

利用實數數值建構音樂料庫之特徵索引

羅有隆
朝陽科技大學
資訊管理系
副教授
e-mail :
yllo@cyut.edu.tw

蔡怜怡
朝陽科技大學
資訊管理系
研究生
e-mail :
s9614634@cyut.edu.tw

摘要

近年來，大部份的音樂資料擷取研究領域，都是利用特徵的萃取，例如：旋律、節拍、和弦等等，然後利用這些特徵建立索引，以加速對音樂資料的搜尋。許多研究指出，可以將音樂資料特徵轉換成字串或數值，然後建立字串或數值索引，以幫助音樂資料的搜尋。而現有字串索引（例如：suffix tree），缺乏擴充性。此外，研究則較少著重於音樂數值索引上，且現存的技術，是將一個特定長度的音樂片段轉換成整數，再套用於現有的數值索引結構（例如：R-tree、B-tree）。然而這個方法，查詢的長度必須符合特定長度的音樂資料轉換，否則將降低查詢的處理效率。在這篇研究報告中，我們提出一個實數轉換函式，它不需要特定長度之音樂片段，而且可讓查詢長度更有彈性。最後，我們的實驗結果證明，此新方法優於現有的索引結構，在音樂檢索上更有效率。

關鍵詞：多媒體資料庫、音樂資料庫、內容擷取、字串索引、數值索引

Abstract

In the most of current works, the researchers extract the features, such as melodies, rhythms and chords, from the music data and develop indices that will help to retrieve the relevant music quickly. Several reports have pointed out that these features of music data can be transformed and represented in the forms of music feature strings or numeric values such that string indexing or numeric indexing is created, respectively, for music retrieval. For string indexing, there is only limited index structure (ex. suffix tree) suitable for music retrieval and it is lack of scalability. Moreover, for numeric indexing, there is only few research emphasized on this issue. The existing approaches all transform a specific length of music segments (features) into integers such that various numeric index structures can be applied (ex. R-tree, B-tree). In this approach, however, the length of

query (query by example) is required to match the specific length of transformation of music data otherwise it will harm the efficiency of query processing. To address these problems, in this paper, we will present a real value transformation function for without specific length of music segment and for more flexible of query length. Our experimental results also show that the new approach outperforms existing index schemes.

1. 前言

由於在這多媒體應用的爆炸性時代中，有愈來愈多的非符號系統資料需要立即被執行處理。而這些資料，不同於傳統的數值型態或文字型態資料，它包含圖片、聲音、影像、文件等等。為了這些新資料型態，許多的研究報告指出，是集中在以內容為主的檢索技術去做聲音、影像和文件的處理[5][6][7][8][12][15]。近年來，在音樂資料數位時代的快速發展下，如何有效的管理音樂資料是愈來愈受到關注，而且愈來愈多的研究顯示，在做大量音樂擷取技術是愈來愈引人注意，例如：Chen 等人的 the Query by Rhythm[3]、Chen 等人的 Query by Music Segments[4]、Lee 等人的 Multi-Feature Index Structures[14]和 Lo 等人[21][22]、Liu 等人的 Non-Trivial Repeating Pattern Discovering for music data[16]和 Lo 等人[23]、Uitdenbogerd 等人的 Melodic Matching Techniques [30]、Liu 等人的 Approximate Melody Matching [17][33]、Tseng 的 Key Melody Extraction and N-note Indexing[29]、Lo 等人的 Numeric Indexing for Music Data[18][19]還有其他的研究[1][10][15][18][20][23][26][28]。

在音樂資料擷取的研究領域中，以內容為主的擷取技術，都是利用音樂資料的特徵做萃取，例如：主旋律(key melodies)、節拍(rhythm)和合弦(chords)，並且將這些特徵建立索引，使得加速在相關音樂資料上的搜尋[13][16][20][23]。一些報告也指出，這些音樂

特徵可以做轉換或對應成字串 [3][4][10][14][15][23][29]、數值型式 [18][19]，並利用這些特徵建立索引而幫助音樂資料的檢索。然而，使用在音樂檢索上的字串索引是個有長度限制的結構索引（例如：suffix tree [24][31][32]），因為這個限制，使得字串索引會有擴充性缺乏的問題。另外，現今使用在音樂檢索上的數值索引技術是較少被探討的。現有的方法是轉換一個特定長度的音樂片段（特徵字串值）為整數值，例如：R-tree [9]。而在這個方法上，長度查詢（範例查詢）是必需要符合音樂資料特定的長度，否則會降低音樂資料搜尋處理的效率。

在此報告中，我們將提出一個以實數數值為索引結構的音樂資料檢索技術，而此技術是利用轉換函式去導出使用一個音樂片段轉換為一個實數數值。在這個轉換的音樂片段長度是不需做特定的限制，因此，我們新的研究方法對於搜尋的長度是更有彈性的，如此，查詢的處理也更有效率。

接下來，本文將整理為以下的幾個章節：第二章我們會簡述回顧現有的音樂索引結構，並且介紹在音樂資料檢索中的字串索引和數值索引；接著在第三章的部份，則為我們所提出的實數數值音樂索引技術，然後詳述索引技術的搜尋方法。而第四章的部份，是此技術的實驗數據結果。最後，則為結論和未來研究方法的敘述。

2. 文獻探討

近年來已有愈來愈多的探討是在音樂資料庫檢索議題上。在這章節中，我們介紹在音樂資料檢索中的字串索引和數值索引，並且簡述回顧現有的音樂索引結構。

2.1 音樂檢索的字串索引

Suffix Tree 是一種樹型態的索引結構，它代表所有的字尾字串符號，而且它解決了字串比對上的問題 [24][31][32]。簡單的來說，在建構 Suffix Tree 有以下幾點特色：

- (1) 建構一個 suffix tree 是以字串長度為 m 字元建制而成，葉節點也剛好為 m 個，故這些葉節點的編號是為 1 到 m 。
- (2) 任何兩個非葉節點分支出去的分支，在標示的部份是不可以為相同字元。
- (3) 葉節點的編號，剛好指出根節點到葉節點所代表的子字串，也就是說，為原始字串的起始位置。

近來，Suffix Tree 也被用來作為音樂特徵字串索引，以字串索引去幫助音樂資料庫的搜尋。假設音樂資料包含了 n 個音符，例如：Do、Re、Mi、…等等。並且分別將每個音符對映成各別的音樂符號 ‘a’、‘b’、‘c’、…等等。因此，這些音樂符號的一個旋律可以代表成一個音樂字串 [10][16][17][20][23][29]。舉個例子說明，假如現在有一個音樂字串為 “ababc”，那這個字串則分別拆成五個音樂旋律片段 “ababc”、“babc”、“abc”、“bc”、“c”，其所建構出來的 Suffix Tree 索引為圖 1。因此，在這個 Suffix Tree 索引技術上是可以於音樂資料檢索中完成字串比對的問題。

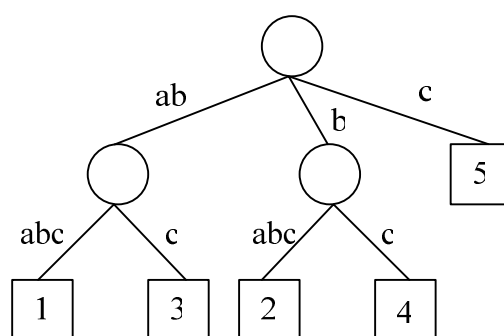


圖 1、音樂字串 “ababc” 的 Suffix Tree

另外，在音樂裡的旋律如果太多，可能會使索引在建構時造成太長的局面，因此，有幾個研究指出，在一首音樂中的主旋律 (themes) 或重複片段是簡短的，而且，是令人熟悉的 [1][10][13][16][20][23][25][27]。所以，我們可以使用主旋律或重複片段去建構索引，使得在音樂檢索上的查詢執行更加的快速。

2.2 音樂檢索的數值索引

在 2000 年，Jagadish 等人發表出 numeric mapping，這個研究是將字串對應成數值 [11]，而他們使用 R-tree 和 B-tree 結構去做數值索引的建構，然後利用數值轉換的方式去幫助字串的比對搜尋。後來，Lo 等人提出一個用於音樂資料的整數數值索引技術 [18][19]，這個研究是嘗試著把音樂資料對應成一個數值，再利用 R-tree 和其他的數值索引結構建立成該整數數值的音樂資料索引。故首先我們將音樂資料轉換成數值，讓這些音樂符號假設為 ‘a’、‘b’、‘c’、…、‘m’，並且各別將音樂符號對映成 0、1、2、…、 $m-1$ 的整數數值。如果我們每一次所選擇的是為 n 個連續音樂旋律特徵音符，則此 n 個音樂符號為 $x_1、x_2、\dots$ 、

x_i ，故於每個音符所對應的數值公式為 $P(x_i)$ ，而 x 的範圍限制是為 $1 \leq i \leq n$ 。因此，可以利用數值轉換函式 $v(n)$ ，將連續的 n 個音符轉換為整數數值。

$$v(n) = \sum_{x=1}^n P(x) \times m^{x-1} \dots (1)$$

我們以一個範例來說明，假設音樂符號種類的代表為 'a'、'b'、'c'、...、'j'，而且每個符號所對映的各別整數數值為 '0'、'1'、'2'、...、'9'。如果有一個音樂片段為 "bcdab" 的 4 個音樂符號，故利用數值轉換函式 $v(n)$ ，則可以得到 $v(n) = 1 \times 10^0 + 2 \times 10^1 + 3 \times 10^2 + 1 \times 10^3 = 1321$ 。然後，再將這個數值 1321 插入於 R-tree 的索引結構中，以此 R-tree 索引結構做為音樂資料的搜尋比對。此外，這個技術是在音樂片段索引中以同一長度 (n) 做查詢。而在做查詢時，是將音樂片段轉換為整數值，然後再以相同的方法去做索引上的搜尋。

3. 音樂資料的實數數值索引

現有的數值索引技術，都是為查詢長度需符合轉換音樂資料特定的長度，否則將會降低做查詢執行的效率問題。我們舉例說明，假設所有的音樂片段所建立而成的數值索引是由連續音符之長度為 4 和長度為 5 (例如："bcdab")，在查詢部份應該分為 2 個音樂片段 (例如："bcdab" 和 "cdab")，然後將這兩個音樂片段轉換成數值，以做索引資料的搜尋；另外，再把查詢結果尋找到的最終樂曲做交集的動作，而故此交集的動作，將使查詢執行的效率降低。關於論及這個問題，我們提出一個在音樂資料中以實數數值為索引的技術。

3.1 實數數值的音樂索引之轉換函式

在這一章節，介紹我們提出一個以音樂特徵轉換成為一個實數數值的轉換函式。假定 s 是一個音樂片段，其長度以 $length(s)$ 做為表示，並且假設在音樂中有 m 種不同的音符種類，轉換的函式為 $v_{real}(s)$ ，則我們設計的轉換實數數值函式如下：

$$v_{real}(s) = \sum_{x=1}^{length(s)} P(x) \cdot (m+1)^{y-x} \dots (2)$$

在這個函式裡， $P(x)$ 是為類似整數數值所定義的函式 (1)，然而，原本整數數值開始是

從 1、2、...、 m ，不過現在此方法的每個音符是以 0、1、...、 $m-1$ 所代替。因為一個實數數值有兩個部份，第一部份是：整數、第二部份是：小數，那在小數部份將以 0 代表為一個結束符號。如此，假設小數部份是停止在非零的小數數位時，那轉換後之最後小數位數的任何音樂是不為 0。再者， y 所代表的是為整數位數，而這個對於小數位數的 y 部份是可以降低溢位的問題。

舉個例子說明，假設那個音樂音符種類為 9 種，且對映到的整數數值為 '1'、'2'、'3'、...、'9'。此時，如果以 3 個音符做轉換成整數部份 ($y=3$)，而一個音樂片段為 "abcdab"，故利用實數數值轉換函式，則可以得到： $v_{real}(\text{"abcdab"}) = 1 \times 10^2 + 2 \times 10^1 + 3 \times 10^0 + 4 \times 10^{-1} + 1 \times 10^{-2} + 2 \times 10^{-3} = 123.412$

3.2 實數數值索引的建構

範例查詢 (Query By Example, QBE) 的方式，是已被廣泛地使用在以多媒體資料內容為主的檢索技術裡。而使用者可以以一個音樂片段做為一個查詢例子，以這個搜尋例子可能會被完全的找出或搜尋到包含於某些相關的音樂樂曲。因此，一個音樂片段和最後字尾應被設於這個索引中，以使得在所有可能的查詢結果裡成功的搜尋到。這一節中，我們用一個例子來做說明是如何以實數數值音樂特徵字串值建構 R-tree 索引。為了簡單化，我們只假設了 9 種音樂符號 'a' ~ 'i'，並且於每種音樂符號各別的對應成整數數值為 '1' ~ '9'。因此，從數值轉換的函式 (2) 可以看成一個十進制的型態，因為，十進制的型態是較於大眾所容易理解的。假設現在有兩首音樂片段是為兩隻老虎 (S1: Do Re Mi Do Do Re Mi Do) 和小蜜蜂 (S2: Sol Mi Mi Fa Re Re)，故此兩首樂曲的片段即表示為 "abcaabca" (S1) 和 "eccdbb" (S2)。如果以 4 做為轉換的整數部份 ($y=4$)，故利用實數數值轉換函式 $v_{real}(s)$ ，以此兩首音樂片段 S1 和 S2 做換算數值於表 1 和表 2 所示。接著，我們用兩隻老虎這首音樂片段所轉換出來的實數數值建於 R-tree 索引結構中，其所呈現的索引結果於圖 2。而在這個圖 2 的 R-tree 索引中，每個節點有三個分支度，並且在這三個非葉節點部份，為子樹的下限和上限之特定鏈結。對於音樂片段的音樂資訊和儲存位置的字尾值也表示在葉節點的 R-tree 中。如果，所轉換的實數數值為一個以上 (可能為一個或更多個音樂片段) 相同值的

話，只需利用一個鏈結儲存於這音樂資訊。接著，我們再繼續以表 2 的這首小蜜蜂之音樂片段裡的實數數值插入兩隻老虎的 R-tree 索引結構裡，其所呈現的結果於圖 3 所示。此音樂資料庫的索引，是可以相同方式去建構而成的。此外，對於音樂資料的實數數值之搜尋方法，我們將於下章節繼續做詳述。

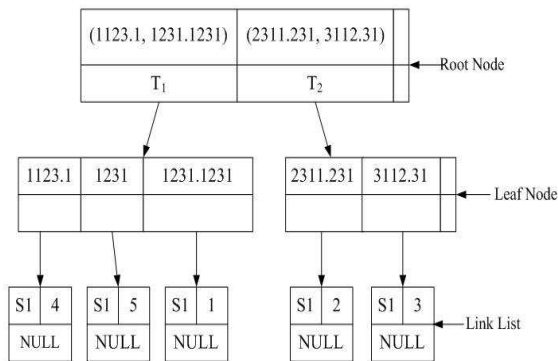


圖 2、兩隻老虎的 R-tree 索引

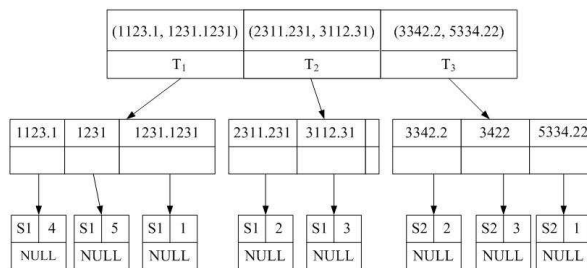


圖 3、兩隻老虎和小蜜蜂的 R-tree 索引

表 1、在兩隻老虎的音樂片段 (S1) 中轉換成實數數值

主旋律	音樂片段	換算過程	實數數值
Abcaabca	abcaabca	$1*10^3+2*10^2+3*10^1+1*10^0+1*10^{-1}+2*10^{-2}+3*10^{-3}+1*10^{-4}$	1231.1231
	bcaabca	$2*10^3+3*10^2+1*10^1+1*10^0+2*10^{-1}+3*10^{-2}+1*10^{-3}$	2311.231
	caabca	$3*10^3+1*10^2+1*10^1+2*10^0+3*10^{-1}+1*10^{-2}$	3112.31
	aabca	$1*10^3+1*10^2+2*10^1+3*10^0+1*10^{-1}$	1123.1
	abca	$1*10^3+2*10^2+3*10^1+1*10^0$	1231

表 2、在小蜜蜂的音樂片段 (S2) 中轉換成實數數值

主旋律	音樂片段	換算過程	實數數值
Eccdbb	Eccdbb	$5*10^3+3*10^2+3*10^1+4*10^0+2*10^{-1}+2*10^{-2}$	5334.22
	Ccdbb	$3*10^3+3*10^2+4*10^1+2*10^0+2*10^{-1}$	3342.2
	Cdbb	$3*10^3+4*10^2+2*10^1+2*10^0$	3422

3.3 於音樂的實數數值索引做搜尋

在我們提出的音樂資料實數數值索引做搜尋的執行，而查詢片段應為是轉換成實數數值的函式 (2)，然後於這個音樂索引中去比對可能的音樂樂曲，並且，我們考慮了四個比對的條件。而且假設 *Qvalue* 是從查詢片段的實數數值所轉換而成。以下我們詳述了四個比對條件：

條件 1: 以在索引樹數值中做 *Qvalue* 的完全比對。最後會有一個或多個樂曲，是符合我們所要找的音樂。利用一個例子來說明，如第 3.2 節中的敘述，假設一個查詢片段是“caabca”，其轉換成的實數數值為 3112.31，那於圖 3 的索引中可以被找出，並且此音樂片段則為音樂樂曲 S1 中的第 3 個音開始。

條件 2: 假設 $Qvalue$ 為 k 個位數 (查詢長度是 k 個音符)。如果 $Qvalue$ 的位數比索引中某些值還小, 則 $Qvalue$ 的第 k 位數就會完全符合的被搜尋出; 在這種情況下, 所要搜尋之音樂樂曲就可能被找到。舉個例子說明: 如果有一個查詢片段為 “abcaa”, 它轉換後的實數數值 1231.1 會無法在圖 3 的索引中被發到。但是, 我們卻可以完全搜尋到和相同 $Qvalue$ 值的 1231.1231 前 5 個位數 (音符), 所以此音樂片段則為音樂樂曲 $S1$ 的第 1 個音開始。在這個條件可以由以下說明來做簡單的測試:

IF $(Target - Qvalue) < (m-1)^{y-length(query\ segment)}$
THEN found.

如果, 搜尋的樂曲是實數數值, 那在索引中就有可能把符合的樂曲給找尋出來, 換言之, $(1231.1231-1231.1)=0.0231 < 10^{-4.5}$, 此音樂樂曲就會被搜尋到。

條件 3: 如果這個尋找值是 k 位數的話, 那如果 $Qvalue$ 比索引中所要尋找到的值還大, 則 $Qvalue$ 的第 k 位數就會完全符合的被搜尋出來。在這種情況下, 所要搜尋之音樂樂曲就可能被找到。舉個例子說明: 如果有一個查詢片段為 “ccdbbab”, 它轉換後的實數數值 3342.212 會無法在圖 3 的索引中被發到。但是, 我們卻可以完全搜尋到前 5 個位數 (音符) 3342.2 的 $Qvalue$ 值, 不過由於這個查詢片段無法完全比對, 但是, 我們仍然可以幫助使用者找到最為可能的音樂樂曲。而這個條件也可以用以下說明來做簡單的測試:

IF $Target == \text{the first } length(Target) \text{ digits of } Qvalue$
THEN possible found.

也就是, 3342.2 等於 3342.212 的前 5 位數, 這個音樂樂曲就可能被搜尋出來。

條件 4: 假如, 最後都不為上述條件 (條件 1~條件 3) 的話, 那就是找不到符合我們要的音樂樂曲。

4. 效能研究

在這章節中, 為了驗證我們提出的實數數值索引技術, 以一系列的實驗去做效能的評

估。我們仿做[16]和[18]的研究去建立一個實驗用的音樂資料庫, 其中包含了兩個部份: 真實音樂資料和人工音樂資料。在真實的音樂資料中, 我們收集了 145 首的音樂樂曲, 並且計算了這 145 首音樂之每個音符所出現的次數。在圖 4 中為最常出現的 19 個音符, 而在這 19 個音符中分別是低音的 Mi~Si、中音的 Do~Si 和高音的 Do~Si。那個中音的 Sol (表示為 ‘j’) 是出現次數最高的音符。我們還發現, 此外觀曲線圖所呈現的是常態分配, 於是, 我們使用了這 19 個音符做隨機產生, 得到的人工音樂資料將做為我們實驗的音樂資料庫。

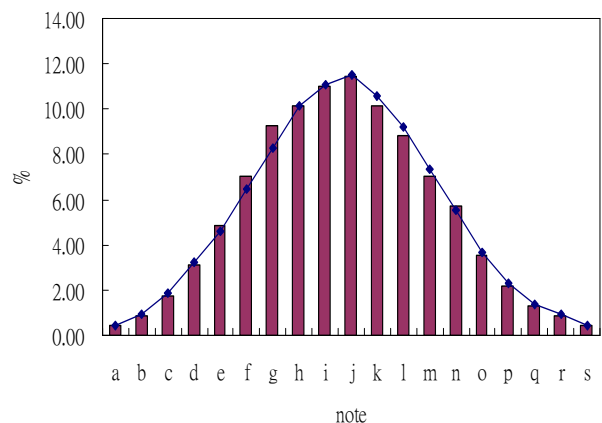


圖 4、最常出現的 19 個音符

由於在這個音樂資料庫檢索的研究裡, 通常在資料庫中是只儲存音樂片段或則儲存重複片段, 而以這兩種方式代替了完整的音樂旋律[10][17][20][23], 因此, 我們的實驗資料庫將是由音樂片段之平均長度為 20 所建構而成的。然後, 我們使用一台 AMD 的 Athlon 64X2 Dual Core 處理器和 2GB 的記憶體電腦做實驗模擬。在實驗中, 我們討論了以下兩個要素, 並且, 我們所使用的參數是列於表 3 中。

- (1) 查詢長度的影響
- (2) 轉換整數位數的影響

表 3、實驗參數

參數	數值範圍
音樂資料庫(音樂片段)	10,000
整數位數(y)	3~6
查詢長度(notes)	4~8
音符種類(m)	19
查詢的次數(QBE)	> 50,000

4.1 查詢長度的影響

我們認為有效率的查詢處理可能會受到查詢長度和索引技術的影響，所以在這節中，我們討論了在不同音樂索引的搜尋比對上，對查詢長度的影響力。由於整數數值音樂索引方法宣稱是優於字串音樂索引方法[18]，所以我們將利用實數數值索引方法，和這項數值索引研究做比較。

在這個實驗探討裡，我們使連續的音符(n)數轉換成整數為 4 的數值索引，和整數部位(y)為 4 的實數數值音樂索引。然後以範例查詢 (QBE) 的方法，讓查詢長度由 4 個變化到 8 個。還有我們對每種查詢長度至少做 50,000 次的測試，且其為了實驗的客觀性，以匯集所有的搜尋處理時間做平均的計算。再者，雖然我們的資料庫只有 10,000 首音樂片段，不過卻因為在以平均片段長度為 20 下，我們得到了 170,000 的字尾音樂片段，其中只剩下整數部份的 y ，是可以被置入於索引中去進行音樂的檢索。圖 5 為在這個實驗結果的數據圖，而數據圖上我們把整數數值音樂索引用「整數」做縮寫代表，把實數數值音樂索引用「實數」做縮寫代表。

在圖 5 中，我們發現整數數值音樂索引對於查詢片段的長度之曲線是波動較大，而對於整數數值音樂索引的平均回應時間是當我們增加查詢長度時，就會跟著高幅度的上升。但是，在實數數值音樂索引對查詢長度的增加，曲線卻是可以保持平穩的，由減少花費時間的 20.19% 至 87.15% 數據可知，此實數數值索引對於在範例查詢音樂是優於整數數值索引。這也可以解釋說，在整數數值音樂索引的研究中，查詢所需的長度和將音樂資料做特定長度的轉換必須完全一樣，否則將會降低查詢執行時間的效率問題[18]。如果查詢長度是比連續音符(n)數還長的話，應該先轉換至數值索引的整數值，當查詢時，應該分成幾個長度為 n 的音符子片段，然後將這些音符子片段做轉換，以做搜尋的動作。可是，我們所提出新的實數數值音樂索引研究，在查詢長度上是沒有這樣的限制，也就是為什麼，在這個實驗中，查詢長度為如果 4 的時候，其此兩項技術的平均回應時間是接近的，但是，當查詢長度增加時，平均回應時間就會愈差愈大。

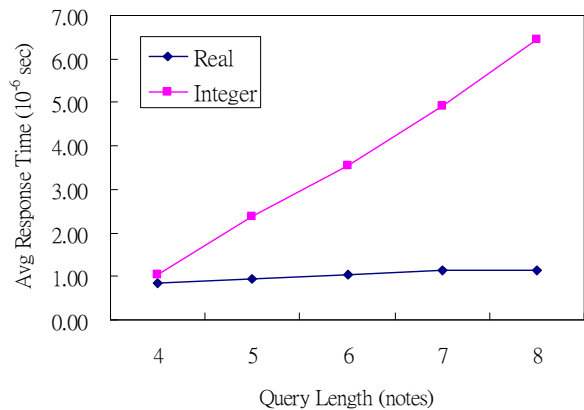


圖 5、查詢長度的分析

4.2 轉換整數位數的影響

接著，我們描述在實數數值音樂索引的整數部位可以降低小數部位溢位問題。在這節中，我們將在查詢處理上對音符(y)的整數位數做測試模擬影響。這個實驗中，我們將 y 假設為 3，然後增加到 6，而且此資料庫仍然是 10,000 首音樂片段，接著對於查詢片段的查詢長度，我們以 6 個音符和 8 個音符做調查。並且，我們對每種查詢長度至少做 50,000 的測試，也其為了實驗的客觀性，匯集所有的搜尋處理時間做平均的計算。圖 6 為我們這個實驗結果的數據圖。

在圖 6 中可得知，查詢長度為 6 時，所需的處理時間是較查詢長度為 8 的處理時間少，也就是說，當查詢長度愈長，平均回應處理時間就愈長。再者，由於這兩條曲線所呈現出來的是為水平線，故可以得知，對於整數部位的音符(y)數，是不會影響查詢處理時間。在這個部份我們有注意到，如果查詢的長度比 y 少的話，則查詢執行處理將在檢查比對條件 2 (第 3.3 節) 下會花費較多的時間找尋我們要的樂曲。所以，查詢長度比連續音符(n)數轉換成數值少時，這樣的情況也會發生在整數數值索引技術中。因此，除了在我們確定所有長度的查詢以外， y 為 3 或 4 時是最為合適滿足我們所建議的。

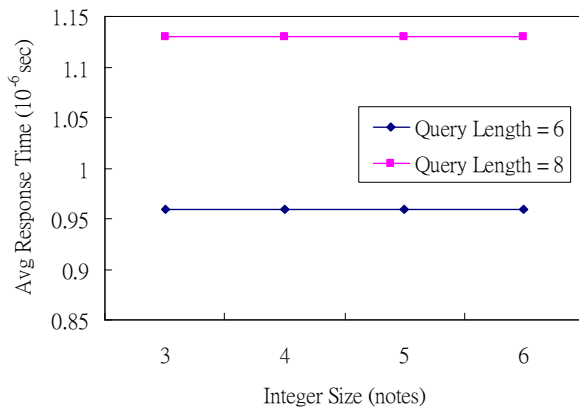


圖 6、整數數位查詢的分析

5. 結論與未來研究方向

此篇研究報告最主要的目的，是我們提出了一個音樂資料庫的實數數值索引技術，以改善音樂檢索的效能。我們以設計的轉換函式將音樂片段轉換成實數數值，然後利用這些實數數值去建構索引。而且，我們也分析了在實數數值音樂索引中，做為搜尋的比對條件。我們的音樂索引技術，對於查詢長度是更加為有彈性的，並且在查詢執行處理是更加的有效率。實驗結果也證明，實數數值音樂索引優於現有的整數數值音樂索引。未來我們也期望，可以將此音樂資料庫檢索技術加以改善，以適用在相同多媒體資料庫的應用上。

參考文獻

- [1] V. Bakmutova, V. D. Gusev, and T. N. Titkova, "The Search for Adaptations in Song Melodies," *Computer Music Journal*, Vol. 21, No. 1, pp.58-67, 1997.
- [2] S. Blackburn and D. DeRoure, "A Tool for Content-based Navigation of Music," *In Proc. Of ACM Multimedia*, pp. 361-368, 1998.
- [3] James C.C. Chen and Arbee L.P. Chen, "Query by Rhythm An Approach for Song Retrieval in Music Databases," *In Proc. Of Int'l Workshop on Research Issues in Data Engineering*, pp. 139-146, 1998.
- [4] Arbee L.P. Chen, M. Chang, J. Chen, J.L. Hsu, C.H. Hsu, and Spot Y.S. Hua, "Query by Music Segments: An Efficient Approach for Song Retrieval," *In Proc. of IEEE Int'l Conf. on Multimedia and Expro*, 2000.
- [5] G. Davenport, T.A. Smith, and N. Pincever, "Cinematic Primitives for Multimedia," *IEEE Computer Graphics & Applications*, pp. 67-74, July 1991.
- [6] Y. F. Day, S. Pagtas, M. Iino, A. Khokhar, and A. Ghafoor, "Object-Oriented Conceptual Modeling of Video Data," *In Proc. Of IEEE Data Engineering*, pp. 401-408, 1995.
- [7] E. A. El-Kwae and M. R. Kabuka, "Efficient Content-Based Indexing of Large Image Databases," *ACM Trans. On Information Systems*, Vol. 18, No. 2, pp.171-210, April 2000.
- [8] S-T. Goh and K-L. Tan, "MOSAIC: A Fast Multi-Feature Image Retrieval System," *Data & Knowledge Engineering* 33, pp. 219-239, 2000.
- [9] A. Guttman, "R-Trees A Dynamic Index Structure For Spatial Search," *Proceedings of the 1984 ACM SIGMOD international conference on Management of data*, pp. 47-57, 1984.
- [10] J. L. Hsu, C. C. Liu, and Arbee L.P. Chen, "Efficient Repeating Pattern Finding in Music Databases," *In Proc. of ACM Int'l Conf. on Information and Knowledge Management*, 1998.
- [11] H. V. Jagadish, N. Koudas, and D. Srivastava, "On Effective Multi-Dimensional Indexing for Strings," *In Proc. of ACM SIGMOD*, pp. 403-414, May 2000.
- [12] S. Kiranyaz and M. Gabbouj, "Hierarchical Cellular Tree: An Efficient Indexing Scheme for Content-Based Retrieval on Multimedia Databases," *IEEE Transactions on Multimedia*, Vol. 9, No.1, pp. 102-119, January 2007.
- [13] C. L. Krumhansl, "Cognitive Foundations of Musical Pitch," *Oxford University Press*, New York, 1990.
- [14] W. Lee and A.L.P. Chen, "Efficient Multi-Feature Index Structures for Music Data Retrieval," *In Proc. Of SPIE Conf. on Storage and Retrieval for Image and Video Database*, 2000.
- [15] C. H. Lin and Arbee L. P. Chen, "Indexing and Matching Multiple-Attribute Strings for Efficient Multimedia Query Processing," *IEEE Transactions On Multimedia*, Vol. 8, No. 2, April 2006.
- [16] C. C. Liu, J.L. Hsu, and Arbee L.P. Chen, "Efficient Theme and Non-Trivial Repeating Pattern Discovering in Music Databases," *In Proc. of IEEE Data Engineering*, pp. 14-21, 1999.
- [17] C. C. Liu, J. L. Hsu, and Arbee L. P. Chen, "An Approximate String Matching Algorithm

- for Content-Based Music Data Retrieval,” *In Proc. of IEEE Int’l Conf. on Multimedia Computing and Systems*, pp. 451-456, 1999.
- [18] Y-L. Lo and S-J. Chen, “The Numeric Indexing For Music Data,” *Proceedings of the IEEE 22nd Int’l Conference on Distributed Computing Systems (ICDCS’2002) Workshops – the 4th Int’l Workshop on Multimedia Network Systems and Applications (MNSA’2002)*, Vienna, Austria, pp. 258-263, July 2002.
- [19] Y-L. Lo and S-J. Chen, “Multi-feature Indexing For Music Data,” *IEEE 23rd International Conference on Distributed Computing Systems (ICDCS’2003) Workshops – the 5th International Workshop on Multimedia Network Systems and Applications (MNSA’2003)*, Providence, Rhode Island, USA, pp. 654-659, May 19-22, 2003.
- [20] Y-L. Lo, H-C. Yu, and M-C. Fan, “Efficient Non-trivial Repeating Pattern Discovering in Music Databases,” *Tamsui Oxford Journal of Mathematical Sciences*, Vol. 17, No. 2, pp. 163-187, Nov. 2001.
- [21] Y-L. Lo and C-H. Wang, “Hybrid Multi-Feature Indexing for Music Data Retrieval,” *The 6th IEEE International Conference on Computer and Information Science (ICIS 2007)*, Melbourne, Australia, pp. 543-548, July 11-13, 2007.
- [22] Y-L. Lo and C-H. Wang, “Economical Structure for Multi-feature Music Indexing,” *The International MultiConference of Engineers and Computer Scientists (IMECS 2008)*, Hong Kong, pp. 497-501, March 19-21, 2008.
- [23] Y-L. Lo, W-L. Lee, and L-H. Chang, “True Suffix Tree Approach for Discovering Non-trivial Repeating Patterns in a Music Object,” *Accepted by Journal of Multimedia Tools and Applications*, Springer, Vol. 37, No. 2, pp. 169-187, April 2008.
- [24] E. McCreight, “A Space-Economical Suffix Tree Construction Algorithm,” *Journal of Association for Computing Machinery*, pp. 262-272, 1976.
- [25] E. Narmour, *The Analysis and Cognition of Basic Melodic Structures*, The University of Chicago Press, Chicago, 1990.
- [26] B. Schuller, G. Rigoll, and M. Lang, “Multimodal Music Retrieval for Large Databases,” *IEEE International Conference on Multimedia and Expo (ICME)*, Vol. 2, pp. 755-758, June 2004.
- [27] J. Sundberg, A. Friberg, and L. Fryden, *Common Secrets of Musicians and Listeners: An Analysis-by-Synthesis Study of Musical Performance, In Representing Musical Structure*, P. Howell, R. West, and I. Cross, eds., Academic press, London, 1991.
- [28] T-H. Tsai and J-H. Hung, “Content-Based Retrieval of MP3 Songs for One Singer Using Quantization Tree Indexing and Melody-Line Tracking Method,” *ICASSP 2006 Proceedings*, pp. 505-508, May 2006.
- [29] Y. H. Tseng, “Content-Based Retrieval for Music Collections,” *In Proc. of ACM SIGIR’99*, pp. 176-182, 1999.
- [30] A. L. Uitdenbogerd and J. Zobel, “Melodic Matching Techniques for Large Music Databases,” *In Proc. of ACM Multimedia*, pp. 57-66, 1999.
- [31] E. Ukkonen, “On-Line Construction of Suffix Tree,” *Algorithmica*, Vol. 14, pp. 249-260, 1995.
- [32] P. Weiner, “Linear Pattern Matching Algorithms,” *In Proc. Of IEEE Ann. Symp. On Switching and Automata Theory*, pp. 1-11, 1973.
- [33] Y-D Wu, Y. Li, and B-L Liu, “A New Method for Approximate Melody Matching,” *Proceedings of the Second International Conference on Machine Learning and Cybernetics*, Vol. 5, pp. 2687-2691, Nov. 2-5 2003.