

國立交通大學

工學院聲音與音樂創意科技碩士

學位學程

碩 士 論 文

數字簡譜系統與相關樂曲自動分析研究

**The Study of the Numbered Musical Notation System
and Automatic Musical Analysis**

研 究 生：周怡瑩

指 導 教 授：黃志方 教授

成維華 教授

中 華 民 國 九 十 九 年 七 月

數字簡譜系統與相關樂曲自動分析研究

The Study of the Numbered Musical Notation System and
Automatic Musical Analysis

研究生：周怡瑩

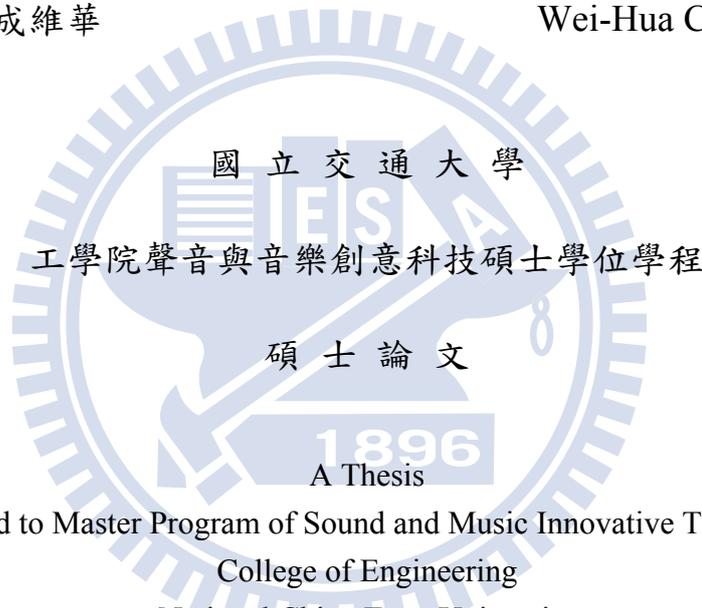
Student : Yi-Ying Chou

指導教授：黃志方

Advisor : Chih- Fang Huang

成維華

Wei-Hua Chieng



Submitted to Master Program of Sound and Music Innovative Technologies
College of Engineering
National Chiao Tung University
in partial Fulfillment of the Requirements
for the Degree of Master
in

Engineering

July 2010

Hsinchu, Taiwan, Republic of China

中華民國九十九年七月

數字簡譜系統與相關樂曲自動分析研究

學生：周怡瑩

指導教授：黃志方
成維華

國立交通大學工學院聲音與音樂創意科技碩士學位學程

摘要

本論文研製之自動分析 MIDI 檔案產生數字簡譜系統，這個系統可將 MIDI 音訊檔轉譯為樂譜，並在 MIDI 資訊可能不完整的情況下(e.g.演奏即時錄製的 MIDI 檔案等)分析出正確的音符事件與整首樂曲的樂理性質，並且把音符訊息以相對簡單的方式呈現，使不太懂音樂理論的人也可以快速理解樂譜的表現。針對 MIDI 資訊記載不完整的檔案，透過多種分析與追蹤技術的運用，整合出一個簡單易懂的音樂呈現機制，不論是使用錄音介面所產生的 MIDI 檔案或從 wav 檔案轉譯的 MIDI 檔案，甚至是樂器演奏等資訊有所誤差的檔案能夠得到最佳化過的結果。

本系統整合了以下分析追蹤技術與規則理論：

1. 音名拼寫，此分析採用可即時拼寫音名的研究技術。
2. 調性分析，也是利用螺旋陣列研究取得最可能的調性。
3. 拍子追蹤與節奏的歸納，從權重周期的分析再歸納節拍。
4. 和絃的判斷。
5. 考慮音樂的通用性，我們加入了音樂理論的規則例如：數字音符標示法則、拍子的基本架構、調性轉移觀念、流行樂的調性標示法則和流行樂的和弦標示法則。

關鍵字：數字簡譜，MIDI，音名拼寫，調性分析，節奏歸納，和絃分析。

The Study of the Numbered Musical Notation System and Automatic Musical Analysis

student : Yi-Ying Chou

Advisors : Dr.Chih-Fang Huang
Dr. Wei-Hua Chieng

Master Program of Sound and Music Innovative Technologies
National Chiao Tung University

Abstract

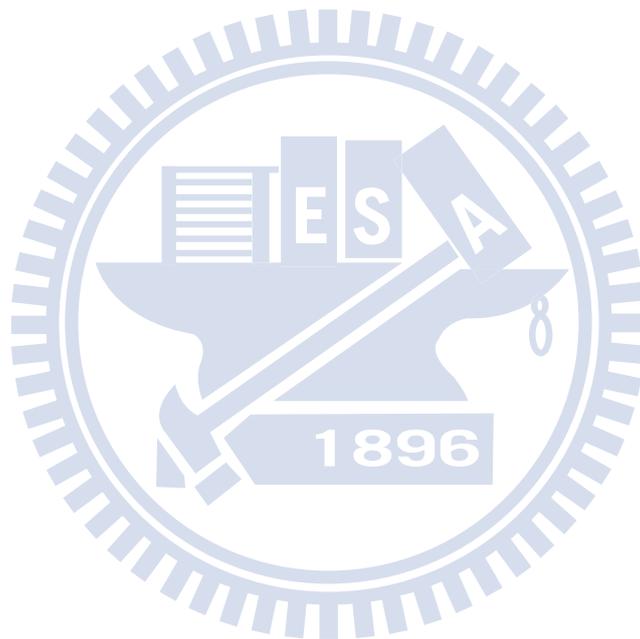


This study provides a number musical notation system which can be used to convert MIDI files to number musical notation. Writing music in number musical notation can be easily stored and transported electronically, which enables the understanding of music for with less music knowledge. The following analysis and tracking technologies and rule theories were integrated in number musical notation system:

1. Pitch spelling: An in-situ technology of pitch spelling was used.
2. Key finding: A spiral array model also was used to optimize the tonality of music.
3. Beats tracking and rhythm induction: This study adopted a process from the analysis of beats weights and duration to rhythm induction.
4. Chord identification: The center of effect in spiral array model was utilized to identify the chord of music.
5. Rule theories: Some rule theories were also integrated in this system, such as number musical notation rule, beat basic framework, tonality transfer, tonality and chord marking rule for pop music.

Based on the integration of pitch spelling, tonality analysis, beats tracking and chord identification, number musical notation system can analyze the MIDI files without complete information, such as the files recorded from audio recorders or converted from WAV files, to find correct note events and the characteristics of music in MIDI files, hence get the optimized results.

Key words: numbered musical notation, MIDI, pitch spelling, key finding, rhythm induction, chord identification



誌謝

想當初才剛為剛考上研究所而慶祝，一轉眼兩年碩士即將進入尾聲，特別感謝黃志方老師、呂向斌學長以及李宏儒學長在這兩年來的指導與提攜。另外，也謝謝實驗室的同儕鶴雄、毅凡、人慈、尹均、阿智的扶持協助以及給予的無限歡笑，陪伴一年多學弟妹科科方、朱朱方、旻璟、粘粘、志翰和明明就是同學的志豪，你們充沛的活力真是十足的感染”成熟穩重”的碩二學長姐呢！還有要感謝清大化工永遠的3 2 5，那群陪吃陪喝陪聊的好朋友，還有那舒適的研究室提供我一年認真的好去處。最後感謝我的家人的支持以及感謝豬博士的陪伴與扶持，你的心靈雞湯無與倫比！最最後再感謝兩隻兔白鼻，天真可愛偶而頑皮的你們，常常讓我忘卻憂慮。



目錄

摘要.....	i
Abstract	ii
誌謝.....	iv
目錄.....	v
圖目錄.....	vii
表目錄.....	ix
第一章 概論	1
1.1 研究動機.....	1
1.2 研究方法與目標	1
第二章 相關研究	4
2.1 簡譜與其他的簡譜系統.....	4
2.1.1 數字簡譜元素介紹.....	5
2.1.2 其他相關簡譜軟體.....	7
2.2 MIDI轉譯樂譜的相關研究與應用.....	8
2.3 音名拼寫的相關研究	9
2.4 調性尋找的相關研究	11
2.5 小節與拍追蹤的相關研究及應用	11
2.6 和弦分析的相關研究.....	12
第三章 音名拼寫與調性分析	13
3.1 螺旋陣列模型.....	13
3.2 影響中心CE之公式制訂.....	15
3.3 調性分析.....	18
3.4 音名拼寫.....	19
3.4.1 音符中心的計算.....	19
3.4.2 二步驟的疊代法.....	22
3.5 系統中音名拼寫與調性分析整合模組.....	24

第四章 節拍歸納與和弦分析	28
4.1 節拍歸納	28
4.1.1 拍子優先權的給定	28
4.1.2 系統中節拍歸納模組	30
4.2 和弦分析	31
4.2.1 和弦CE公式	32
4.2.2 和弦分析模組	33
第五章 系統架構與功能	34
5.1 系統整合系統架構	34
5.2 模組與機制設計	35
第六章 實作結果	42
6.1 使用者介面	42
6.2 比較其他數字簡譜系統	45
第七章 結論與未來工作	47
參考資料	48
附錄一	50
附錄二	51

圖目錄

圖 2.1 其他國外abc language簡譜軟體介紹	8
圖 2.2 Music Information Content Extraction and Analysis	9
圖 3.1 螺旋陣列	13
圖 3.2 螺旋上音到音之間的距離	15
圖 3.3 大、小和絃計算中心意義	16
圖 3.4 大、小調計算中心意義	17
圖 3.5 尋找調性流程	19
圖 3.6 使用螺旋陣列決定音名	20
圖 3.7 拼寫音名流程	22
圖 3.8 滑動視窗方法($w=4$)	23
圖 3.9 兩步驟指派方法($w=4$, $w_r=2$)	24
圖 3.10 音名拼寫與調性分析流程圖	26
圖 3.11 前置作業流程	27
圖 4.1 給定優先權後的音樂結構	29
圖 4.2 節拍歸納的模組流程	31
圖 4.3 和弦分析模組與流程	33
圖 5.1 數字簡譜系統流程圖	35
圖 5.2 系統中主要分析模組的流程圖	37
圖 5.3 系統流程與人機互動	37
圖 5.4 音符物件的建立流程	38
圖 5.5 調性轉移機制的流程圖	39

圖 5.6 定義音符輸出格式流程	40
圖 5.7 音樂架構規則流程圖	41
圖 6.1 系統介面	42
圖 6.2 系統Demo”搖籃曲”樂譜畫面	43
圖 6.3 系統Demo” Come On Over”單一音軌樂譜畫面	44
圖 6.4 系統Demo” Come On Over”多音軌樂譜畫面	44
圖 6.5 CuteMIDI 簡譜作曲家Demo” Come On Over”多音軌樂譜畫面	45



表目錄

表 2.1 Pitch class和簡譜數字的對應表	5
表 3.1 五度線上 k 值對應的音名	14
表 3.2 音名拼寫的對照表	21
表 3.3 音名拼寫的實驗數據	25
表 5.1 系統建立的MIDI通用格式	38



第一章 概論

1.1 研究動機

數字簡譜的使用常見於華人地區，雖為一般非正統之音樂所採用，但在流行音樂上的廣泛使用提供更多更容易學習音樂。因為數字簡譜的簡便性，為很多記譜法不甚統一或過於複雜的國樂譜所採用，對於很多沒有學過樂理的人更是較為簡單且接受度高的。目前樂譜軟體以五線譜的發展最為完善，市售商業軟體相當多，而以數字簡譜為主的打譜軟體較少，尤其可以由系統自動分析樂曲資訊的更少，因此我們希望建立一個自動產生數字簡譜的平台，透過此平台可以轉換音訊檔案為樂譜輸出，未來將可以繼續建構更多的應用程式在平台上，例如自動作曲的簡譜輸出或歌唱創作的即時輸出等。

1.2 研究方法與目標

本研究主要目的是希望提供一套以數字簡譜呈現的系統，這個系統可將 MIDI 音訊檔轉譯為樂譜，並在 MIDI 資訊可能不完整的情況下(e.g. 演奏即時錄製的 MIDI 檔案等)分析出正確的音符事件與整首樂曲的樂理性質，並且把音符訊息以相對簡單的方式呈現，使得就算不太懂複雜音樂理論的人也可以快速且容易的理解樂譜的表現。

分析這個目的，本系統應具有以下目標：

- a. 在數字簡譜系統中需要兼具功能的實用性與操作的便利性。
- b. 讓使用者不必擔心樂理上調的複雜性，提供絕對音高與相對音高的兩種數字記譜法讓一般使用者選擇，可以簡單的閱讀並加強了解調性關係。

上述目標為本研究主要的研究方向，而目前的樂譜系統並沒有達到兼具上述的兩個目標。即使是市面上完整的譜曲軟體，包括五線譜的打譜軟體也較少將音符以相對音高表現，通常是透過調整轉移才能得到較易懂的譜號，這對於一般沒有樂理概念的使用者來說必須花更多時間學習或閱讀樂譜，很有可能會降低了音樂學習意願甚至是抹煞了音樂創造的可能性。因此，我們希望使用者可以減少這些障礙，進而提升音樂的學習樂趣與豐富性。

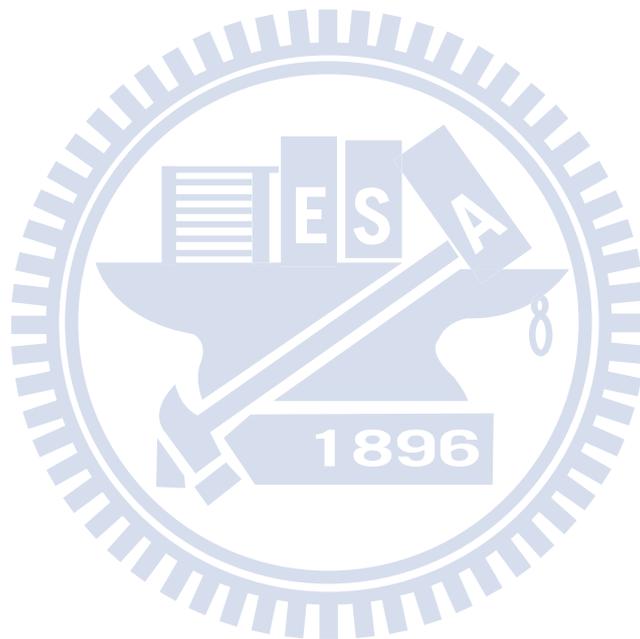
然而，若系統上所呈現的樂譜只依靠MIDI檔案上所記錄的事件，這對於MIDI資訊記載不完整的檔案，將會使樂譜輸出產生不可預期的錯誤。而我們所希望的是透過多種分析與追蹤技術的運用，整合創造出一個簡單易懂的音樂呈現機制，讓使用者只要透過系統就可輕鬆取得音符資訊，不論是使用錄音介面所產生的MIDI檔案或從wav檔案轉譯的MIDI檔案甚至是樂器演奏等等各種可能產生資訊不完整或資訊有所誤差的檔案都還是能夠得到最佳化過的結果。

因此，我們針對各項音樂資訊結合音樂理論的分析分別做研究探討以達到整合後的系統可處理多聲部的 MIDI 音樂、有雜訊(e.g. 錄音所產生的訊號延遲或少數的音高失真等等不為音樂真正所表達的音符)的 MIDI 音樂、或即時性的片段 MIDI 音樂等，我們必須有以下的分析歸納技術：

- a. 調性分析演算法實作。
- b. 音名拼寫演算法實作。
- c. 拍子追蹤與節拍歸納演算法實作。
- d. 和弦分析演算法

有上述的分析歸納技術後我們更加確立了系統的目標，本系統將整合下列的分析追蹤技術與樂理規則的結合：1. 音符分析的第一步為音名拼寫，此分析採用可即時拼寫音名的研究技術(Chew, 2005)[18]，使用由五度線進化的立體模型—螺旋陣列模型，此演算法以先不決定調性並且透過立體的螺旋模型更易表現五度圈每個

音之間的關係。2. 每一個音符的正確拼寫後再透過與 1.相同的螺旋模型做調性分析(Chew, 2000)[8]取得最可能的調性。3. 拍子追蹤與節奏的歸納採用 Temperley(2001)[11]的技術，透過在時間上優先權的給予，從權重周期的分析再歸納節拍。4. 和絃的判斷以小節為單位，利用(Chew, 2000)[8]所提出的中心理論判斷最可能的和絃。5. 爲了考慮音樂的通用性，我們加入了音樂理論的規則例如：數字音符標示法則、拍子的基本架構、調性轉移觀念、流行樂的調性標示法則和流行樂的和弦標示法則。



第二章 相關研究

本研究實作一個數字簡譜產生系統，其中需要包含簡譜系統模型架構、音名拼寫機制、調性尋找機制和節拍歸納機制。為了使本研究的方向、貢獻、定位更能確立，因此系統的建立必須詳加了解許多相關的研究與發展，本章將介紹系統中所採用的相關研究以及技術發展。

相關研究可分為：

- a. 其他簡譜系統與簡譜介紹
- b. MIDI 轉樂譜的相關研究與應用
- c. 音名拼寫的相關研究
- d. 調性尋找的相關研究
- e. 小節與拍追蹤的相關研究及應用
- f. 和弦分析的相關研究

2.1 簡譜與其他的簡譜系統

數字簡譜泛指一般流行樂所採用的簡單記譜法又可稱作數字譜或簡譜，是以阿拉伯數字來表示「標記」音符的一種記譜法。我們在 2.1.1 節中介紹系統中所使用的數字簡譜主要元素，2.1.2 介紹其他相關的簡譜軟體包含國外常見的 abc 簡譜。

2.1.1 數字簡譜元素介紹

音符

以數字 1 至 7 表示 DO~SI。代表的是大調中的自然音階。下列表 2.1 分別為大調與小調音階、MIDI 中所使用的 Pitch class 和簡譜數字的對應表：

音階	C	D	E	F	G	A	B
Pitch class	0	2	4	5	7	9	11
數字	1	2	3	4	5	6	7

音階	A	B	C	D	E	F	G
Pitch class	9	11	0	2	4	5	7
數字	6	7	1	2	3	4	5

(a) 大調

(b) 小調

表 2.1 Pitch class 和簡譜數字的對應表

八度

以中央 C 為中心，此中央在 MIDI 的音符編號是 60，作為基準的位置，簡譜上不做任何位移提高音域或降低音域的表示。提高一個八度記譜方法是在數字上方加上一點，降低一個八度，則就是在數字下方加上一點。若以中央為準提升到兩個八度，即在數字上方加上兩點，反之降低亦為如此。更多個八度則以此類推。

音長

通常只有數字沒有多加橫線或附點的代表四分音符。數字下方一條橫線即為四分音符的長度減半，為八分音符；而兩條橫線則是四分音符長度減半再減半，即成為十六分音符，以此類推。而附點的表示方法即在音符後方加一點將此音符長度增加原來的一半。

休止符

休止符以數字「0」來表示。

小節線

小節與小節之間必須以縱線(稱小節線)分隔表示，而樂曲終止則是特別以一粗一細的縱線表示。

拍號

拍號通常表現方式為：「2/4」、「3/4」、「4/4」、「6/8」等等。分子代表有幾拍，分母代表幾分音符作為一拍。

調號

一般而言，表示調號有兩種方法：可以直接寫「C調」，「C小調」或「Key: C」，「Key: Cm」。還有一種方式是以1或6代表的主音來呈現調號，例如1=C，就是C大調，而6=C就是C小調。在本系統中以第一個表示方法為主。

升降記號

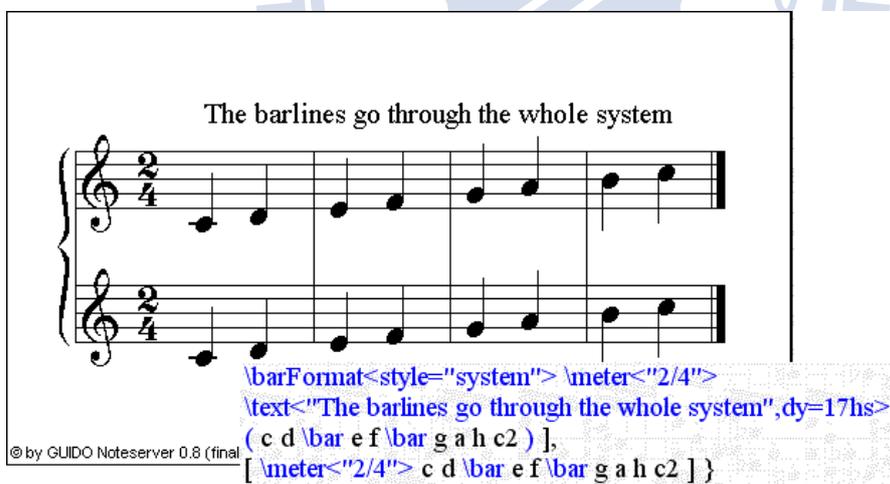
表達升降號即把升或降符號加在數字前面，即為該音升高或降低。

其他記號

結合線，一個音符長度有可能超過一個小節還沒結束，諸如此類的問題必須把音拆成兩個音分別置於一前一後但必須以一個結合的符號將音符標在一起，代表只彈奏一次但音長為兩個音的長加總，結合線通常放置在兩個以上的數字上方。

2.1.2 其他相關簡譜軟體

國外也有所謂的簡單記譜法，但是多以英文字母表示，也就是所謂的 ABC Language，例如：GMN (GUIDO Music Notation)[6][7]所開發的 GUIDO Music Notation Format 在 1992-1993 開始被研究，目前已經完成到一個階段且持續發展，如圖 2.1(a)；還有像 Runabc 開放式的軟體，這是一個可以讓大家所寫的程式掛在軟體上執行，並且開放 code 給大家新增與修改功能，也是應用在 ABC Language 上，如圖 2.1(b)；另外 AbcNavigator 2 此軟體亦是轉譯數位聲音為五線譜以及 abc Language 之簡譜，如圖 2.1(c)；以 XML 延伸標記語言為基礎編寫 Music Space [17]，並且發展各種 Music Markup Language[13] 設計成可以統一或交換樂譜符號，統一許多不同的語法轉換成樂譜，至今都還有各種語法再持續研究。

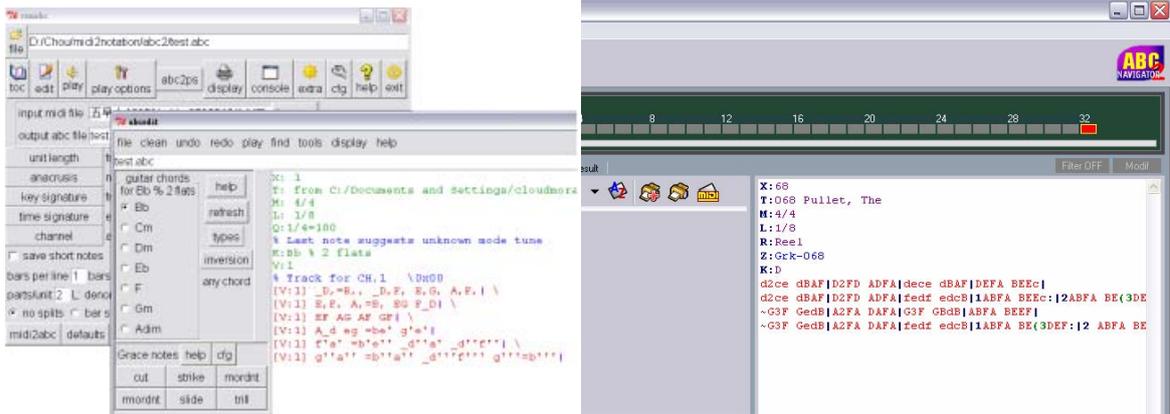


The barlines go through the whole system

```
\barFormat<style="system"> \meter<"2/4">  
\text<"The barlines go through the whole system",dy=17hs>  
( c d \bar e f \bar g a h c2 ) ,  
[ \meter<"2/4"> c d \bar e f \bar g a h c2 ] }
```

© by GUIDO Noteserver 0.8 (final)

(a) GUIDO 的五線譜與 ABC Language 之轉換



(b) 軟體 runabc 之 ABC Language 應用

(c) 軟體 AbcNavigator 2 之 ABC Language

圖 2.1 其他國外 abc language 簡譜軟體介紹

國內目前沒有一個比較完善的數字簡譜軟體，中國則有幾套開發的數字簡譜軟體，像是 CuteMIDI 簡譜作曲家這一套商業軟體，這一套軟體主要是數字簡譜的製作，其中也附加有輸入 MIDI 檔案轉譯為數字簡譜的功能，但尚無系統自動分析之功能。

2.2 MIDI 轉譯樂譜的相關研究與應用

Cambouropoulos(2000)[9]提出實做MIDI轉成傳統的五線譜，當中曾提出了一些必要做到的任務：確認基本的音樂元素、計算音樂事件的聲調、拍的歸納、拍的追蹤、音符起始的量化、音符長度的量化、音名的拼寫等。

Rowe(2001)[12]表示具有音樂才能的程序分為音高結構與時間結構，Chew(2003)[15]也衍伸提出了兩個結構的項目或規則的詳細分類，其中如下圖 2.2 所示，其中音高結構包含了音名拼寫、和絃辨識、調性尋找，而時間結構包含了拍追蹤以及拍歸納。

Meredith(2006)[19]提到，從數位音樂錄製到樂譜的呈現，需要的不只是聲音轉譯系統，也需要 MIDI 轉譯樂譜(MIDI-to-notation)的演算法，包含音名拼寫的技術。

由上述可知，MIDI 轉譯樂譜有其研究價值，且轉譯系統必須具有音名拼寫及拍的追蹤和歸納分析。

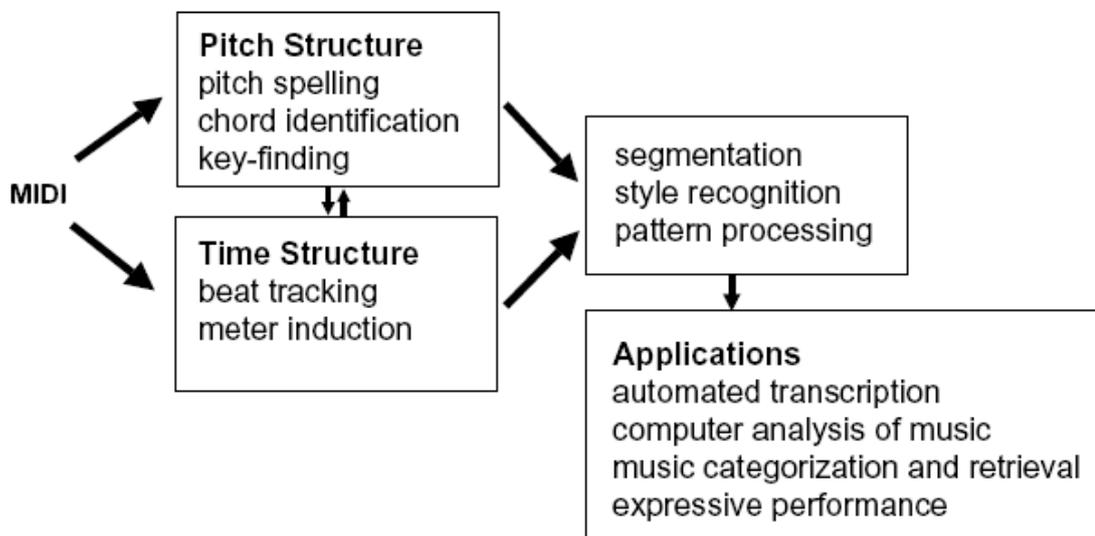


圖 2.2 Music Information Content Extraction and Analysis

(資料來源：MAPPING MIDI TO THE SPIRAL ARRAY: DISAMBIGUATING PITCH SPELLINGS, E. Chew)

2.3 音名拼寫的相關研究

正確的音名跟正確的調性基本上是密切且不可分的，如果要知道一段音樂的調性那一定要先知道它所有正確的音名不可，但音名也必須先有調性的判斷才能準確給予，所以音名拼寫和調性分析通常是並存的。

近年來很多音名拼寫的演算法包含 Temperley(2001)[11]的五度線法 (line-of-fifths approach)，Cambouropoulos(2003)[10][14]的音程最佳化法(interval optimization approach)，Meredith(2006) [19]的 ps13 演算法。這些音名拼寫的研究方法分離了調性尋找與音名拼寫的問題，產生的音名，再做為調性尋找的輸入。事實上，他們在設計中隱藏了調性結構的知識，例如音程的優先權或和聲的解析。而 Chew(2005)[18]年提出建立在螺旋陣列模型上，由中心理論直接拼寫音名。

Temperley 對應音符到五度線，將拼寫結果對應到五度線上最接近的集合，每一組拼寫過後的音符集合會產生一個重量的中心，下一個音名決定於目前中心位置，也就是說下一個音被拼寫的位置必須最接近目前中心的位置。另外，這個演算法防止和聲半音階彼此接連在一起透過和聲導進規則，也透過和聲回饋機制選擇適合的和聲表現。這個音名拼寫的精準度取決於韻律和和聲分析的可信度。

Cambouropoulos 強調音程的關係，選擇很可能發生在調性音階上的音程作音名拼寫。這個演算法的基本原則是 1.臨時升降記號的最少使用 2.音程的最佳化，盡量不要出現增減音程。透過限制音程的選擇，Cambouropoulos 根本上限制了五度線上的特定區域作音名拼寫。Cambouropoulos 認為五度線法可以被視為一個特別的音程最佳化演算法的實例。

Meredith 的 ps13 演算法分成了兩個階段：第一階段利用最像和聲半音階的主音所產生的音符名稱，透過表決決定最可能的拼寫結果，同時在第二階段掃描和修正局部聲音導進的錯誤。

Chew 的螺旋陣列模型拼寫音名演算法由兩部分組成：1. 評估一個片段音樂的樂段(context)並且適當地指派此片段中的所有音名；2. 選擇定義樂段事件的視窗之策略。

然而，Meredith(2007)[20]的 *Optimizing Chew and Chen's Pitch-Spelling Algorithm* 詳細的討論比較 Chew & Chen 的建立在螺旋陣列的音名拼寫與其他的演算法，發現在乾淨處理過的測試資料中 Chew & Chen 的演算法正確率不會有較高的表現，反之，存在雜訊的測試資料例如演奏錄音的音樂則螺旋陣列法正確率較高，由此可見 Chew & Chen 的方法試用於演奏時人所產生的 MIDI 檔。因本研究之目的為建立能處理資訊不完整或資訊有所誤差 MIDI 檔案的系統，因此在本系統中的音名拼寫技術選擇採用螺旋陣列模型。

2.4 調性尋找的相關研究

調性幾乎是決定了音樂的架構，音名和和絃幾乎是取決於調性而來的，因此做音樂分析幾乎沒有不先提到調性或先分析出最可能的調性。

Longuet-Higgins(1971) 提出一個演算法來計算片段單音旋律的調性，其模型建立在調性和音階的連接。當一段音樂開始的時候，模型進行從音樂的左到右；在每一個音符上，模型將會消除每一個不包含此音符的調（以自然音階為主要依據），最後結果即很有可能為此音樂的調性。

Krumhansl (1982) 提出了一個探測音調的數據分布來分析調性。在 1990 年，Krumhansl-Schmuckler 調性尋找演算法，簡稱 K-S，，建立在這個探測方法上，先建立大調和小調的調性分布(Key-profiles)，透過這些分布，K-S 演算法產生一個”輸入向量(input vector)”來判斷一段音樂的調性。也就是說，有 12 個數的向量代表著在這一段音樂中，每一個音類(pitch-class)的總時間長度。接著，計算每一個調性分布向量和輸入向量之間的相關係數，相關係數最高者即為預測的調性。

Chew(2000)提出用螺旋陣列模型做調性尋找，利用中心(Center of effects) 理論可在螺旋陣列上找出具體的調性中心座標，而片段音樂從開始的每一個音符都可以找到一個中心，計算音符中心與每個調性的距離，與音符中心距離最短的即為最可能的調性。本系統亦採用此螺旋陣列模型作調性尋找。

2.5 小節與拍追蹤的相關研究及應用

節拍對於時間維度來說就如同調性對音高維度一樣重要，Gabrielsson(1973) 發現節拍是節奏模式相似性(rhythmic pattern similarity)的一個重要因素，有同樣節拍(雙拍或三拍)的節奏模式會被人認為相似。但近年來從 MIDI 檔案作節拍分析和拍的追蹤的研究並沒有像從音訊檔案做分析來的多。

Povel 和 Essens(1985) 建立一個簡單的演算法，爲了計算一連串音符隱含的節拍或是稱做計時器(clock)。而透過這個模型，有音符出現的地方，特別是長音音符即很有可能爲強拍的位置。

Temperley(2001)的音樂結構概念將優先權的規則建立在 Lerdahl and Jackendoff(1983) 的 *A Generative Theory of Tonal Music* 中所提及的四個規則之上並且加以改進。透過給予拍子優先權來做節拍歸納，主要的優先權規則訂爲五個等級，最後在拍子上可以從權重分析出周期，即可歸納拍子和節奏。

2.6 和弦分析的相關研究

Chew(2000)提出用螺旋陣列模型做調性尋找的同時也提出了利用中心 (Center of effects) 理論找出具體的和弦中心座標，同樣的以計算音符中心與每個和弦的距離，與音符中心距離最短的即爲最可能的和弦。本系統採用此螺旋陣列模型作和弦分析。

第三章 音名拼寫與調性分析

本章中將介紹系統所採用的 Chew(2000) *Towards a Mathematical Model of Tonality* 中調性尋找的方法和 Chew(2005) *Real-Time Pitch Spelling Using Spiral Array* 中的音名拼寫的技術。這兩者的研究都建立在螺旋陣列模型上，但是螺旋的大小不同，亦即半徑與高度的比不同，這影響音名的位置和判斷的機制。下面將先介紹 Chew 的螺旋模型核心概念，系統中公式與變數制定、調性的分析以及音名的拼寫，然後最後 3.5 節再介紹本系統的音名拼寫與調性分析的模組。

3.1 螺旋陣列模型

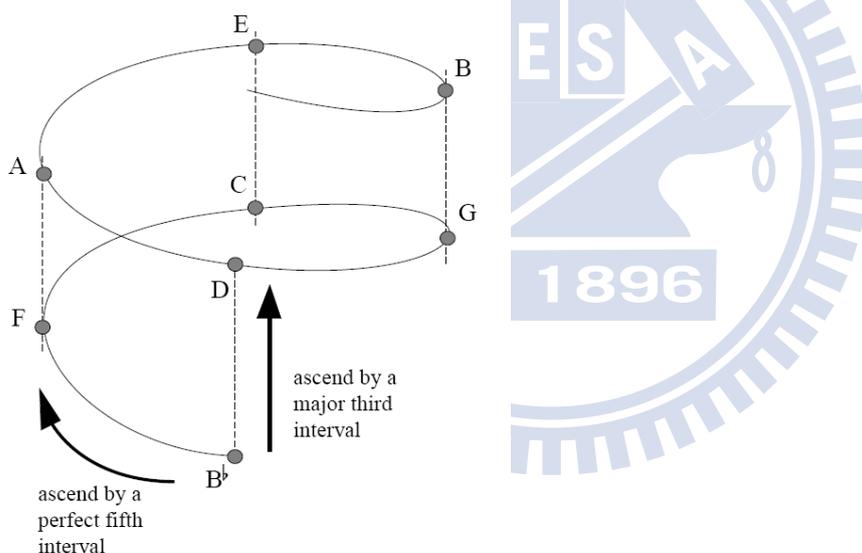


圖 3.1 螺旋陣列

(資料來源：Chew's(2000) *Towards a Mathematical Model of Tonality*)

如圖 3.1 所示，螺旋陣列的運用方法是將每個音名都列在立體螺旋的座標上，音名在螺旋陣列上的座標為公式 3.1 所列。

$$P(k) = \begin{bmatrix} x_k \\ y_k \\ z_k \end{bmatrix} = \begin{bmatrix} r \sin(k\pi / 2) \\ r \cos(k\pi / 2) \\ kh \end{bmatrix} \quad 3.1$$

公式 3.1 中， k 為螺旋陣列上的音符代號， $P(k)$ 代表在螺旋陣列上音符 k 的位置。 $k=0$ 對應的音名可以先任意指定，而其他音名再依照五度線法則推移，如下表 3.2 為 k 值對應的音名。本研究設定音名 C 的 k 值為 0，令螺旋半徑 $r=1$ ，螺旋高度 $h=\sqrt{\frac{2}{15}}$ ，則得到座標 $P(0)=[0,1,0]$ 為音名 C 在螺旋陣列上的位置。

...	F \flat	C \flat	G \flat	D \flat	A \flat	E \flat	B \flat	F	C	G	D	A	E	B	F \sharp	C \sharp	G \sharp	D \sharp	A \sharp	E \sharp	B \sharp	...
...	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	8	9	10	11	12	...

表 3.1 五度線上 k 值對應的音名

利用 3.1 式可計算出各種音名的 k 值在螺旋陣列上的座標，則任兩個音之間的音程距離可被計算，如圖 3.2 為兩音在螺旋陣列上的位置與相對應音程的示意圖。螺旋陣列上，除了音程可以以具體的距離量化外，和絃的影響中心(Center of effect)，以下簡稱 CE，也可以從和絃三個音的座標計算得到。調性的 CE 則取決於調性的 I 級、IV 級和 V 級和絃的 CE 位置。螺旋陣列的 CE 為本章重點，將在下節詳述和絃、調性的 CE 計算。

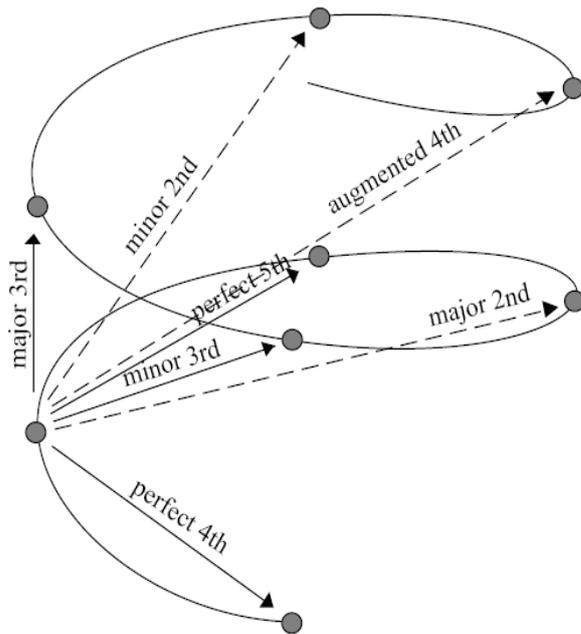


圖 3.2 螺旋上音到音之間的距離

(Chew2000, *Towards a Mathematical Model of Tonality*)

3.2 影響中心CE之公式制訂

和絃的 CE：圖 3.3 顯示大三和絃 CEG 的 CE 表示為 $C_M(0)$ ，小三和絃 ACE 的 CE 表示為 $C_m(3)$ 。 $C_M(k)$ 和 $C_m(k)$ 分別代表的是大三和絃三個音的 CE 與小三和絃三個音的 CE，其中 k 為三和絃根音的值，大小三和絃的 CE 計算如下：

大三和絃的 CE

$$C_M(k) = w_1 \cdot P(k) + w_2 \cdot P(k+1) + w_3 \cdot P(k+4),$$

$$\text{where } w_1 + w_2 + w_3 = 1, w_1 \geq w_2 \geq w_3 > 0$$

3.2

小三和絃的 CE

$$C_m(k) = u_1 \cdot P(k) + u_2 \cdot P(k+1) + u_3 \cdot P(k-3),$$

where $u_1 + u_2 + u_3 = 1, u_1 \geq u_2 \geq u_3 > 0$

3.3

其中 w 及 u 分別為大和絃、小和絃 CE 的權重係數。

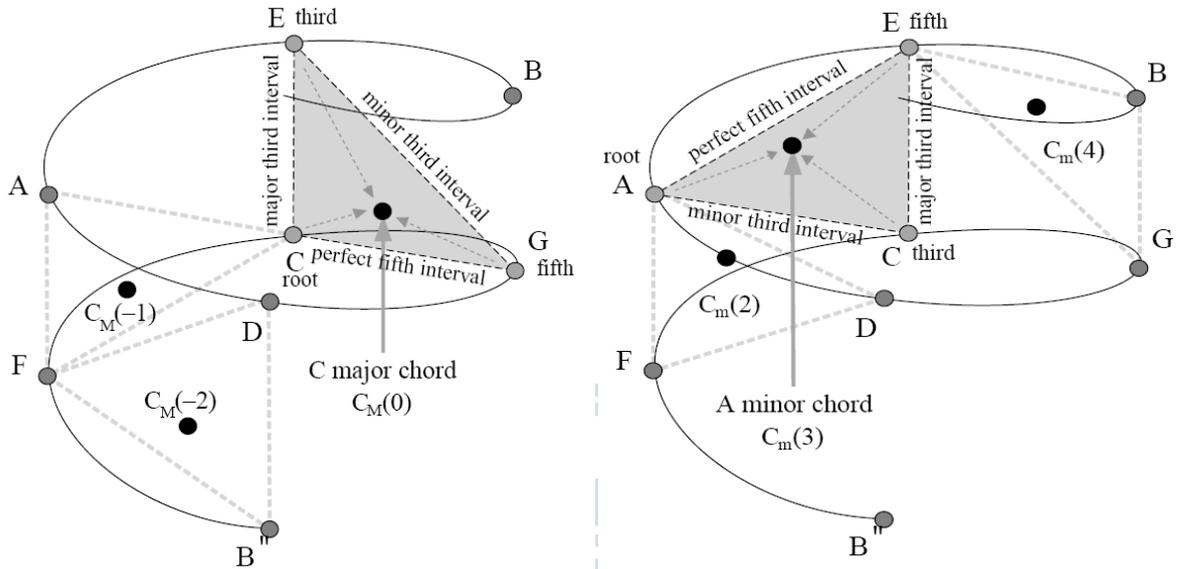


圖 3.3 大、小和絃計算中心意義

(Chew2000, *Towards a Mathematical Model of Tonality*)

調性的 CE: 圖 3.4 為大調的 CE, $T_M(k)$, 由大和絃 I 級、V 級和 IV 級的 CE, $C_M(k)$ 、 $C_M(k+1)$ 及 $C_M(k-1)$ 再計算得到; 小調的 CE, $T_m(k)$, 則由大和絃 V 級以及小和絃 i 級 iv 級的 CE, $C_m(k)$ 、 $C_M(k+1)$ 、 $C_m(k-1)$ 再計算得到。其大小調 CE 的計算公式為:

大調的 CE

$$T_M(k) = \omega_1 \cdot C_M(k) + \omega_2 \cdot C_M(k+1) + \omega_3 \cdot C_M(k-1)$$

where $\omega_1 + \omega_2 + \omega_3 = 1, \omega_1 \geq \omega_2 \geq \omega_3 > 0$

4.4

小調的 CE

$$T_m(k) = v_1 \cdot C_m(k) + v_2 \cdot [\alpha \cdot C_M(k+1) + (1-\alpha) \cdot C_m(k+1)] + v_3 \cdot [\beta \cdot C_m(k-1) + (1-\beta) \cdot C_M(k-1)]$$

where $v_1 + v_2 + v_3 = 1, v_1 \geq v_2 \geq v_3 > 0 \alpha = 0.75, \beta = 0.75$

4.5

其中 ω 、 v 分別為大調以及小調 CE 的權重係數。

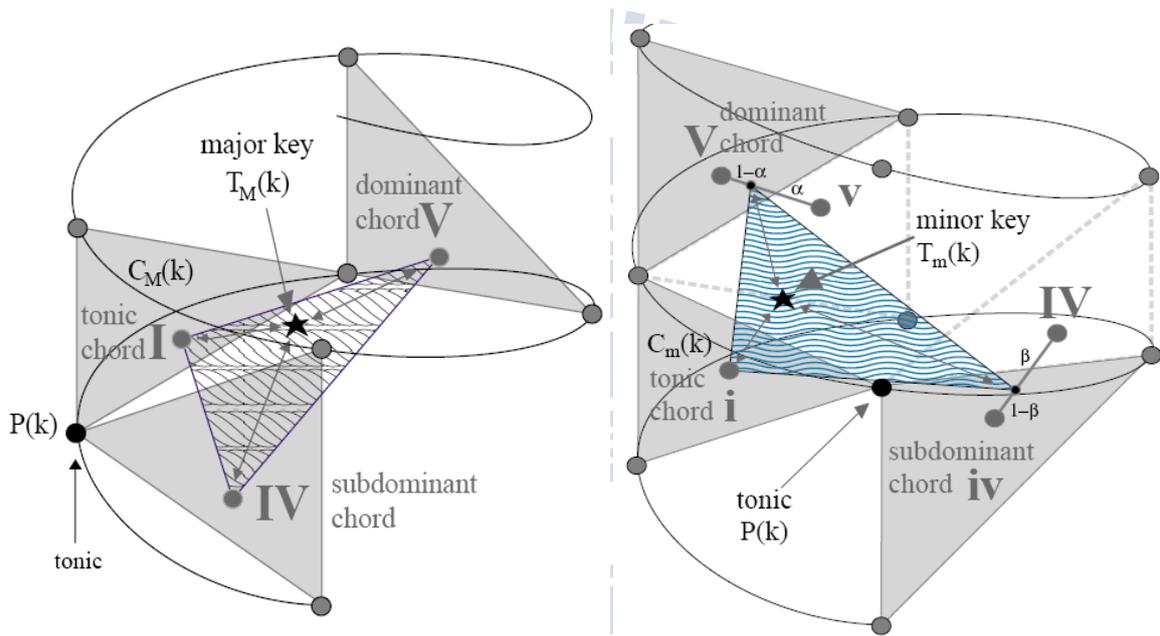


圖 3.4 大、小調計算中心意義

(Chew2000, *Towards a Mathematical Model of Tonality*)

公式 3.2-3.5 中的 w 、 v 、 ω 及 u 在 Chew 的研究中，使用 Flip-Flop Heuristic 找到適當的權重，如下為 Chew 測試音樂中的一組權重係數：

$$w=v=\omega=[0.6025, 0.2930, 0.1045]$$

$$u=[0.6011, 0.2121, 0.1868]$$

3.3 調性分析

中心理論提到，一段樂曲若為調性音樂則音符的中心會落在該調性 CE 的附近，其分析樂曲中的音符越多則越明顯靠近。也就是說，我們可以從計算一段樂曲音符的中心找到最接近的調性。音符的中心計算見公式 3.6，其中的 P_i 是第 i 個音在螺旋上的位置， d_i 為第 i 個音符的時間長度， D_i 則為從起始到 i 為止所有音符的長度加總。

$$c_i = \frac{1}{D_i} \sum_{j=1}^i d_j \cdot p_j, \text{ where } D_i = \sum_{j=1}^i d_j \quad 3.6$$

圖 3.5 為尋找調性的流程圖，利用公式 3.6 得到音符的中心後，再分別計算跟 24 個調的 CE 座標之距離；與音符的中心距離最相近的調，則為可能性最高的調。

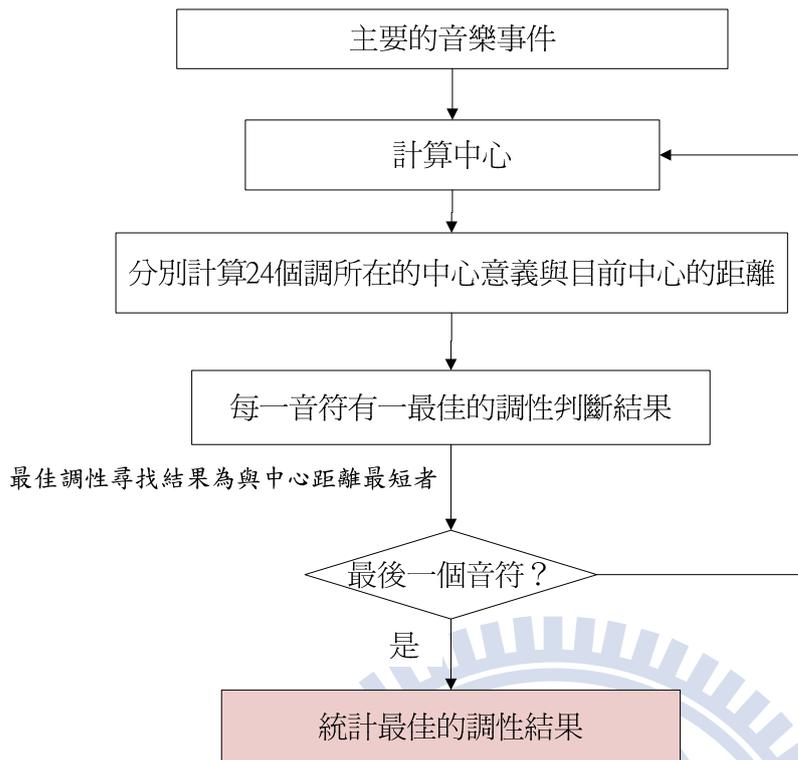


圖 3.5 尋找調性流程

3.4 音名拼寫

然而，尋找調性之前必須先做音名拼寫，因為要確定每個音在調性上的正確音名，才能作調性分析，下面介紹音名拼寫的研究的兩個部分。

3.4.1 音符中心的計算

音名拼寫主要也是利用螺旋陣列上音名座標位置如公式 3.6 所得的音符中心之距離來計算；但不同的是，Chew(2003)在這裡做了區塊的切割，將一首音樂分割成很多個區塊，因此中心的計算將修改成公式 3.7。

$$c_{a,b} = \frac{1}{D_{a,b}} \sum_{j=a}^b \sum_{i=1}^{n_j} d_i \cdot p_i, \quad \text{where } D_{a,b} = \sum_{j=a}^b \sum_{i=1}^{n_j} d_{a,b} \quad 3.7$$

其中 a 和 b 為區塊指標， $C_{a,b}$ 為從 a 區塊至 b 區塊。

圖 3.6 的例子為一段樂曲中前面的音符已經計算出中心，若下一個音符可能的兩個音名有兩個分別是 G#、Ab，而 G#的座標位置距離中心明顯的較近，所以音名拼寫的結果傾向為 G#。更具體的說，從 MIDI 檔讀取的每個音符將可能符合兩個或三個最可能的音名，表 3.2 為音名拼寫的對照表。

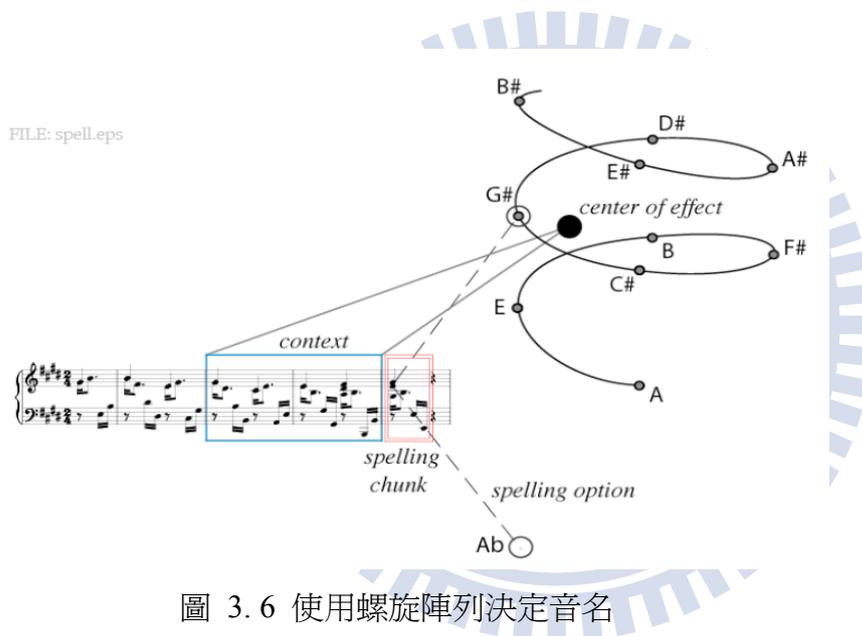


圖 3.6 使用螺旋陣列決定音名

(資料來源: Chew's(2005) *Real-Time Pitch Spelling Using Spiral Array*)

Option 1	(SA index)	Option 2	(SA index)	Option 3	(SA index)
B#	(12)	C	(0)	D $\flat\flat$	(-12)
C#	(7)	D \flat	(-5)	B##	(19)
C##	(14)	D	(2)	E $\flat\flat$	(-10)
D#	(9)	E \flat	(-3)	F $\flat\flat$	(-15)
D##	(16)	E	(4)	F \flat	(-8)
E#	(11)	F	(-1)	G $\flat\flat$	(-13)
E##	(18)	F#	(6)	G \flat	(-6)
F##	(13)	G	(1)	A $\flat\flat$	(-11)
G#	(8)	A \flat	(-4)		
G##	(15)	A	(3)	B $\flat\flat$	(-9)
A#	(10)	B \flat	(-2)	C $\flat\flat$	(-14)
A##	(17)	B	(5)	C \flat	(-7)

表 3.2 音名拼寫的對照表

(資料來源: Chew's(2005) *Real-Time Pitch Spelling Using Spiral Array*)

圖 3.7 為目前所提到的拼寫音名流程，除了第一個區塊之外，已經拼寫正確音名的區塊會計算出一個中心，在目前的區塊中，給予音符可能拼寫的兩個或三個音名 $\langle I_{i,j}-12, I_{i,j}, I_{i,j}+12 \rangle$ ，這些音名分別到中心的距離中最短的那一個就是最可能被拼寫的音名，見公式 3.7。而第一區塊的初始中心給定為 2，但必須特別注意的是在音名拼寫的螺旋陣列中，Chew 設 $r : h$ 的比例為 $\sqrt{\frac{2}{15}}$ ，與調性尋找所設定的比例是相反的。

$$I_{i,j}^*(c_j) = \arg \min \{ \|P(I_{i,j}-12) - c_j\|, \|P(I_{i,j}) - c_j\|, \|P(I_{i,j}+12) - c_j\| \} \quad 3.8$$

其中 j 代表目前區塊的指標。

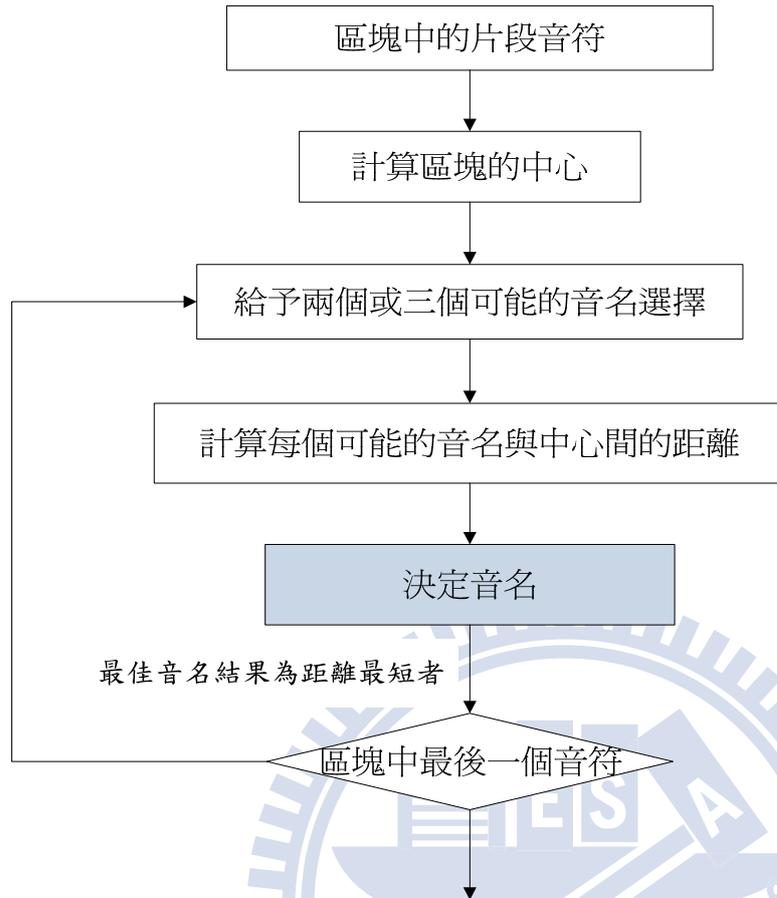


圖 3.7 拼寫音名流程

3.4.2 二步驟的疊代法

另外一個部分是運用疊代的方式再做一次音名拼寫的檢查。在本研究中疊代的方法分為兩個步驟：第一步驟透過滑動的視窗計算中心；第二步驟再以加重靠近目前音符局部樂段的權重，做第二次的中心計算。一個視窗所包含的區塊數設為變數 w ，用於第一步驟；從目前區塊往前包含 w_r 個區塊用於第二步驟。

步驟 1：利用音符中心 c_j 音名拼寫，其公式如下，

$$C_j = C_{j-w}, j-1, \quad 3.9$$

其中 j 為目前的正在拼寫音名的所屬區塊。

步驟 2：用混合的音符中心再進行一次音名拼寫，其公式如下，

$$c_j = f \cdot c_{j-w_r, j} + (1-f) \cdot c_{1, j-1} \quad 3.10$$

其中 f 為局部樂段的權重， $0 \leq f \leq 1$ 。

圖 3.8 中 $w_r=0$ ，意味著步驟 2 沒有作用，這個作法就是只有做滑動視窗而沒有再次疊代；而圖 3.9 中 $w=4$ ， $w_r=2$ ，執行第一步驟的滑動視窗而計算中心之後，接著執行二步驟的方法，計算 1. 包含本身的區塊與前一個區塊的計算並給予較高權重，2. 從整個樂曲中的第一個區塊到目前的前一區塊計算值中心並給予較低權重，將 1.和 2.的值以不同權重的方式相加所得才會是最後真正拼寫的中心。

圖 3.8 滑動視窗方法($w=4$)

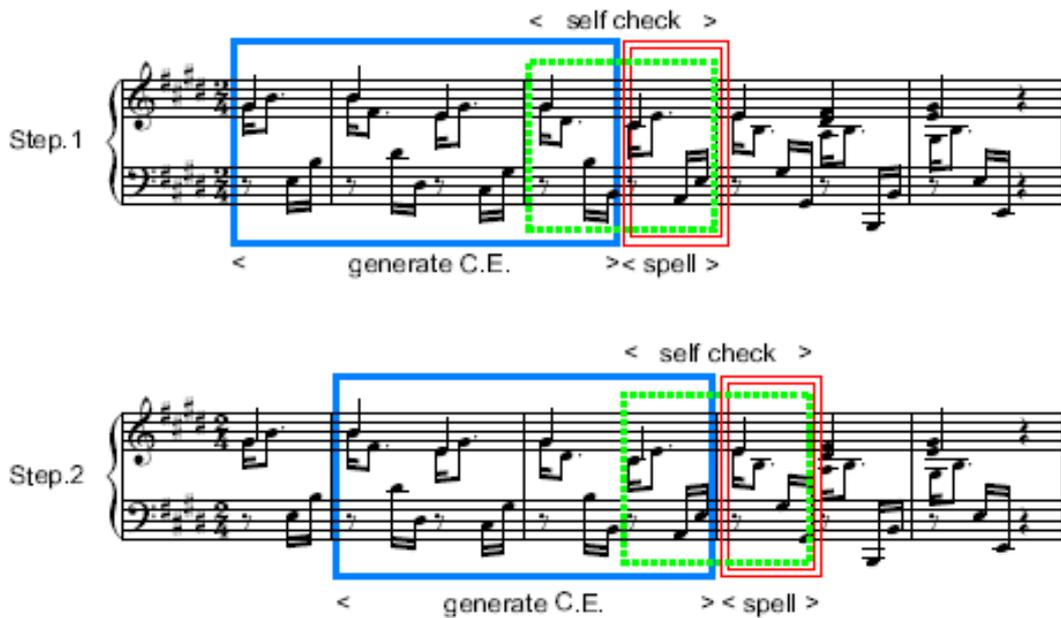


圖 3.9 兩步驟指派方法($w=4$, $w_r=2$)

3.5 系統中音名拼寫與調性分析整合模組

表 3.3 為 Chew(2005)[16]在 *Determining Context-Defining Windows: Pitch Spelling using the Spiral Array* 中的實驗數據，此音名拼寫有三種演算法：1. 如 3.4.1 節所提的部分，只有透過前面區塊所計算的中心；2. 在本小節所提到的滑動視窗計算中心但不做疊代計算；3. 滑動視窗後疊代再計算中心。然而，從表 3.3 可以看出三種演算法的正確率都超過了 90%的高準確率，有滑動視窗或是滑動視窗再疊代的方法提升了約 2~3%的準確率，因此系統實作音名拼寫考慮到系統的效能以及三種演算法準確率都相當高的情況下，採取上述的演算法 1.，以累加的方式進行計算中心。

Method	Parameters			Errors (1516 notes)	% Correct
Cumulative				73	95.18
Sliding Window (w)	4			31	98.00
	8			47	96.90
	16			40	97.36
Two-Phase (w, w _r , f)	4	2	0.6	28	98.15
	4	3	0.8	27	98.22
	4	3	0.7	27	98.22
	8	2	0.9	31	97.96
	8	4	0.9	40	97.36
	8	6	0.9	27	98.22
	16	4	0.8	40	97.36
	16	6	0.8	30	98.02
	16	8	0.9	37	97.56

表 3.3 音名拼寫的實驗數據

(Chew 2003, Determining Context-Defining Windows: Pitch Spelling using the Spiral Array)

圖 3.10 為音名拼寫與調性分析流程圖，每個音符先做音名拼寫，音名通常有兩個到三個選擇分別計算中心做以最短距離的音名做拼寫，音名拼寫模組請見前面 3.4 節介紹；拼寫正確音名後再透過調性分析模組找每一個音符最有可能的調，同樣的以計算中心後跟 24 個調比較距離，詳細方法請見前面 3.3 節介紹；最後在統計每個音符所記錄最佳調性決定整首樂曲的調號。

計算音符中心皆以公式 3.6 所累加，使用同一個大小的模型，也就是前面調性分析所建立的半徑 $r=1$ ，高度 $h=\sqrt{\frac{2}{15}}$ 的螺旋。因為螺旋的大小跟 Chew(2005) 原來的不一樣，所以前置第一個音符的音符中心也不會是前面 3.2 節所提到的以 2 為初始。我們必須先做一個前置作業計算初始的音符中心，如圖 3.11 的前置流程。

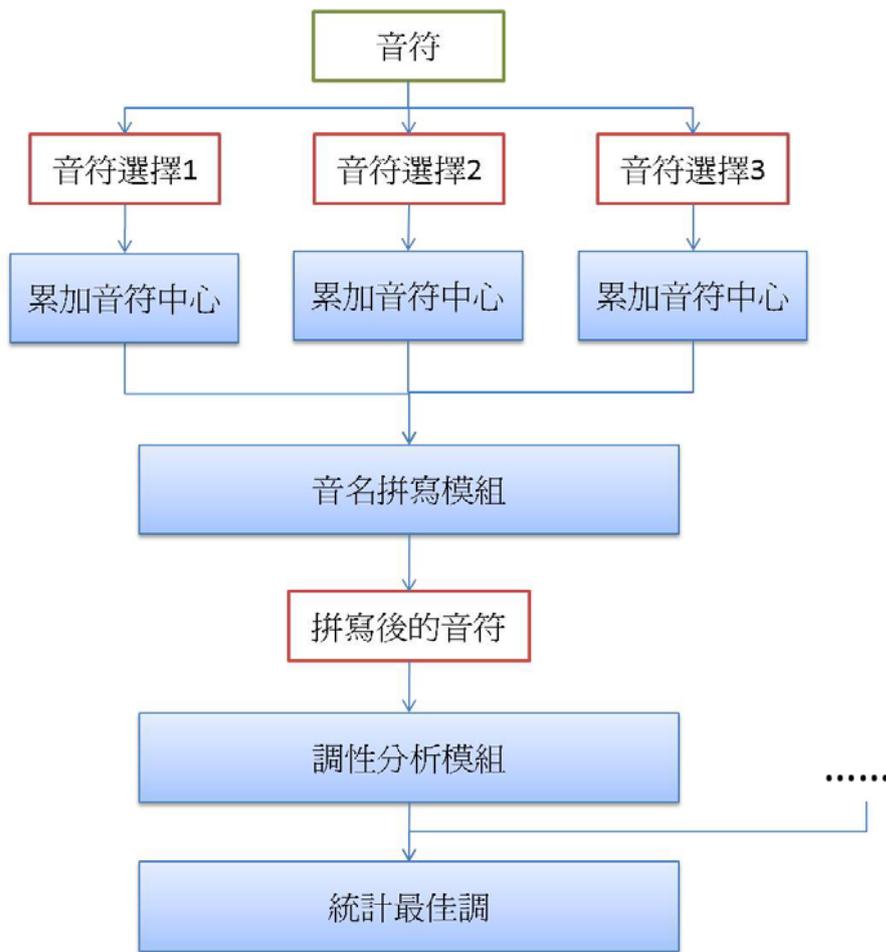


圖 3.10 音名拼寫與調性分析流程圖

圖 3.11 為前置計算初始中心，開始時樂曲第一個音符給定三個可能的音名選擇，三個音名選擇請見表 3.2 的音名拼寫的對照表，取一小段做為前置分析的樂段，分別往下執行音名拼寫與調性，前置分析得樂段音符數由系統給定，目前設定前置計算的音符數 n 為 50 個音，若樂曲本身音符數少於 50 則 n 為樂曲的總音符數。每一個音名選擇給定後立即計算 n 個音符數的音符中心並估計調性，再紀錄一組調性至 n 個音符數的中心最短距離，前置作業最後會有三組距離的結果，再將三組結果做比較。第一個音符選擇擁有最短距離的音名做拼寫，並且以此前置作業所計算的音符中心為下一個音符的累加音符中心初始值。

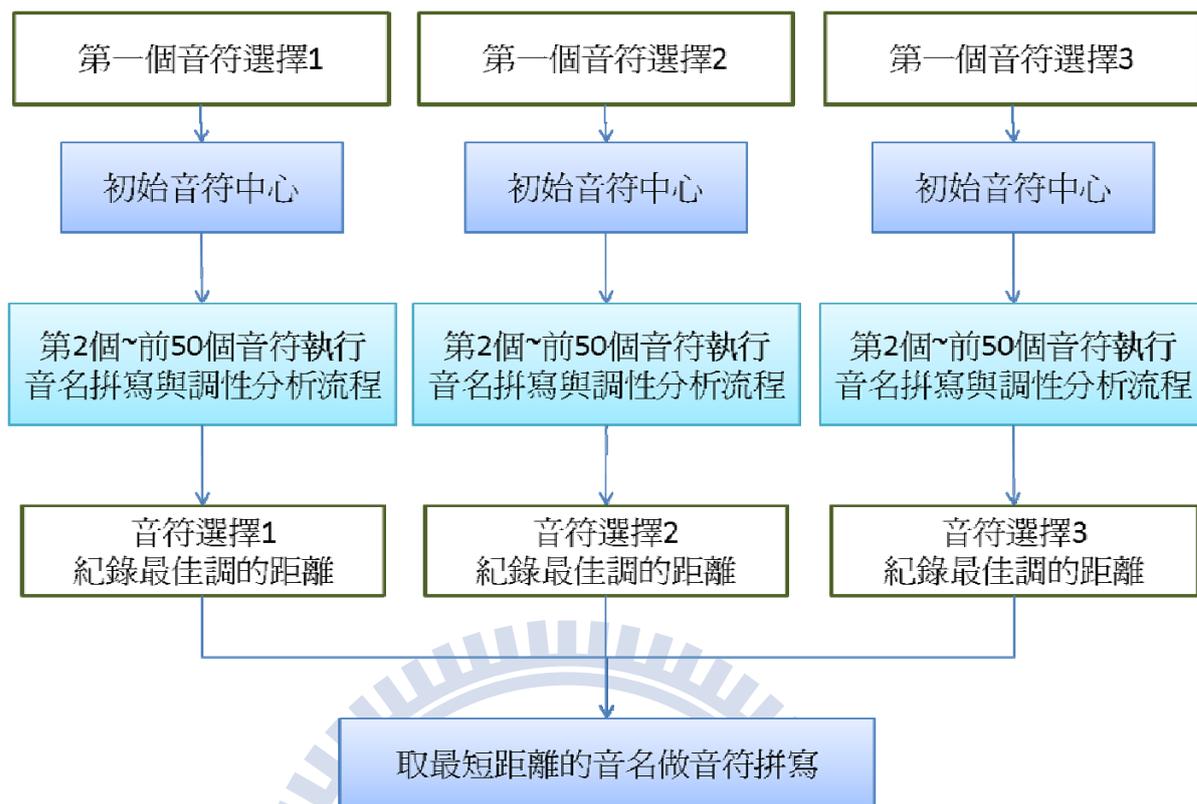


圖 3.11 前置作業流程

第四章 節拍歸納與和弦分析

本章將介紹系統中的節拍歸納模組以及和弦分析模組。本研究的節拍歸納是採用 Temperley(2001) *Cognition of Basic Musical Structures* 中的音樂結構概念，主要是透過給予拍子優先權來做節拍歸納。而和弦分析則採用 Chew(2000) *Towards a Mathematical Model of Tonality* 所提出的中心理論找尋和弦的方法。

4.1 節拍歸納

本系統利用 Temperley 的音樂分析系統中節拍尋找(meter finding)程式，Meter program。執行後的結果會呈現：在每一個時間間隔上，越強的拍子權重總和越大，這裡權重等級分為 0~4，給予權重的規則將在 4.1.1 節中介紹。4.1.2 節再介紹系統中節拍歸納的模組流程。

4.1.1 拍子優先權的給定

在音樂的結構上，平均間隔開來的拍子有好幾個等級，而所謂的等級由下面的幾個規則所給定，符合一個規則以多加一個權重表示，如圖 4.1 所示，符合一個規則即以一個點表達優先權，透過這些等級不同的權重可以看出拍子的強弱與分布示意，如此將可以用於歸納節拍。下面五個主要的音樂優先權規則(Metrical Preference Rules，簡稱 MPR)：



圖 4.1 給定優先權後的音樂結構

(D. Temperley, *The Cognition Of Basic Musical Structures*:MIT Press, 2001.)

MPR 1 (音符事件規則)：偏好將強拍置於音符事件開始(event-onsets)的位置。換句話說，越多音符事件開始的同一時間點上，即為較佳的拍子位置，主要用在多音樂曲的判斷。

MPR 2 (音符長度規則)：偏好將強拍置於長音音符事件開始的位置。也就是說，音符的長度越長，即為較佳的拍子位置，而此規則主要是用在單音樂曲的判斷；音符的長度是指音符開始到下一個音符開始的時間間隔(inter-onset interval，簡稱 IOI)。

MPR 3 (規律規則)：偏好拍子被最大化地平均間隔開來。這裡希望不要先量化拍子。在原來的 Lerdahl and Jackendoff 的模型中，節拍尋找之前已經先做了量化，這可能讓一些特殊音符長度，例如像三連音，被平均的間隔起來，如此會失去原有的表現意義並且呆板化。但是不先做量化則會存在了不

精確也不很規律的拍子，這將會使得節奏尋找會沒辦法運作，所以音樂結構必須要不斷的調整去符合音樂。

MPR 4 (群組規則)：偏好將強拍置於靠近群組開始的位置。

群組化分為兩部分：第一部分是將結構群組化，一般在越句的尾端通常出現長拍，這會造成強化尾端拍子的問題。第二部分則是邊界群組化，邊界群組化發生在長拍和休止之後，這一樣會有誤判拍子的情形。因此，解決方式為在片段的第一個等級 3 的拍子上加上一個等級 4 的權重。

MPR 5 (雙拍規則)：偏好在等級之間的關係以雙拍為主而不是三拍。

也就是說，就是傾向先假設拍子為雙拍結構而不是三拍子的結構。

4.1.2 系統中節拍歸納模組

執行程式之前，系統必須先將 MID 資訊轉譯為其所需要的輸入檔，.note 檔；圖 4.2 為節拍歸納的模組流程，系統產生執行檔 meter.exe 所要求的輸入檔.note 檔，執行 meter.exe 之後將輸出每個時間點上的權重如圖 4.2 右下角顯示，我們再依權重結果計算時間周期得到節奏周期，即為我們所要的節拍。然而，要如何歸納節拍呢？我們統計每一組周期除了第一組，因為第一組出現在強拍上得很大的機率是整首曲子的第一個音，這很有可能誤判為小節中的第一個音但事實上很有可能是弱起拍子，最後統計每一組周期取多數決，以最多數的周期為此樂曲的節拍。

透過規則中的 MPR 1 和 MPR 2 使系統中不論是單音或是多音的樂句都可以更準確的找到節拍的周期。但另外一個問題是，如果是 3/4 拍的樂曲最後取得結果很有可能為 6/4 拍，這是因為在規則中 MPR 5 是以雙拍為主而不是三拍給定權重，所以會三拍子的樂曲在出現等級 4 可能從三個拍子變成六個拍子為一循環周

期。所以在本系統中，將出現 6/4 拍判斷為 3/4 拍。因為在一般樂理中，節拍上通常以 3/4 拍為主，6/4 則較少見，因此本系統採取此政策矯正這個問題。

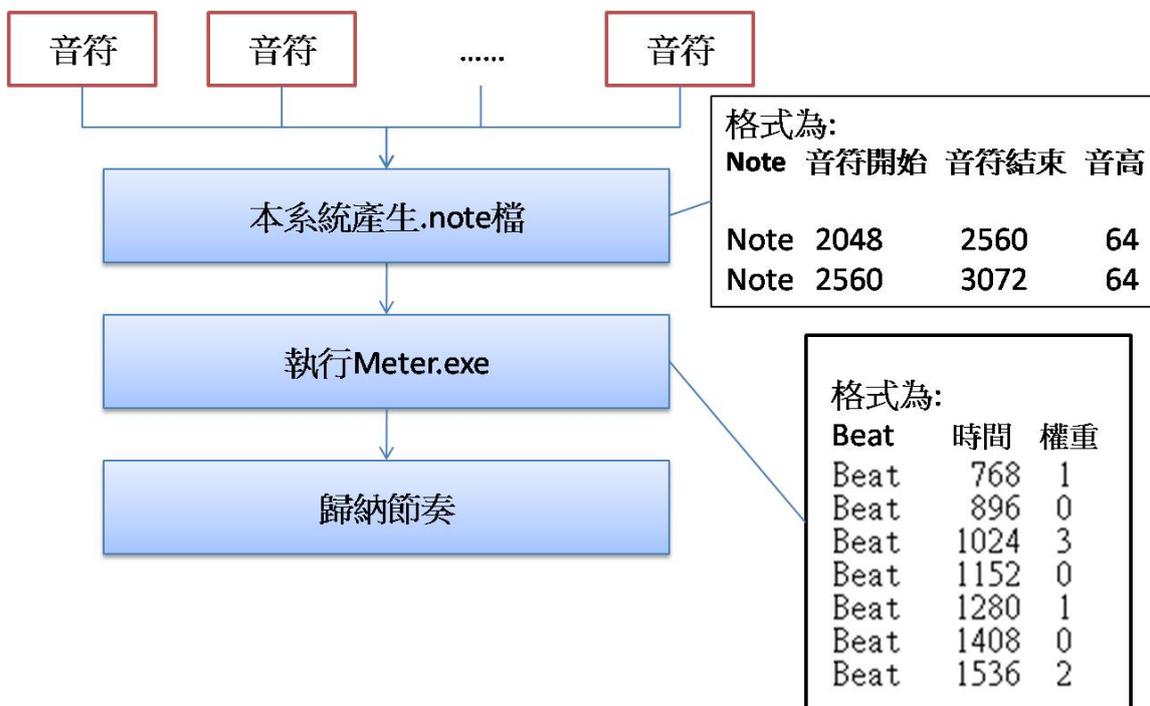


圖 4.2 節拍歸納的模組流程

4.2 和弦分析

在第四章我們介紹了 Chew 的螺旋陣列，也知道透過螺旋的核心理論可以找到其調性和每個音符的音名。而在分析調性的同時，也計算了和弦的 CE，計算調性的中心可以計算，和弦中心也可以計算，那麼我們就也可以運用與第四章分析調性相同的理論分析尋找和弦的可能組合。也就是說，我們也可以從計算一段樂曲音符的中心找到其和弦。音符的中心計算見公式 3.6。

4.2.1 和弦CE公式

分析和弦不僅僅只有大三和弦和小三和弦，大小三和弦得公式請見公式 3.2 以及 3.3。所以除了透過 3.2 節已經建立的大小三和弦 CE 計算，另外至少要再建立減三和弦、屬七和弦、大七和弦以及小七和弦請見下面公式 4.1 到 4.4，其中減三、屬七和小七和弦 Chew(2000)已經定義。

減三和絃的 CE

$$C_M(k) = \frac{1}{3} \cdot P(k) + \frac{1}{3} \cdot P(k-3) + \frac{1}{3} \cdot P(k-6) \quad 4.1$$

屬七和絃的 CE

$$C_M(k) = \frac{3}{8} \cdot P(k) + \frac{1}{8} \cdot P(k+1) + \frac{2}{8} \cdot P(k+4) + \frac{2}{8} \cdot P(k-2) \quad 4.2$$

大七和絃的 CE

$$C_M(k) = \frac{3}{8} \cdot P(k) + \frac{1}{8} \cdot P(k+1) + \frac{2}{8} \cdot P(k+4) + \frac{2}{8} \cdot P(k+5) \quad 4.3$$

小七和絃的 CE

$$C_M(k) = \frac{3}{8} \cdot P(k) + \frac{1}{8} \cdot P(k+1) + \frac{2}{8} \cdot P(k-3) + \frac{2}{8} \cdot P(k-2) \quad 4.4$$

4.2.2 和弦分析模組

圖 4.3 為和弦分析模組與流程，此分析模組主要分為兩個部分：一個部分是先以調性的音階產生各個可能發生的六種和弦 CE，另外一個部分預設以小節為單位計算小節內所有音符的中心，再以音符中心找尋距離最短的和弦，即為小節內的和弦。分析和弦和分析調性不同的地方是在於累加音符中心的區塊不同，簡單的說，調性決定於全部的音符不能單看某一個音符或某一小段內的音符；而和弦決定於某一區塊，可能是以小節為單位或是以拍為單位即產生一個不同的和聲。因此在圖 4.3 中，以小節塞選音符也可以置換成以兩拍為單位尋找一個和弦。

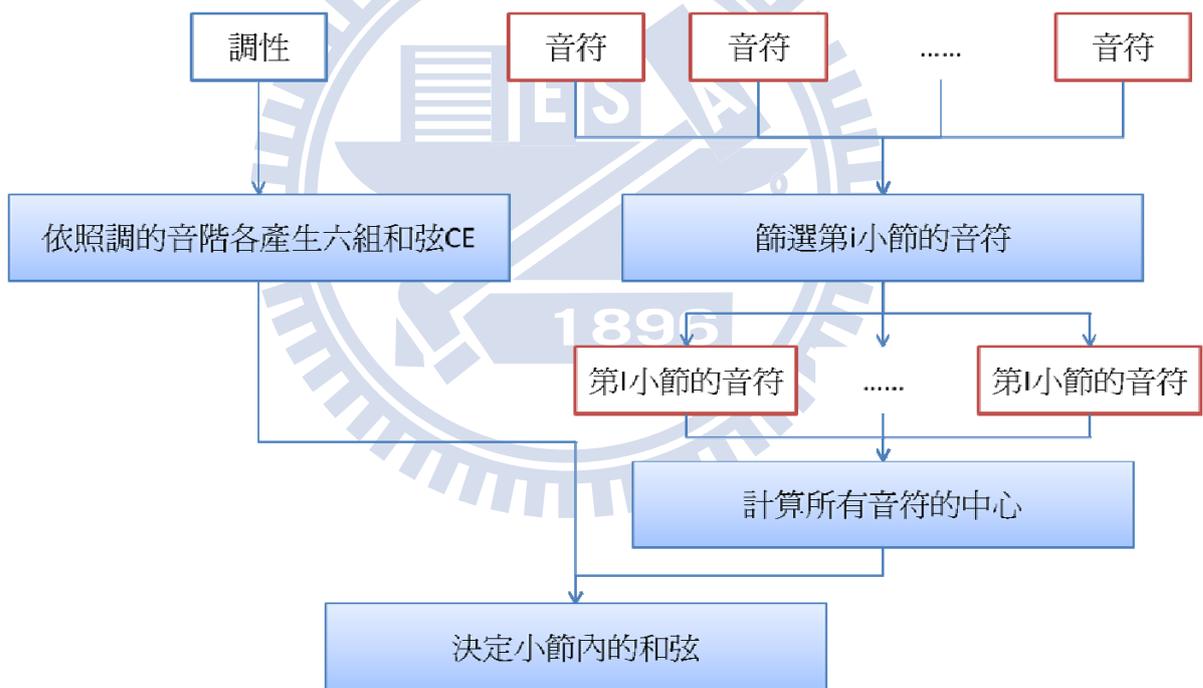


圖 4.3 和弦分析模組與流程

第五章 系統架構與功能

本章將介紹本系統的架構與流程、功能設計和使用界面三部分。5.1 節為整個系統的架構從 MIDI 轉樂譜的分析到結果輸出，5.2 節將說明系統的模組與機制設計流程。

5.1 系統整合系統架構

圖 5.1 為數字簡譜系統的流程圖，這亦是本系統所建立的基本架構。而圖 5.1 虛線區塊為本研究探討的範圍，簡單來說，系統在分析 MIDI 資訊中的音符出現與結束時間、音高、音長等之後，做音名正確拼寫、調性分析以及節拍的歸納，最後分析和弦加強簡譜的功能性，透過這些分析與歸納的結果再整合後，成爲一個擁有基本且重要架構的簡譜平台。

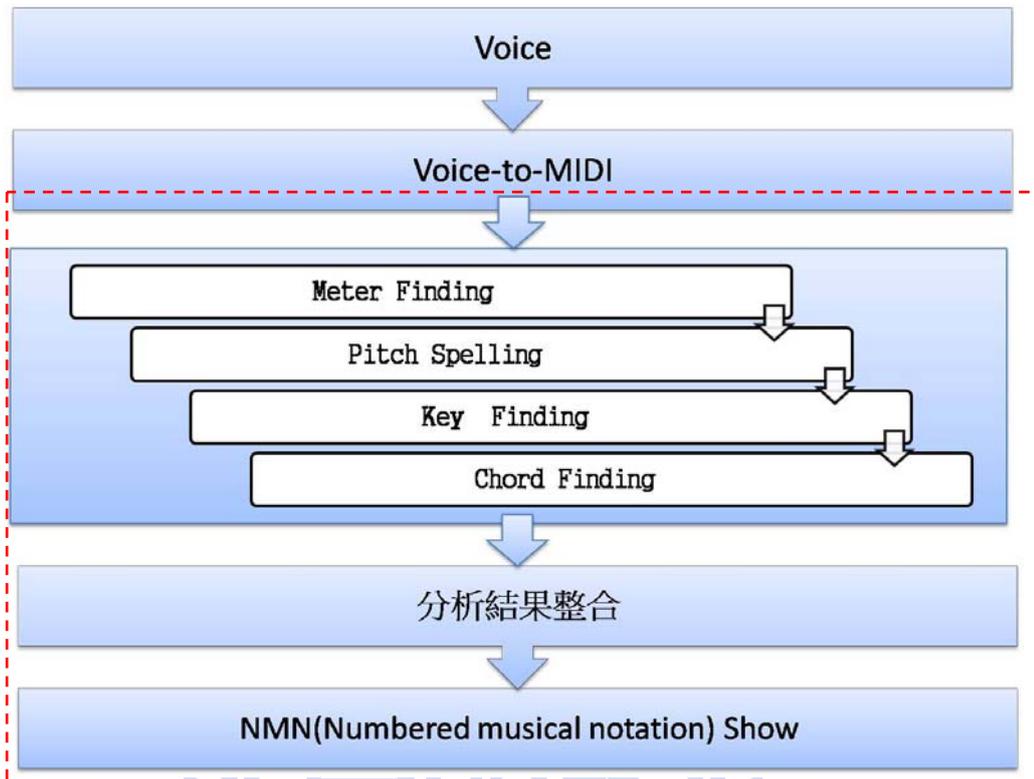


圖 5.1 數字簡譜系統流程圖

5.2 模組與機制設計

從使用者的角度來看，輸入一首 MIDI 檔案透過系統可以方便的得到樂譜，樂譜預設以相對音高的數字簡譜呈現，當然系統也提供選擇將相對音高換成絕對音高呈現。但無論是相對音高或是絕對音高，一首音樂的基本架構首要具備：明確的找到調性，其次就是節拍的定義，有了前面兩者可以分析和弦的組合。為此我們規劃了調性分析模組、音名分析與拼寫模組、拍追蹤與節拍歸納模組以及和弦分析模組。

圖 5.2 為系統中主要分析模組的流程圖，大致分為兩個主線：一個是音符中音高、旋律的處理，另一部份是以音符的時間為主要處理對象。旋律處理的流程為音名拼寫後，再做調性分析與統計；而時間則是分析所有音符的開始時間與音符間的時間長度，歸納拍子的周期與節拍。最後再結合兩個主線的析後的結果進行和弦分析。

在輸入 MIDI 檔案之後，必須有一個前置作業將 MIDI 的訊號轉譯為易讀的事件字串，而在 MIDI 解碼後，我們分析音符事件建立了音符物件，其中包括有音符的 MIDI

音高代號、音符的名稱、音符開始與結束的時間、音符的調性以及音符可能的和絃，透過音符物件更方便模組之間的群取與溝通。圖 5.3 即為完整的系統流程與人機互動，對照上圖 5.2 的模組圖，在模組分析之前我們將前置作業細分為 MIDI 解碼和音符事件分析與物件建立兩個部分。MIDI 解碼後輸出字串再分析音符的事件與和建立每個音符物件；而在調性分析模組完成後，我們已經在音符物件中記錄了每個音符的音名和調號，並且傳送調號給調性轉移機制，透過此轉移機制可將絕對音高轉置為相對音高；最後以樂理規則解讀機制將所有包括音符、和絃、小節線等整合，也就是說當所有分析模組皆已完成後，再透過樂理規則解讀機制將所有分析結果一一解讀為數字簡譜的形式，在使用者介面輸出。

MIDI 訊息可以依照定義的不同，而有不同的解讀，不同定義之間的樂器將會無法正確溝通。因此業界制定了一套控制信號以及合成樂器的規範，稱為「通用 MIDI」(General MIDI)，並且依照制定的詳細程度給予不同的「等級」(Level)。General MIDI Level 1，給予各種控制信號編號；其中包括了音高、調變、預設樂器、延音程度、音量、力度等常見的數位樂器表現方式。但為了解讀 MIDI 訊息信號，程式設計者須以背誦位元編號的方式來了解每個 MIDI 訊息裡面所包含的資訊。因此，系統將使用者輸入的 MIDI 檔案解碼並建立描述事件的字串方便程式編輯者讀取和分析，輸出結果如表 5.1 所表現的 MIDI 格式字串。

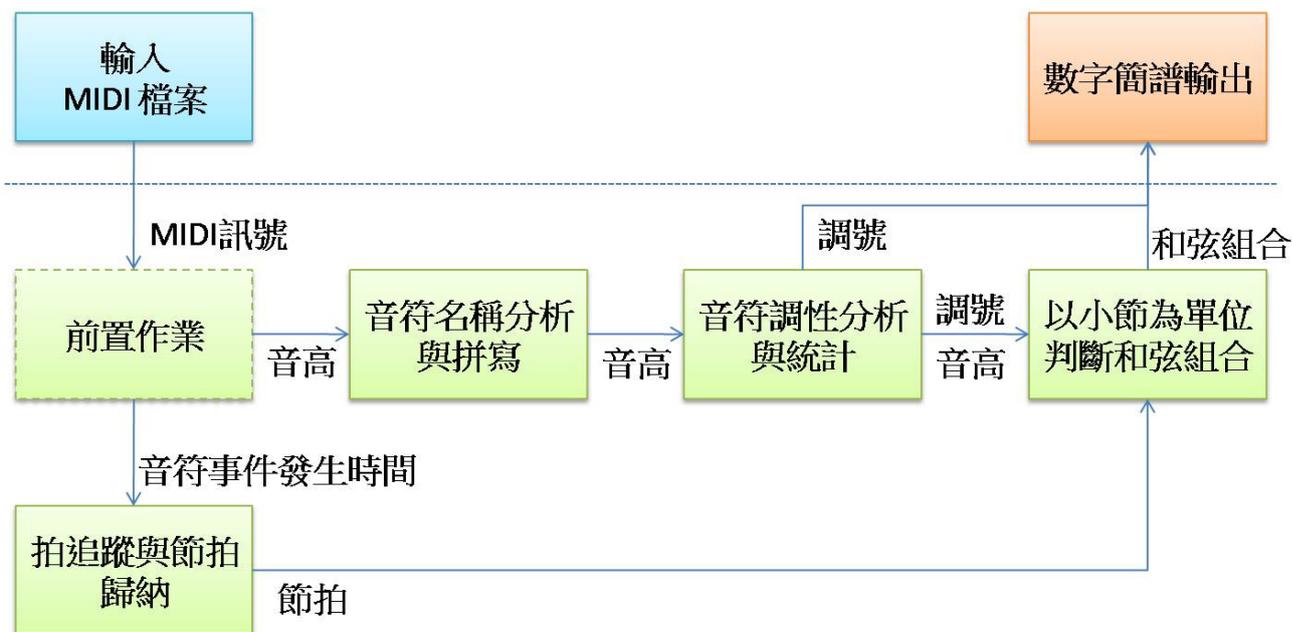


圖 5.2 系統中主要分析模組的流程圖

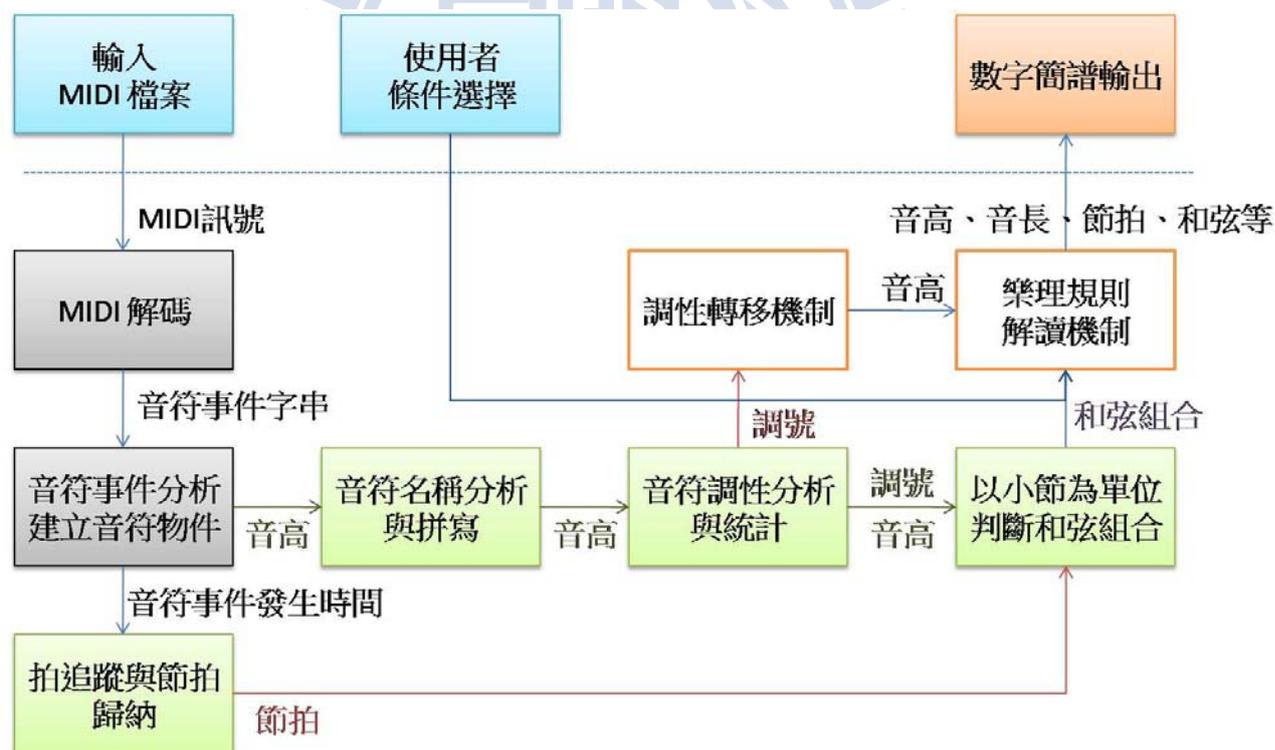


圖 5.3 系統流程與人機互動

檔頭	Header	format=1	ntrks=1	division=1024	
音軌	Track 1 contains 999 bytes				
樂曲資訊	0 Metatext time signature=4/4				
	0 Metatext tempo=120.00 bpm				
音符資訊	tick	音符開始	Channel	音符	音量
	0	Note on	1	71	57
	1024	Note on	1	71	0
音軌結束	122880 Meta event, end of track				

表 5.1 系統建立的 MIDI 通用格式

下圖 5.4 為音符物件的建立流程，輸入的事件字串包含檔案的標頭訊息與音符事件，所以在流程中先分析檔案所給的資訊包括檔案格式、音軌數、還有拍的時間單位 (tick) 等等，再來才是解析出每一個音符的資訊，把每個音符的開始與結束找到，最後分析音符的基本資訊並且建立音符的物件。另外，每一個音符物件建立時會包含音符 MIDI 音高編號、音符開始時間和音符結束時間和音符的時間長度這些基本訊息。而音符物件本身是系統模組間重要的溝通物件，運作在模組間，音符物件會將更多重要的音符資訊記錄下來並且使用在下一個模組中。

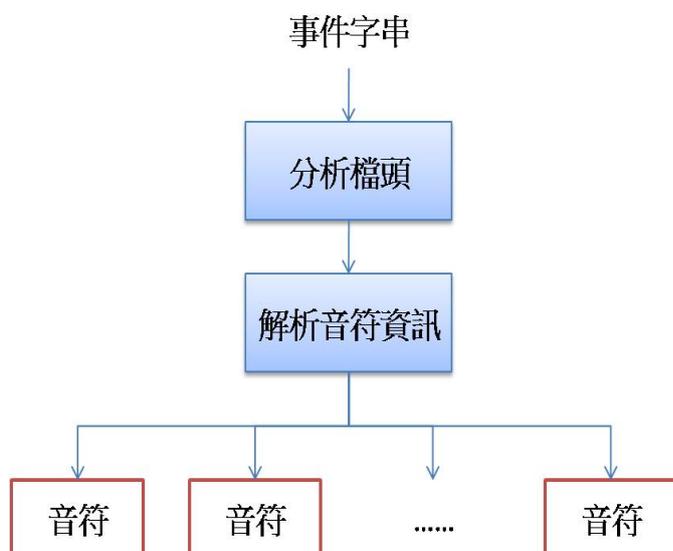


圖 5.4 音符物件的建立流程

圖 5.5 為調性轉移機制的流程圖，系統在對調性做了正確的分析後，首先先確定此樂曲應為大調或小調，然後再計算樂曲本身的調性主音至 C 大調或 a 小調的音程距離即為位移的距離，最後每一個音符都必須移動上步驟所計算出來的位移量。調性轉移對相對音高的表示方法是很重要的一個步驟，簡單的說，相對音高表示法即是將各種大調移置 C 大調表示，小調則以 a 小調表示。

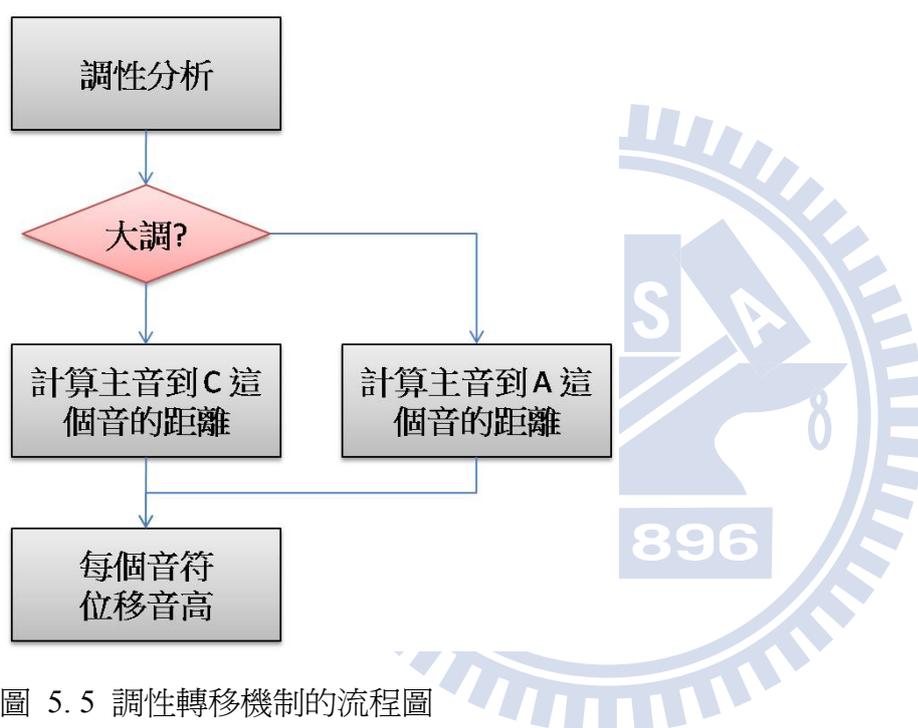


圖 5.5 調性轉移機制的流程圖

樂理規則解讀機制主要在處理需要用到的各種樂理定義，例如同樣都是 Do 這個音在不同八度上表現的差異等等。此機制分為音符規則和音樂架構規則兩部分，先對音符的定義對應出音符的字型，再透過音樂架構規則來決定輸出。

圖 5.6 為定義音符輸出格式的流程，首先決定音高對應的數字符號，再從每一個音長定義音長符號的表現，接著判斷音符的臨時升降記號，最後依照音高八度的位置附加上加點或下加點的高度符號，在使用者介面以數字簡譜字型輸出。

圖 5.7 則為音樂架構規則的流程圖，分作兩個部分，一個是調性的表現定義，另一個是與時間軸和輸出的座標位置有關的：先定義小節線的寫法和座標位置，然後決定每一個音符的座標位置，最後再定義和弦的寫法並且決定和弦的座標位置。另外，在前面分析了正確調性之後，調性表現定義決定了大小調的寫法，例如大調以大寫英文字母 M 表示，而小調則為小寫英文字母 m。而判斷小節線則使用節拍歸納模組執行後的結果，在時間軸上切割小節，並在此步驟決定了小節線的座標位置；決定小節線的座標之後，定義過字型的音符接著決定其座標位置；最後，和絃依照流行樂上常用的註記符號，定義和絃的輸出型態，同樣的，在此步驟決定和絃的座標位置。

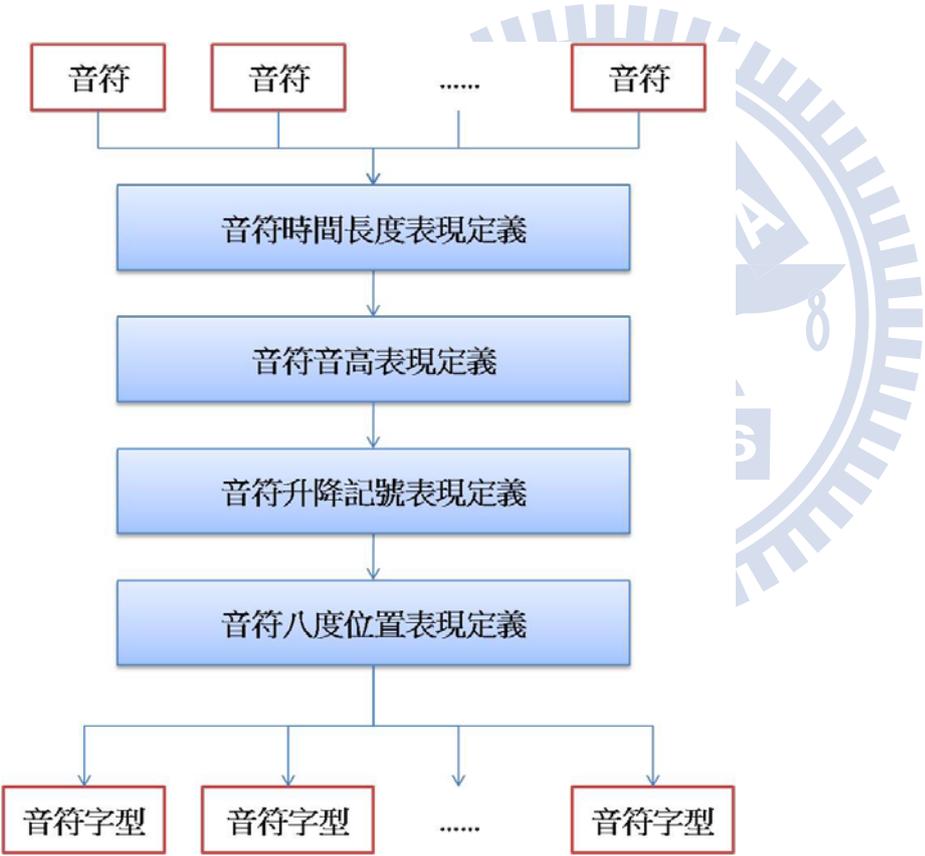


圖 5.6 定義音符輸出格式流程

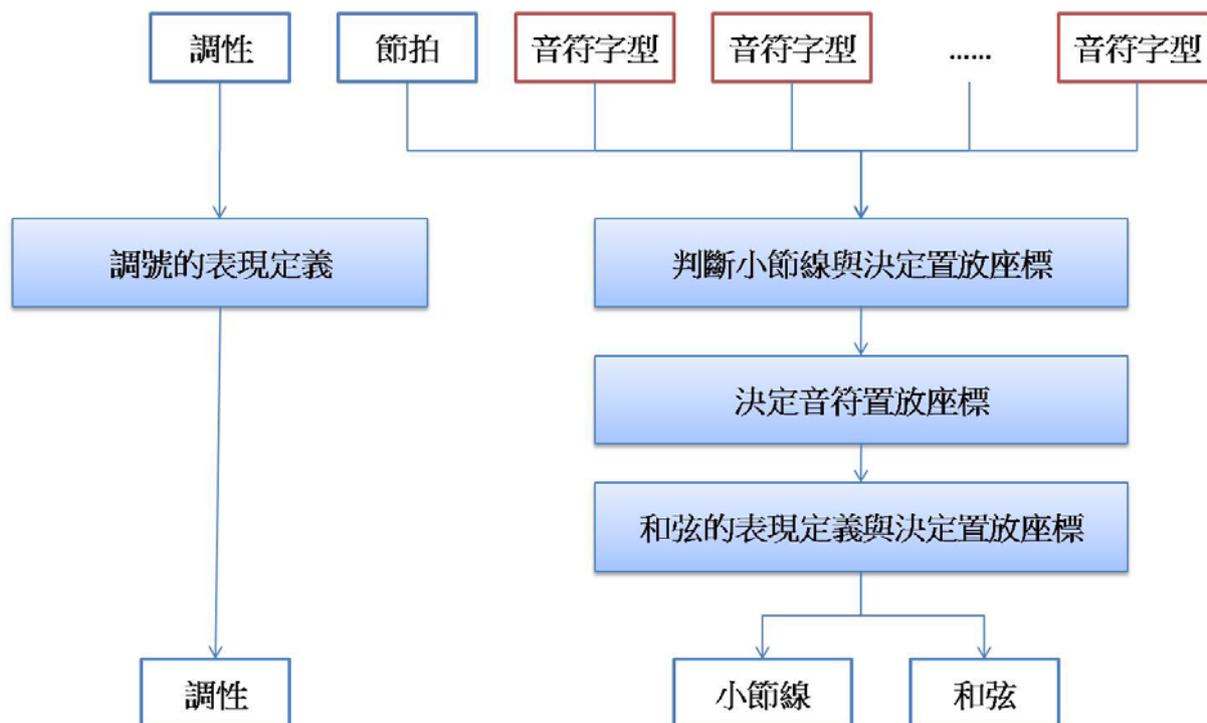


圖 5.7 音樂架構規則流程圖

調性分析和音名拼寫模組流程與演算法請見第三章，拍追蹤與節拍歸納模組流程與方法、和弦分析模組流程與演算法請見第四章。

第六章 實作結果

6.1 使用者介面

由圖 5.3 的模組與人機互動的流程圖看，人機介面由使用者選曲的主要有、選擇絕對或是相對音高、調整小節寬度等按鈕。而圖 6.1 為簡譜系統主要的人機介面設計，大致分為四個面板：

1. 最上方的控制面板：從左至右依序為，讀取 MIDI 檔、播放 MIDI 檔、輸出數字簡譜和離開系統四個按鈕。
2. 主畫面中左上的條件控制面板：從上到下依序為，絕對音高或相對音高的選擇鈕以及調整小節寬度的滑動桿。
3. 主畫面中左下的資訊面板：主要是呈現樂曲的資訊輸出面板，包括調性、節拍、MIDI 音軌數。
4. 主畫面中右邊的輸出面板：主要是呈現數字簡譜的輸出面板，包括小節線、小節數、音符、結合線、和弦。

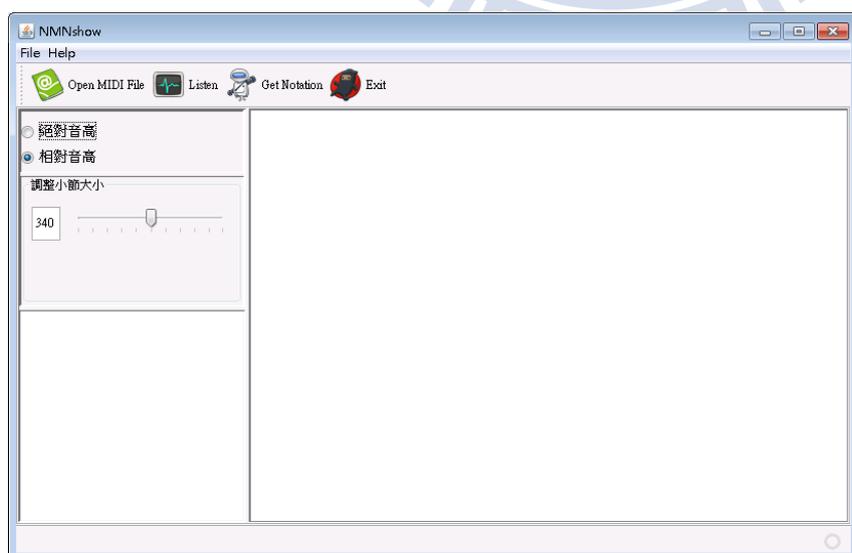


圖 6.1 系統介面

以搖籃曲為例，圖 6.2 即為系統 Demo 樂譜的輸出結果，樂曲資訊部分顯示此樂曲的調性為 C 大調，節拍為 3/4 拍子，MIDI 檔案的音軌數為 1。樂譜顯示部分有以相對音高表現的數字音高，固定小節寬度的小節線、小節線上顯示小節編號、以及有變化的和弦標示。而圖 6.3 為以 Come On Over 單一音軌版本的 MIDI 檔 Demo 例子，與上個例子不同的是，此首音樂為 Eb 大調，節拍為 2/4 拍子，很明顯的，這裡調性為非 C 為主音的大調，所以從相對音高看來主音表示為 1，但事實上是 Eb 這個音。當然，若是選擇以絕對音高表現樂譜的話，則主音 Eb 在簡譜上會標示為 b3。而圖 6.4 則是以 Come On Over 多個音軌版本的 MIDI 檔 Demo，每個音軌的排列方式為從左到右，上下小節對齊。

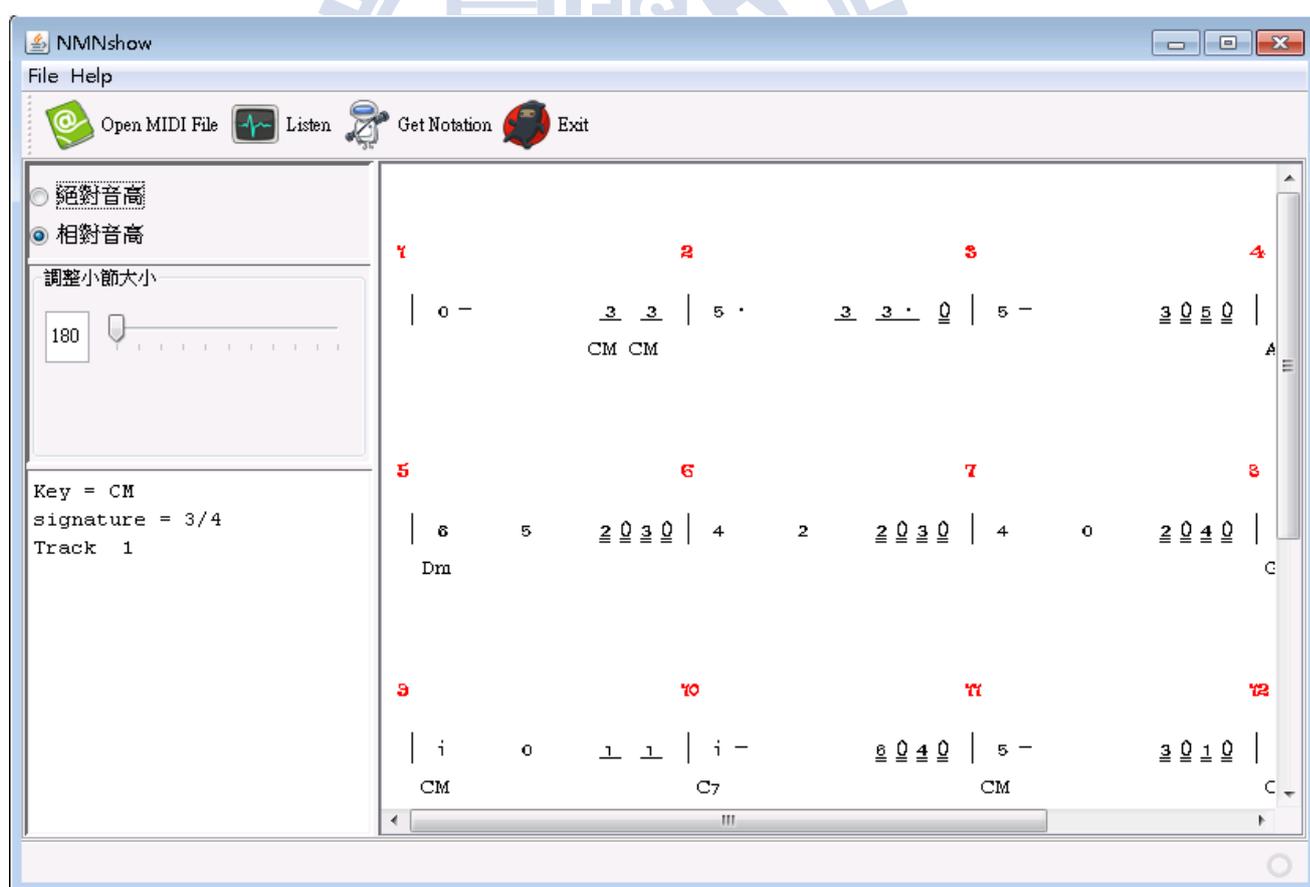


圖 6.2 系統 Demo”搖籃曲”樂譜畫面

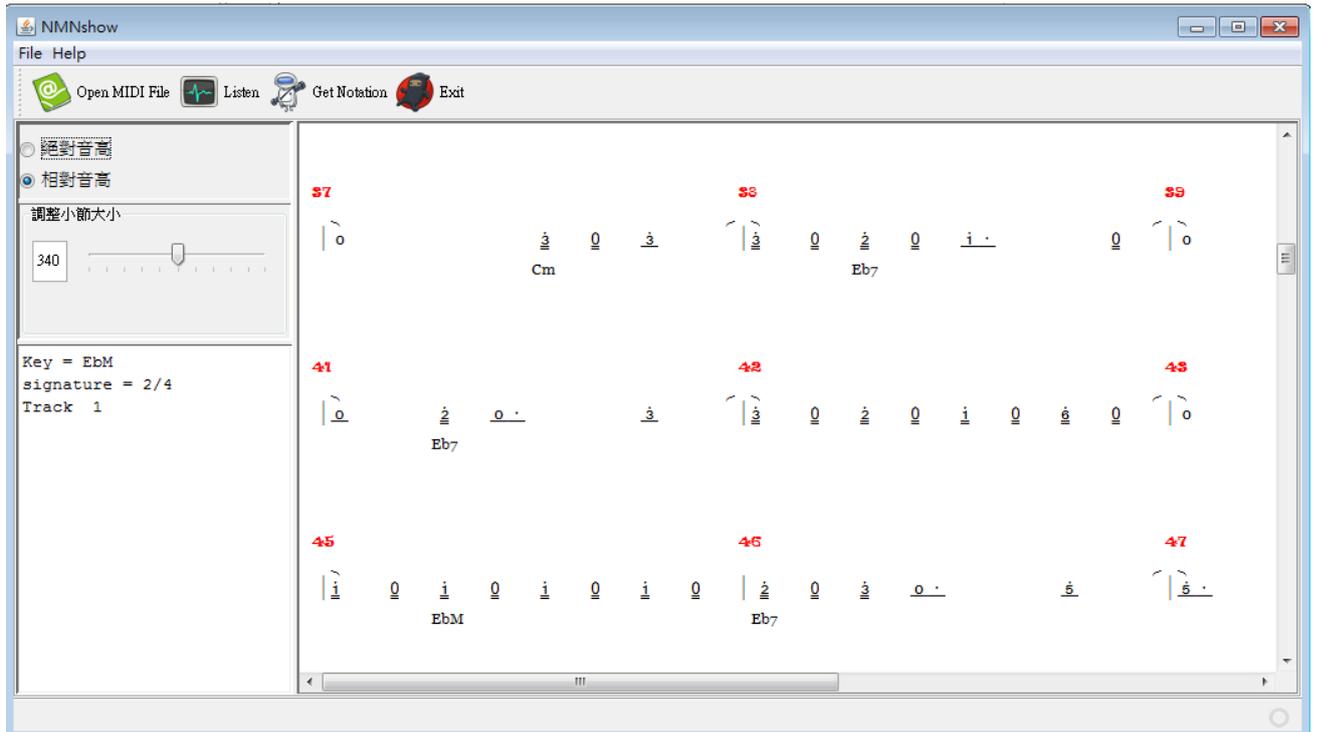


圖 6.3 系統 Demo” Come On Over”單一音軌樂譜畫面

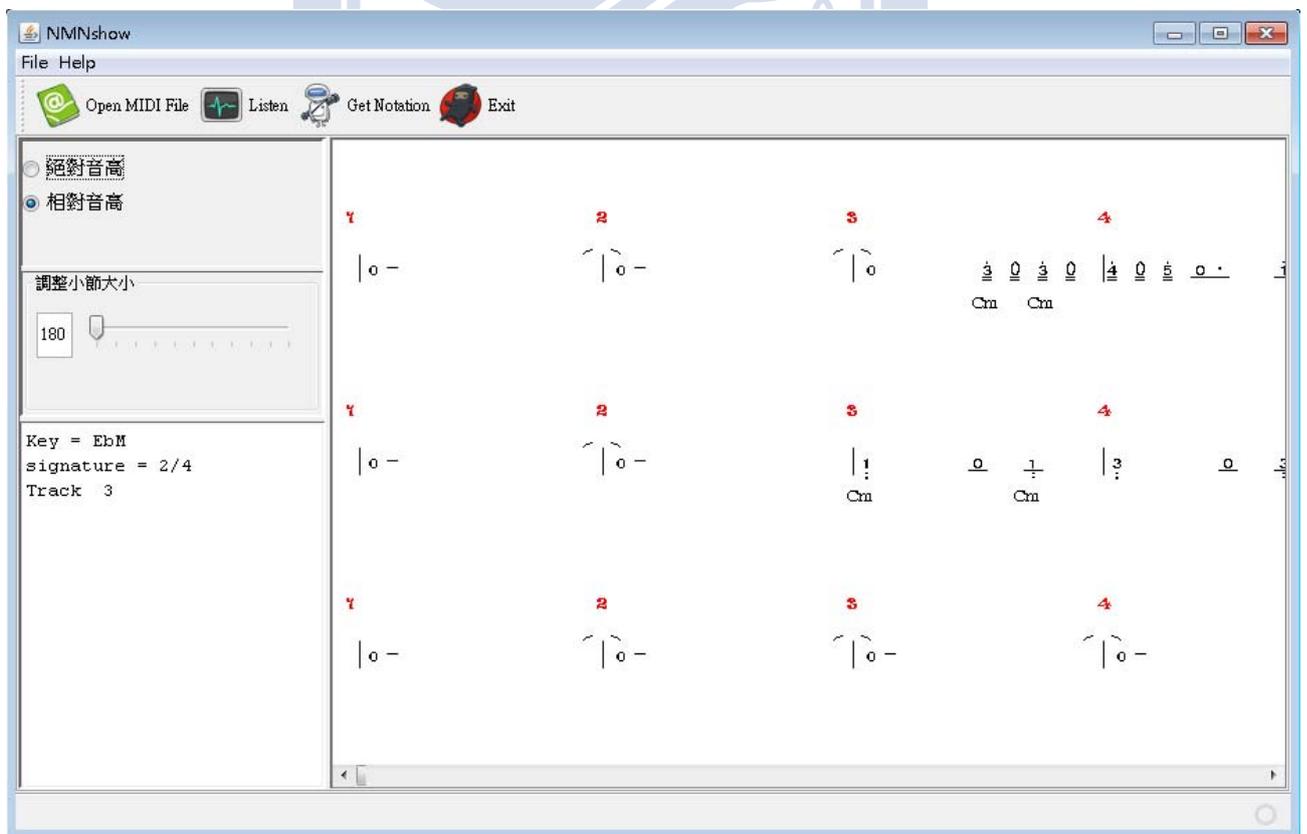


圖 6.4 系統 Demo” Come On Over”多音軌樂譜畫面

6.2 比較其他數字簡譜系統

圖 6.5 為商業軟體”CuteMIDI 簡譜作曲家” 與圖 6.4 本文所提之演算法處理相同的 MIDI 檔案，在主畫面中，左邊有顯示其調性，因 MIDI 檔案沒有記載此樂曲的調性所以此軟體會以預設的 C 大調為主，而所有升降記號皆為臨時記號表示。

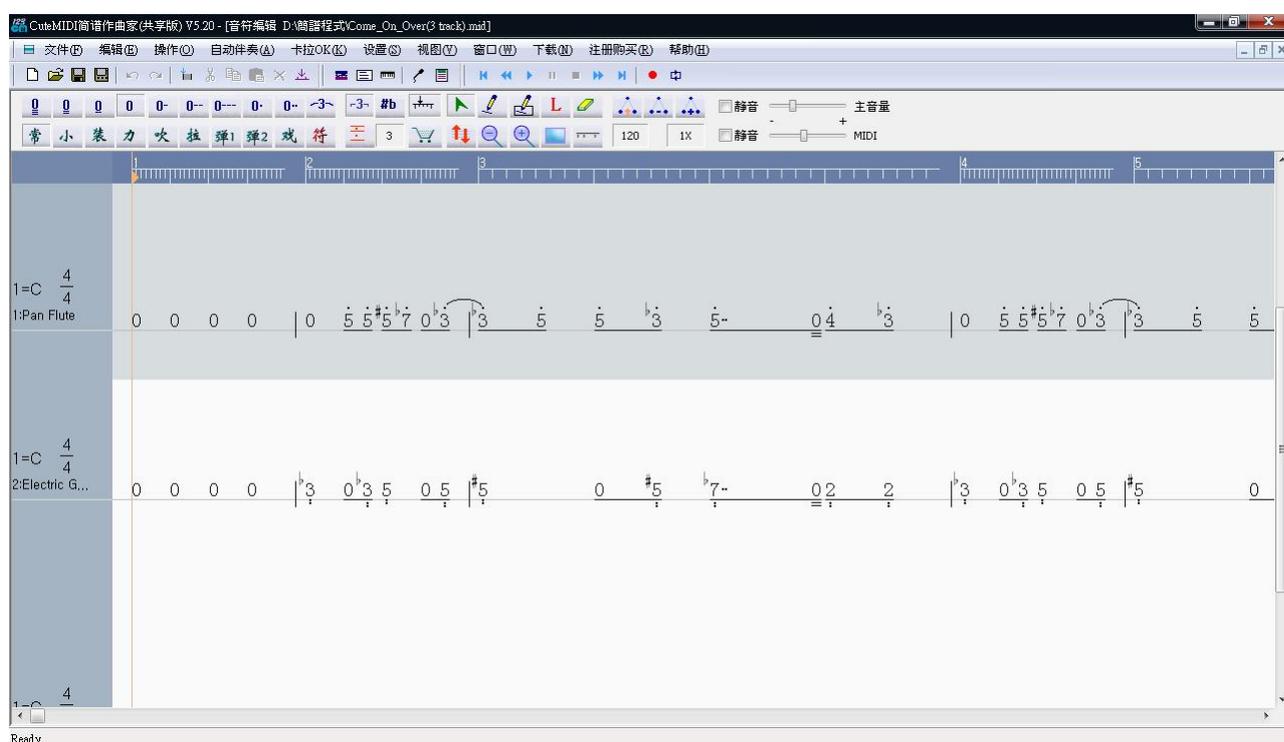
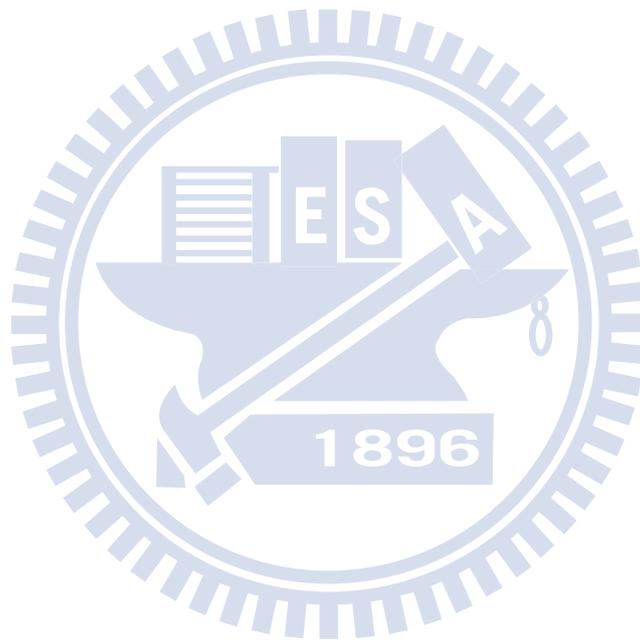


圖 6.5 CuteMIDI 簡譜作曲家 Demo” Come On Over”多音軌樂譜畫面

由此可見，本系統比 CuteMIDI 多增加了：

1. 最佳調性判斷：分析調性除了有能給予正確的相對音高表現，主要是可以讓使用者了解真正的調性，而不是以預設的 C 調表現
2. 即時音名拼寫：CuteMIDI 判斷升降記號從已建立的表，而我們可以即時的判斷音高的升降記號不需建表。

3. 和弦組合標示：每一個小節可以標示出參考的和弦，讓使用者可以更容易自己加上伴奏或和聲。
4. 單軌或多軌的判斷：CuteMIDI 預設最多 16 音軌，若 MIDI 檔案沒有這麼多音軌，很容易讓使用者混淆總音軌數。而本系統可以判斷有幾個音軌就產生幾行旋律，若為單音軌則不以從左到右一直延伸下去的顯示，而是以一行四小節為主的頁面顯示。



第七章 結論與未來工作

本系統已建立了擁有下列幾項分析功能的數字簡譜軟體：

- 1.音名拼寫
- 2.調性分析
- 3.拍子追蹤與節奏的歸納
- 4.和絃分析
- 5.基本音樂理論規則

透過上面的分析技術與規則，系統使用者可以從輸入一個 MIDI 檔案即可得到一個容易閱讀的相對音高樂譜呈現；另外對於未知節拍音樂也可預測，尤其對於創作音樂的人只需要錄下直覺的旋律即可得到合理的節拍分析；最後和弦分析的呈現提升使用者對於音樂伴奏的方便性。因此，此系統除了音符閱讀的便利性之外，對於創作音樂也提供了一個快速的路徑。

此外，雖然這個系統整合的技術不全是準確率最高或是最有效率的方法，並且系統的功能只達到基礎的分析功能，但未來若希望能更進一步改進此系統，除了維持原有的閱讀樂譜的便利性外，仍有更多分析追蹤技術可以和目前架構相結合。如此一來，就可以讓使用者有更多細緻的操作功能，更多提升音樂創作效率的空間。

參考資料

- [1] H. C. Longuet-Higgins and M. J. Steedman. On interpreting bach. *Machine Intelligence*, 6, 221-241, 1971.
- [2] Gabriellsson, A. Studies in rhythm. *Acta Universitatis Upsaliensis*, 7, 3-19, 1973.
- [3] C. L. Krumhansl and E. J. Kessler. Tracing the dynamic changes in perceived tonal organisation in a spatial representation of musical keys. *Psychological Review*, 89(4):334-368, 1982.
- [4] Povel, D.-J., & Essens, P. Perception of temporal patterns. *Music Perception*, 2, 411-40, 1985.
- [5] Carol L. Krumhansl. *Cognitive Foundations of Musical Pitch*. Oxford University Press, New York, NY, 1990.
- [6] Holger H. Hoos, Keith Hamel, Kai Renz, Jurgen Kilian; The GUIDO Notation Format - A Novel Approach for Adequately Representing Score-Level Music; in: Proc. of ICMC-98, 1998.
- [7] Kai Renz, Holger H. Hoos; A WEB-based Approach to Music Notation using GUIDO; in: Proc. of ICMC-98, 1998.
- [8] E. Chew. *Towards a Mathematical Model of Tonality*. PhD thesis, MIT, Cambridge, MA, 2000.
- [9] Cambouropoulos E., From MIDI to Traditional Musical Notation. In *Proceedings of the AAAI Workshop on Artificial Intelligence and Music: Towards Formal Models for Composition, Performance and Analysis 30 July - 3 Aug 2000, Austin, Texas*.
- [10] Cambouropoulos E., Automatic pitch spelling: From numbers to sharps and flats. In *Proceedings of the VIII Brazilian symposium on Computer Music, Fortaleza, Brazil, 2001*.

- [11] D. Temperley, *The Cognition Of Basic Musical Structures*: MIT Press, 2001.
- [12] Rowe R., *Machine musicianship*. Cambridge, MA: MIT Press, 2001.
- [13] Jacques Steyn; *Framework for a Music Markup Language*; First International MAX Conference, Milan (Italy), 2002.
- [14] Cambouropoulos E., *Pitch Spelling: A Computational Model*. *Music Perception* 20(4):411-430, 2003.
- [15] E. Chew and Y.-C. Chen, "Mapping MIDI to the spiral array: disambiguating pitch spellings," in *Proceedings of the 8th INFORMS Computing Society Conference (ICS '03)*, pp. 259–275, Chandler, Ariz, USA, January 2003.
- [16] E. Chew and Y.-C. Chen, "Determining Context-Defining Windows: Pitch Spelling using the Spiral Array", *Proceedings of the Fourth International Conference on Music Information Retrieval*, Baltimore, Maryland, USA, 2003.
- [17] Jacques Steyn; *Introducing Music Space*; 4th Open Workshop of MUSICNETWORK: *Integration of Music in Multimedia Applications*, 2004.
- [18] E. Chew and Y.-C. Chen, "Real-Time Pitch Spelling Using the Spiral Array," *Computer Music Journal*, vol. 29, pp.61-76, 2005.
- [19] D. Meredith, *The ps13 pitch spelling algorithm*. *Journal of New Music Research*, 35(2), 121-159, 2006.
- [20] David Meredith, *Optimizing Chew and Chen's Pitch-Spelling Algorithm*, *Computer Music Journal*, v.31 n.2, p.54-72, Summer 2007.

附錄一

簡譜輸入按鍵表 (By KAM) 2004/02/27

~`	! i	@ 2	# 3	\$ 4	% 5	^ 6	& 7	* :	(^) ^	- -	+ -
` o	1 1	2 2	3 3	4 4	5 5	6 6	7 7	8 .	9 ^	0 ^	- - = -

說明 1：

9 0 - 是音符上方之連音線，= 是四、八分音符上方較長之連線。

() ^ 音符上方有高音點時位置較高之連音線，+ 是四、八分音符上方較長之連線。

8 是四分音符低一點；* 是四分音符低兩點；~ 是四分音符高兩點。

Q i	W 2	E 3	R 4	T 5	Y 6	U 7	I .	O :	P #	{ ^ }	{ ^ }
q 1	w 2	e 3	r 4	T 5	y 6	u 7	i .	o .	P o	[]

說明 2：輸入八分音符低八度或低二個八度，先輸入沒有高低音的音符，

再於音符的右邊輸入 i (i) 或 I (I)。

下面的十六分音符和三十二分的低八度輸入方法同上。

大寫 O 是十六、三十二分音符的高兩點。

A i	S 2	D 3	F 4	G 5	H 6	J 7	K .	L ^	: b	
a 1	s 2	d 3	f 4	g 5	h 6	j 7	K .	l .	: 0	

Z i	X 2	C 3	V 4	B 5	N 6	M 7	< .	>	?	
z 1	x 2	c 3	v 4	b 5	n 6	m 7	, .	. .	/ -	\

附錄二

三和弦流行樂和弦名對照表(例子：根音為 C)

名稱	音程結構		流行樂和弦名	古典樂和弦名
	根音與第三音	根音與第五音		
大三和弦	大三度	完全五度	CM	I
小三和弦	小三度	完全五度	Cm	i
增三和弦	大三度	增五度	Caug	I ⁺
減三和弦	小三度	減五度	Cdim	i ^o

七和弦流行樂和弦名對照表(例子：根音為 C)

名稱	音程結構			流行樂和弦名
	根音與第三音	根音與第五音	根音與第七音	
屬七和弦	大三度	完全五度	小七度	C ₇
大七和弦	大三度	完全五度	大七度	CM ₇
小七和弦	小三度	完全五度	小七度	Cm ₇
小大七和弦	小三度	完全五度	大七度	CmM ₇
半減七和弦	小三度	減五度	小七度	Chalf-dim ₇
減七和弦	小三度	減五度	減七度	C ^o ₇