBGdGBGGGBdefgeddBGdGBGABcdedBc**

**ddBGdGBGGGBdefgeddBGdGBGABcdedBd**

Figure : Normalisation stages for the A part of the tune “Come West Along the Road”

Secondly, all notes of duration greater than that of a quaver are expanded to be multiple instances of a quaver. This minimises the effect of false negatives in the ODF as *long* notes (false negatives in the ODF) become multiple short notes in both the transcribed phrase and in tunes from the corpus . This also introduces a certain amount of tolerance in the matching subsystem to stylistic variations in the playing of tunes.

Thirdly repeated sections are expanded and bar divisions are removed. ABC supports several notations for different types of repeated phrases (Mansfield 2007). This means for example, that if the transcribed tune was the A part of a tune played twice, this would be correctly matched against the expanded A part of a tune from the corpus.

Finally all notes are transformed to be in the same register. This is achieved by transforming lower case characters in the ABC of tunes to upper case. shows examples of each stage in the ABC normalisation process.

* + 1. Edit distance matching

One final transformation is carried out on strings from the corpus before they are compared with transcribed strings. Occasionally, strings from the corpus are shorter than transcribed strings. For example, the transcribed string might be from a double reel, while the string from the corpus could be from a single reel (a tune half the length). In order to gain the maximal impact from the transcription, corpus strings shorter then transcribed strings are duplicated until their length is greater than the length of the transcribed string. This approximates what a real musician would do in order to extend the duration of a tune (Vallely 1999; Mansfield 2007; Nan Zheng & Bryan Duggan 2007).

The minimum edit distance *eF(c)* for each string *c* from the corpus *Z* then calculated using a cost of one for insertions, deletions and substitutions, for each pair consisting of the transcribed string *s* in substrings of *c*. A variation of the classic edit distance algorithm described in is used to search for the minimum edit distance for a search string in substrings of a target string. This way any phrase from a tune can be matched not just complete tunes and not just incipits. Edit distances are normalised by dividing by the length of the transcribed string to produce *eF(c)*. Two methods for establishing the lowest edit distance were implemented. Firstly, MATT2 returns tunes whose edit distance is less than a configurable threshold. The system also returns the top ten matching tunes in order of lowest edit distance.

* 1. Interface

MATT2 was developed in Java. A screenshot of the system is presented in .

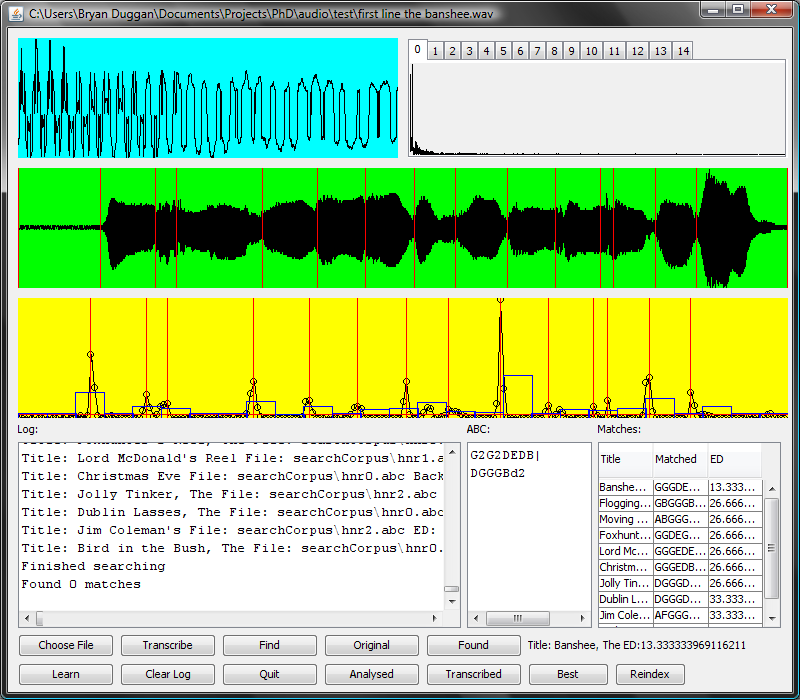


Figure : Screenshot of MATT2

The interface to MATT2 displays several useful plots of the outputs of each stage in the transcription and the matching algorithm such as the current frame being analysed, the onset detection function and the FFT of each detected note. Additionally, the interface displays the transcription in ABC format and the title of the current closest tune match. MATT2 can also play the original WAV file being analysed, the transcribed pitches and durations, the quantised transcription in ABC, the closest match and any of the matched tunes. When the matching algorithm terminates, MATT2 displays and can play any of the top ten closest matching tunes, with their corresponding edit distances. It can also operate in batch mode where it will attempt to annotate all the WAV files in a folder.

* 1. Experiment and results

Precision is used to give an indication of the relevance of the answer

set. It is the number of relevant documents in the answer set, divided

by the total number of documents in the answer set. See Figure 2 for

a graphical representation of precision. In this figure we can observe

two shadings. The darker shaded portion of the figure represents the

documents within the answer set while the lighter shaded portion

represents the documents in the answer set which are relevant.

#of documents in the answer set

#of relevant documents in the answer set

Pr ecision =

**3.2**

Recall gives an indication of how many of the documents returned

are correct. It is defined as the number of documents in the answer

set, divided by the number of documents that are relevant within

the corpus. Figure 3 shows a graphical representation of

recall. The darker shaded portion of the figure represents all the

documents within the corpus which are relevant while the lighter

shaded portion represents the documents in the answer set which are

relevant.

#of relevant documents in the corpus

#of relevant documents in the answer set

Consider a case where an information retrieval system retrieves 12

documents in response to a particular query, and a specialist

has deemed 10 documents within the corpus as relevant. If the

first document in the answer set is one of the relevant ones, then at

that point there is a precision of 1 and a recall of .1 (10% of all

relevant the documents are retrieved). If the second document is

relevant, then the system has a precision value of 1 and .2 recall. If

the third document returned by system is not marked as relevant,

then the system has .66 precision and .2 recall (2 out of 3 retrieved

documents are relevant and 2 out of 10 relevant documents are

retrieved). A typical precision versus recall graph averaged over a

set of queries (50 for example) is shown in Figure 4. Precision and

recall graphs are then generated in order to visualise retrieval

performance as well as providing a method for comparison of

retrieval runs.

The

top-X hit rate reports the proportion of queries for which ri ≤ X

To evaluate MATT2, nine subjects recorded the A and B parts to a number of double and single reels. These recordings were made in imperfect conditions (a kitchen in a house, a school room and a pub) and contain ambient noise such as chairs moving, doors opening and foot taps. The recordings were edited so that the audio being tested contained a mix of complete tunes and segments of tunes taken from the start, middle and end of tunes. Some deliberately challenging audio such as archive recordings, flute duets**,** flute and fiddle duets, and fiddle solos was also included.

For forty three of input audio files, the algorithm correctly identified the tune. A further five tunes were correctly annotated within the top ten closest matches. Just two of the test audio files were incorrectly annotated. Experiments have shown that MATT2 demonstrates a remarkable robustness to tempo variations, musical style, ornamentation, musician variations, instruments and recording environments. Amazingly, one of the tunes tested was recorded over thirty years ago by the flute player Packie Duignan on an analogue tape recorder and the audio had badly degraded. We did not attempt to pre-process this audio before annotating it. For this test, the algorithm did not correctly identify the tune; however the correct tune was the second closest match. The correct tune could have been returned with the addition of a simple heuristic. Further fiddle solos, flute duets and flute and fiddle duet was also included in test audio. In all cases the algorithm correctly identified the tune, despite the fact that the onset detection function used is reported to be inaccurate at detecting onsets in fiddle recordings .

The average normalised edit distance was 39% for the closest matches, while the average of the nearest second match was 53%. The difference between these (14%) can be considered as a confidence level. As an example of the results the system generates, shows the edit distances returned by MATT2 for the tune, “The Golden Keyboard”, played by flute player Eamonn Cotter. This recording was made on a portable MP3 recorder and subsequently transcoded to WAV format.

Figure : Top ten edit distances for a recording of the tune "The Golden Keyboard"

The edit distances shown in the graph are normalised and there is a 14% difference between the correct tune and the nearest closest match. The second and subsequent closest matches have only small variances. These results are summarised in .

|  |  |
| --- | --- |
| Average ED (closest): | 39% |
| Average ED (next): | 53% |
| Difference: | 14% |
| Average ED (closest, correct): | 36% |
| Average ED (next, correct) | 52% |
| Difference: | 16% |
| Average ED (closest, incorrect): | 56% |
| Average ED (next, incorrect): | 57% |
| Difference: | 1% |

Table : Average edit distances for the closest match and the next closest match for tunes correctly and incorrectly annotated

When incorrect matches were considered, it was discovered that in all cases incorrect matches were as a result of transcription errors. Further, the transcription errors were caused by either unusually prominent foot taps or the musician started slow and then speeded up as the tune was played, which affected note duration quantisation.

* 1. Conclusions

In this chapter MATT2, a system for annotating field recordings of traditional Irish music with metadata was described. MATT2 combines a novel transcription system that makes use of ODCF to detect onsets and fuzzy histograms to quantise note durations. MATT2 also makes use of publicly available transcriptions in ABC made by the traditional music community to match transcriptions against. A novel string normalisation technique that takes advantage of the ABC language to eliminate the effect of transcription errors and stylistic variations in input audio was also presented. Further MATT2 improves on pitch contour representations of music strings in MIR systems by using accurate pitch and duration information from input audio in the melody matching subsystem. To our knowledge, MATT2 represents the first attempt to adapt MIR to the specific characteristics of traditional Irish dance music.

Experiments demonstrate that the approach outlined in this paper is robust to variations in musician, style and instrument. In successfully testing MATT2 on badly degraded archive audio, we conclude that the approach presented can be further developed for use on the many thousands of hours of archived recordings of traditional music that currently exist and that are being collected. Our system currently works on segments from tunes or complete tunes. One interesting complication we have not addressed is that tunes are usually played in sets of two or more tunes and each tune is usually played twice. We therefore have the problem that a single audio input file may contain several tunes.

In the cases where the algorithm did not correctly identify the correct tune, we conclude that the transcription subsystem was not able to accurately transcribe the tune. We therefore feel that improving transcription accuracy will lead to more accurate matching and will focus on this problem in future work. Widmer et al. (Widmer et al. 2005) state that transcription algorithms need the kind of higher level musical knowledge that humans poses and we hope to try and develop this approach, possibly using a preference rule style approach similar to the work of Temperly . Support for the transcription of tunes in different time signatures is planned as is experimentation with corpora of tunes in other popular time signatures such as jigs and hornpipes.

1. Machine Annotation of Traditional Sets (MATS)

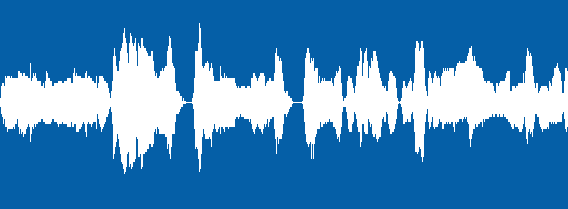
The aim of this chapter is to present to present a new algorithm for annotating sets of traditional Irish dance tunes. The work described in Chapter 6 solves the problem of annotating single tunes, however in traditional music tunes are rarely played singly. More commonly tunes are played in groups of at least two tunes known as a *set* of tunes. A set typically consists of two three or four tunes played in succession without an interval (Vallely 1999; Duggan, B. O'Shea & P. Cunningham 2008). Typically each tune in the set is played twice or three times before musicians advance to the subsequent tune in the set. A repetition or a change from one tune to the next in a set is known as a *turn*. As tunes in sets are always in the same time signature and often in the same key, the challenge therefore is in segmenting sets into tunes and repetitions. The approach presented in this chapter tackles this problem by making use of melodic similarity calculated using a variant of the *edit distance* string matching algorithm described in section 4.6. The MATS algorithm described in this chapter can identify the start and end of each repetition of a tune, can count the repetitions and can identify the title and associated metadata associated with each tune in a set.

This chapter also includes experimental results which establish the effectiveness of this new algorithm.

* 1. Sets of traditional Irish dance tunes

Traditional Irish dance tunes are are typically played as *sets*. Certain common sets were originally put together to accompany set dances (Vallely 1999), while other sets have become popular as a result of being recorded by emigrant Irish musicians in America in the early part of the twentieth century.

turn



Time

Figure : Waveform of the last phrase from the tune "Jim Coleman’s" and the first phrase from the tune "George Whites Favourite" played in a set

The origin of many sets of tunes is unknown and musicians often compile new sets “on the fly” in traditional music sessions. Figure 21 shows a waveform plot from two tunes played in a set. The tunes were played on a wooden flute and as can be seen in the plot, there is no interval between the end of the first tune and the start of the second tune. Maddage *et al.* and other segmentation approaches generally look for repetitive patters in a music recording . This is not the case in our approach, where each tune in the set can be played once or many times.

* 1. MACHINE annotation of TRADITIONAL sets algorithm (MATS)

In this section MATS is described. MATS is an enhancement to MATT2 described in the previous chapter. The purpose of MATS is to annotate tunes played in sets. The shortest tune in the corpus *Z* used by MATT2 is a single jig. A single jig *sj* is a tune in 6/8 time with an A and B part played singly (48 quaver notes in duration). The shortest possible set therefore would contain two single jigs (96 notes) played with no repetitions. To annotate a set of tunes, MATS first uses a heuristic to determine if the string of transcribed notes *t* is longer than the length of the shortest set *length(sj)×2*.

When this is the case, the MATS algorithm is used instead of the minimum edit distance algorithm described in. Pseudocode for the MATS algorithm is presented in Figure 22.

MATS first extracts a substring *ss* from *t* the transcription such that *length(ss) = length(sj)* at position *p=1* in *t*. MATS then searches the corpus *Z* using the edit distance algorithm described in section 3 to find a the closest match for *ss*. When a match is found MATS knows the name of the first tune and has *c'*, a transcription of the tune played with no repetitions from the corpus Z. MATS then generates an edit distance profile *edp* for *c',* the matching tune, in *t* the transcription*.* *edp* is given as the last row of the edit distance matrix and can be understood as the positions where substrings in *t* match *c'* withthe minimum edit distance *.*

Figure 23shows the edit distance profiles for the set of tunes “Jim Coleman’s”, “George Whites Favourite” and “the Virginia” played in a set. The algorithm has identified the first tune as “Jim Coleman’s” and has subsequently generated an edit distance profile (the top plot in Figure 23) for the first tune in the transcription. The two troughs in this graph indicate the end of the two repetitions of the tune in the transcription. These can be considered as turns in the set.

The MATS algorithm then normalises the edit distance profile *edp* and passes the graph through a low pass filter that filters frequencies less than 10Hz. This has the effect of smoothing the graph. An example of a smoothed edit distance profile is given in Figure 24. This graph illustrates the top graph in Figure 23 after filtering has been applied.

The algorithm then detects troughs in the graph less than a threshold initially set to *t=0.3*. The algorithm varies this threshold dynamically by trying different values until the number of troughs in the graph is between one and five. It is rare in traditional music for a tune to be repeated more than five times in a set.

p 🡨 0

rem 🡨 length(t) - p

*while* (rem >= sj)

*begin*

ss 🡨 substring(t, p, p + sj)

*foreach* (c *in* Z)

*begin*

ed\_c 🡨 min(ed(ss, c))

*if* (ed\_c < min\_ed)

*begin*

min\_ed 🡨 ed\_c

c' 🡨 c

*end*

*end*

edp 🡨 ed(c', t)

edp 🡨 normalise(edp)

edp 🡨 filter(edp, 10)

th 🡨 0.3

v 🡨 troughs(edp, th)

*foreach* (tr *in* v)

*begin*

convertToTime(tr)

*end*

r 🡨 length(v)

p 🡨 v[r]

print c’, r

rem 🡨 length(t) - p

*end*

Figure : Pseudocode for the MATS set annotation algorithm

The trough detection algorithm in MATS returns a vector of troughs , such that *length()* is the number of troughs and the elements in are the positions of the bottom of the troughs. A trough in MATS need only have a descending wall as a trough can occur at the end of a tune and hence may not contain an ascending wall. An example of this is the third plot in Figure 23.

The algorithm repeats this process with a new *p* given as the last entry in the troughs vector to extract the second and subsequent tunes in the set until it is no longer possible to extract a substring *ss* of length *length(sj)* starting at *p* because we have reached the end of *t*. The second tune in the set, “George Whites Favourite” was played once and there is a corresponding single trough in the graph of the edit distance function (the middle plot in Figure 23) for the tune from the corpus *c'* in the transcription *t*. The third tune “the Virginia” was repeated twice and so there are two troughs in the bottom plot in Figure 23.

* 1. Experiment and results

In order to test the robustness of MATS we had a traditional musician record ten audio files of flute tunes played in sets. The recorded files are available at http://www.comp.dit.ie/bduggan/mats. The sets played in the input audio were taken from the Foinn Seisiún series of books published by Comhaltas Ceoltóirí Éireann (Brian 2004).



Figure : Edit distance profiles for three tunes played in a set

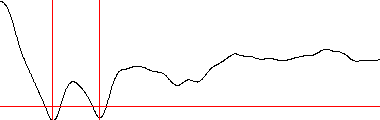




Figure : Filtered version of first graph in Figure 4. The dynamic threshold and detected troughs are marked

The sets consisted of single and double jigs and reels played multiple times in sets. In total, the sets contained 23 separate tunes with 48 turns we were interested in annotating. In carrying out this experiment, we were interested in establishing if MATT2 could correctly figure out the timings of turns and could identify the names of the tunes.

|  |  |
| --- | --- |
| **Correctly identified** | 96% |
| **Incorrectly identified** | 4% |

Table : Correctly and incorrectly identified tunes

MATT2 successfully identified 22 out of the 23 tunes, and recognised each input audio file as a set and so used the MATS set annotation algorithm (Table 5).

Table 6 shows a sample of the data collected in this experiment for the audio file used to generate Figure 23 and Figure 24. To establish a ground truth for the experiment, a human domain expert manually annotated the turns in the sets of tunes. In the human and machine columns are listed the onset time for turns in the set. Onset times for changes from one tune to the next are highlighted. From this table it can be seen that on average MATS was within .85 seconds of the human annotations.

|  |  |  |  |
| --- | --- | --- | --- |
| **Tune** | **Human** | **Machine** | **Difference** |
| **1** | 20.68 | 21.10 | 0.43 |
| **1** | 41.42 | 41.9 | 0.48 |
| **2** | 82.72 | 83.15 | 0.43 |
| **3** | 123.88 | 124.44 | 0.56 |
| **3** | 164.49 | 166.85 | 2.36 |
| **Average** |  |  | 0.85 |

Table : Human & machine annotated turns

The overall annotation accuracy is obtained by calculating two different measures *precision* and *recall.* The value of *precision* is calculated as per (2) where *TP* and *FP* are the true positives (correctly identified turns) and false positives (incorrectly identified turns). *recall* is calculated as per equation (3) where *FN* is the number of false negatives (turns in the input signal not detected by the algorithm).

|  |  |
| --- | --- |
|  | (2) |
|  | (3) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **TP** | **FN** | **FP** | **precision(%)** | **recall(%)** |
| 39 | 9 | 6 | 87% | 81% |

Table : Annotation accuracy

Table 7 shows the annotation accuracy. In can be seen from *precision* and *recall* that the algorithm provides a high degree of accuracy at detecting turns. Because the algorithm can successfully identify turns, it can also correctly extract a suitable prefix from the subsequent tune in the set and so can identify the tune. *FN*’s were caused by the algorithm failing to correctly identify the transitions between tunes in a set. When this happens the algorithm cannot extract a representative prefix from the next tune and so all subsequent turns are usually misidentified. In some cases, *FP*’s were within a few seconds of the two second threshold we had set.

* 1. Conclusions

This chapter presented a novel algorithm that addresses a problem in the domain of Irish traditional dance music, that of annotating sets of tunes. As a set can contain an arbitrary number of tunes played segue without an interval and as tunes in sets are repeated an arbitrary number of times, are always in the same time signature and often in the same key, the significant challenge in this problem is in recognising where one tune ends and the next tune starts. The results presented prove that MATS is effective at segmenting sets, counting repetitions and at annotating individual tunes played in a set. To our knowledge this is the first time this problem has been addressed in an MIR system and we suggest that the proposed approach can be adapted to segmenting repeated tunes from other genres played in a segue.

The corpus used currently contains reels and jigs and in future work it will be augmented with the full complement of traditional tunes in different time signatures. One interesting feature not yet exploited is the metadata typically present in an ABC transcription. Effectively the time and key signature of an input audio file can be determined by *melodic similarity* with a known tune. This can be exploited in several interesting ways. Firstly, if the first tune in a set were to be identified as a reel, the search for subsequent tunes can be limited to reels, thus speeding up annotation. Conversely, if a number of reels were to be identified in a set and a single tune in a different time signature was to be identified this could be recognised as a potential error.

1. Conclusions & Future Work
   1. Conclusions
   2. Future work

Appendix A – Publications

The following are the list of peer reviewed publications:

B. Duggan, B. O'Shea, Mikel Gainza and P. Cunningham, "Machine Annotation of Sets of Traditional Irish Dance Tunes", Ninth International Conference on Music Information Retrieval (ISMIR), Drexel University, Philadelphia, USA, September 2008.

B. Duggan, B. O'Shea, and P. Cunningham, “A System for Automatically Annotating Traditional Irish Music Field Recordings,”, Sixth International Workshop on Content-Based Multimedia Indexing, Queen Mary University of London, UK, Jun. 2008

Duggan, B: TunePal: A Portable Tune Teaching Tool for Traditional Musicians, DIT Annual Showcase of Learning & Teaching Activities, January, 2007

Duggan, B., Zheng, C., Cunningham, P.: MATT - A System for Modelling Creativity in Traditional Irish Flute Playing, Third Joint Workshop on Computational Creativity, ECAI'06, Italy, August 2006

Duggan, B.: Learning Traditional Irish Music using a PDA, IADIS Mobile Learning Conference, Trinity College, Dublin, Ireland, July 2006

References

M-AUDIO - MicroTrack II - Professional 2-Channel Mobile Digital Recorder. Available at: http://www.m-audio.com/products/en\_us/MicroTrackII-main.html [Accessed January 23, 2008].

The Humdrum Toolkit: Software for Music Research. Available at: http://www.musiccog.ohio-state.edu/Humdrum/ [Accessed July 16, 2008].

Adams, N., Bartsch, M. & Wakefield, G., 2003. Coding of sung queries for music information retrieval. *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, 139-142.

Andoni, A. & Indyk, P., 2006. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. *Proceedings of the Symposium on Foundations of Computer Science*.

Baeza-Yates, R. & Perleberg, C., 1996. Fast and practical approximate pattern matching. *Information Processing Letters*, 59(1), 21-27.

Barlow, H. & Morgenstern, S., 1948. *A Dictionary of Musical Themes*, Reprint Services Corp.

Birmingham, W. et al., Musart: Music Retrieval Via Aural Queries. *Ann Arbor*, 1001, 48109-2110.

Breathnach, B., 1963. Ceol Rince na hÉireann Cuid I [Dance Music of Ireland Vol I - Vol V].

Brian, P., 2004. *Foinn Seisiún* , Comhaltas Ceoltóirí Éireann.

Bryan Duggan, 2006. Learning Traditional Irish Music using a PDA. *IADIS Mobile Learning Conference, Trinity College Dublin*.

Bunting, E., 1843. *The Ancient Music of Ireland: Arranged for Piano*,

Chambers, J., 2007. O' Neills Books. Available at: http://trillian.mit.edu/~jc/music/book/oneills/.

Christine Kenneally, 2008. So you think humans are unique? *New Scientist*.

Doraisamy, S., Adnan, H. & Norowi, N., Towards a MIR System for Malaysian Music. *7th International Conference on Music Information Retrieval, Victoria, Canada, 8 - 12 October 2006*.

Downie, J., 1999. EVALUATING A SIMPLE APPROACH TO MUSIC INFORMATION RETRIEVAL: CONCEIVING MELODIC N-GRAMS AS TEXT.

Downie, J., 2003. Music information retrieval. *Annual Review of Information Science and Technology*, 37(1), 295-340.

Dubnowski, J., Schafer, R. & Rabiner, L., 1976. Real-time digital hardware pitch detector. *Acoustics, Speech, and Signal Processing [see also IEEE Transactions on Signal Processing], IEEE Transactions on*, 24(1), 2-8.

Duggan, B., 2007. Enabling Access to Irish Traditional Music Archives on a PDA. *Eight Annual Irish Educational Technology Users Conference, DIT Bolton St, Ireland*.

Duggan, B., O'Shea, B. & Cunningham, P., 2008. A System for Automatically Annotating Traditional Irish Music Field Recordings. *Sixth International Workshop on Content-Based Multimedia Indexing, Queen Mary University of London, UK*.

Eppstein, D. et al., 1992a. Sparse dynamic programming I: linear cost functions. *Journal of the ACM (JACM)*, 39(3), 519-545.

Eppstein, D. et al., 1992b. Sparse dynamic programming II: convex and concave cost functions. *Journal of the ACM (JACM)*, 39(3), 546-567.

Gainza, M. & Coyle, E., 2007. Automating Ornamentation Transcription. *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, 1.

Gainza, M., 2006. *Music Transcription within Irish Traditional Music*, PhD Thesis, Dublin Institute of Technology, Faculty of Engineering.

Ghias, A. et al., 1995. Query by humming - musical information retrieval in an audio database. *In ACM Multimedia 95 Proceedings, San Francisco*.

Grachten, M., Arcos, J. & Lopez de Mantaras, R., 2005. Melody Retrieval using the Implication/Realization Model. *MIREX http://www. music-ir. org/evaluation/mirex-results/article/s/similarity/grachten. pdf*.

Hamilton, C., 1990. *The Irish Flute Players Handbook*, Cork: Breac Publications.

Hitchcock, F., 1941. The distribution of a product from several sources to numerous localities. *J. Math. Phys. Mass. Inst. Tech*, 20, 224-230.

Hoos, H., 2001. GUIDO/MIR—an Experimental Musical Information Retrieval System based on GUIDO Music Notation. *Symposium on Music Information Retrieval: ISMIR*, 41-50.

Howard, J., 1997. Plaine and Easie Code: a code for music bibliography. *Beyond MIDI: the handbook of musical codes table of contents*, 362-372.

Humdrum, 2008. *The Humdrum Toolkit: Software for Music Research*,

Jensen, K., Xu, J. & Zachariasen, M., Rhythm-based segmentation of popular chinese music. *Proceedings of 6th International Conference on Music Information Retrieval (ISMIR’05)*, 374-380.

Joyce, W., 1909. *Old Irish Folk Music and Song*,

Kassler, M., 1966. Toward musical information retrieval. *Perspectives of New Music*, 4(2), 59-67.

Keegan, N., 1992. *The Words of Traditional Flute Style*, MPhil Thesis, University College Cork, Music Department.

Kornstadt, A., 1998. Themefinder: A web-based melodic search tool. *Computing in Musicology*, 11, 231-236.

Krassen, M., 1975. *O' Neil's Music of Ireland*, Waltons.

Larson, G., 2003. *The Essential Guide to Irish Flute and Tin Whistle*, Mel Bay Publications, Inc.

Lemstrom, K. et al., 2003. The C-BRAHMS Project. *Proceedings of the 4th Internationoal Conference on Music Information Retrieval (ISMIR 2003)*, 237-238.

Lemstrom, K. & Perttu, S., 2000. SEMEX-An Efficient Music Retrieval Prototype. *First International Symposium on Music Information Retrieval (ISMIR)*.

Lemstrom, K. & Ukkonen, E., 2000. Including interval encoding into edit distance based music comparison and retrieval. *Proceedings of the AISB’2000 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science’, Birmingham*, 53-60.

Levenshtein, V., 1966. Binary Codes Capable of Correcting Deletions, Insertions and Reversals. *Soviet Physics Doklady*, 10, 707.

Lu, L., You, H. & Zhang, H., 2001. A new approach to query by humming in music retrieval. *Proceedings of the IEEE International Conference on Multimedia and Expo*.

Maddage, N. et al., 2004. Content-based music structure analysis with applications to music semantics understanding. *Proceedings of the 12th annual ACM international conference on Multimedia*, 112-119.

Makinen, V., Navarro, G. & Ukkonen, E., 2003. Algorithms for transposition invariant string matching. *Proc. STACS*, 191-202.

Mansfield, S., 2007. How to Interpret ABC Notation. Available at: http://www.lesession.co.uk/abc/abc\_notation.htm.

McNab, R. et al., 1997. The New Zealand Digital Library MELody inDEX. *D-Lib Magazine*, 3(5), 4-15.

McNab, R. et al., 1996. Towards the digital music library: tune retrieval from acoustic input. *Proceedings of the first ACM international conference on Digital libraries*, 11-18.

McPherson, J. & Bainbridge, D., 2001. Usage of the MELDEX Digital Music Library. *Proceedings of the Second Annual International Symposium on Music Information Retrieval (Bloomington, IN, USA*, 15-17.

Mongeau, M. & Sankoff, D., 1990. Comparison of musical sequences. *Computers and the Humanities*, 24(3), 161-175.

Nan Zheng & Bryan Duggan, 2007. A Combinational Creativity Approach to Composing Traditional Irish Reels. *18th Irish Conference on Artificial Intelligence and Cognitive Science, Dublin Institute of Technology, Ireland*.

Navarro, G. & Raffinot, M., 2002. *Flexible Pattern Matching in Strings: Practical On-Line Search Algorithms for Texts and Biological Sequences*, Cambridge University Press.

Nesbit, A., Hollenberg, L. & Senyard, A., Towards Automatic Transcription of Australian Aboriginal Music. *5th International Conference on Musical Information Retrieval, Barcelona, Spain October 10-14, 2004*.

Norbeck, H., 2007. ABC Tunes. Available at: http://www.norbeck.nu/abc/index.html,

O'Neill, F., 1903. *The Music of Ireland*,

O'Shea, H., 2006. Getting to the Heart of the Music: Idealizing Musical Community and Irish Traditional Music Sessions. *Journal of the Society for Musicology in Ireland*, 2(7), 1.

Parsons, D., 1975. *The directory of tunes and musical themes*, New York: Spencer Brown.

Petrie, G., 1855. *The Petrie Collection of the Ancient Music of Ireland*,

Prechelt, L. & Typke, R., 2001. An interface for melody input. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 8(2), 133-149.

Rho, S. & Hwang, E., 2004. FMF (Fast Melody Finder): AWeb-Based Music Retrieval System. *Computer Music Modeling and Retrieval: International Symposium, CMMR 2003, Montpellier, France, May 26-27, 2003: Revised Papers*.

Rubner, Y., Tomasi, C. & Guibas, L., 2000. The Earth Mover's Distance as a Metric for Image Retrieval. *International Journal of Computer Vision*, 40(2), 99-121.

Ryynanen, M. & Klapuri, A., 2008. Query by humming of midi and audio using locality sensitive hashing. *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, 2249-2252.

Ryynanen, M. & Klapuri, A., 2006. Transcription of the Singing Melody in Polyphonic Music. *ISMIR 2006*, 1.

Schlichte, J., 1990. Der automatische Vergleich von 83 243 Musikincipits aus der RISM-Datenbank: Ergebnisse-Nutzen-Perspektiven. *Fontes artis musicae*, 37, 35-46.

Sødring, T. & Smeaton, A., 2003. Evaluating a Music Information Retrieval System-TREC Style. *“The MIR/MDL Evaluation Project White Paper Collection” Edition# 3*.

Stephen W Smith, 1997. *The Scientist and Engineer's Guide to Digital Signal Processing* 1st ed., California Technical Pub.

Tansey, S., 1999. *The Bardic Apostles of Innisfree*, Tanbar Publications.

Temperley, D., 2004. *The Cognition Of Basic Musical Structures*, MIT Press.

thesession.org, 2007. The session.org Forums. Available at: http://www.thesession.org.

Tony Kearns & Barry Taylor, *A Touchstone for the Tradition - The Willie Clancy Summer School*,

Typke, R., 2007. Music Retrieval Based on Melodic Similarity. *Doctoral thesis, Utrecht University* .

Typke, R. et al., 2003. Using transportation distances for measuring melodic similarity. *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)*, 107-114.

Typke, R., Veltkamp, R. & Wiering, F., 2004. Searching notated polyphonic music using transportation distances. *Proceedings of the 12th annual ACM international conference on Multimedia*, 128-135.

Typke, R., Wiering, F. & Veltkamp, R., 2005. A Survey of Music Information Retrieval Systems. *Proceedings of the International Conference on Music Information Retrieval*, 153-160.

Ukkonen, E., Lemström, K. & Mäkinen, V., 2003. Geometric Algorithms for Transposition Invariant Content-Based Music Retrieval. *ISMIR 2003*, 2, 3.

Vallely, F., 1999. *The Companion to Irish Traditional Music*, New York University Press.

Vallely, F., *Timbre: The Wooden Flute Tutor*,

Wallis, G. & Wilson, S., 2001. *The Rough Guide to Irish Music*, London: Rough Guides.

Walshaw, C., 2007. The ABC home page. Available at: http://www.walshaw.plus.com/abc/.

Widmer, G. et al., 2005. From Sound to `Sense' via Feature Extraction and Machine Learning: Deriving High-Level Descriptors for Characterising Music. *Sound to Sense:Sense to Sound: A State-of-the-Art*.

Wiggins, G., Lemstrom, K. & Meredith, D., 2002. SIA (M) ESE: An algorithm for transposition invariant, polyphonic content-based music retrieval. *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, 283-284.

Stuff to include:

“Cranfield Model” of IR evaluation

principle of parsimony

For systems which return a ranked list, the normalized

precision (NPREC) and normalized recall metrics

capture how closely a ranking system performs relative

to the ideal by including in their calculation information

about the ranks at which relevant documents are listed.

Normalized precision is defined as:

*NPREC*

*Rank*

*m*

*N n REL REL*

*m*

*m*

*REL*







1 1

log

log !/ ( )! !

where N is the number of documents in the database,

REL the number of relevant documents contained in the

database, and RANKm the rank assigned to relevant

document *m.*(see Salton and McGill [21]). Tonta [25]

cites a growing body of literature that contends that users

of IR systems are much more satisfied with search

sessions that have strong precision rather than strong

recall. Thus, our forthcoming analyses, discussions,

conclusions and recommendations, will be based solely

upon the NPREC data.

**Queries and sensitivity evaluation (QQUAL)**