**Musicae Scientiae**
Discussion Forum 4A, 2007, 135-152



# On comparing edit distance and geometric frameworks in content-based retrieval of symbolically encoded polyphonic music

KJELL LEMSTRÖM AND ANNA PIENIMÄKI
Dept. of Computer Science, University of Helsinki

• **Abstract**

This paper deals with content-based music retrieval (CBMR) of symbolically encoded polyphonic music. It is one of the key issues in the field of music information retrieval. Due to extensive research, there are already satisfactory methods for monophonic CBMR. Unfortunately, this is not the case with the polyphonic task. The problem has been approached in various ways; the majority of the methods suggested fall into two frameworks. The first framework models music as linear strings and the similarity is based on the well-known edit-distance concept. The second one models music as sets of two-dimensional geometric objects (consider the piano-roll representation), but the definition of similarity varies considerably within the framework. We scrutinise these frameworks trying to find common, relevant properties that either inhibit or boost the effectiveness of the methods. Although the edit-distance framework offers more efficient solutions, we conclude that the geometric framework is the choice for the CBMR task because of the very natural way of modelling music still preserving the features intrinsic to the task.

## 1. Introduction

Music information retrieval is an interesting and topical multidisciplinary research area. One of its central questions is the content-based music retrieval (CBMR) problem, where a short excerpt of music, called the *pattern*, is searched for in a larger body of music, called the *score*. In this paper we assume that either the pattern or the score, but not necessarily both of them, is polyphonic.

Here we consider only symbolically encoded music, that is, we are interested in the similarity of the music content, not of the sound. There are several ways to represent musical content with symbols, the most widespread being the common music notation. In symbolic representations musical score is presented as a collection of musical events, each described with a symbol assigned to it. The musical events usually represent the minimal unit of the representation at hand, such as note, chord or even a larger musical component, and thus define the level of abstraction. Moreover, each musical event is described by one or more attributes, e.g., onset time, note length and pitch.

Another, well-known symbolic representation is the piano-roll-like representation (see e.g. Figure 6). Let us assume that both the pattern and the score are represented as piano-rolls, then a CBMR search is intuitively equivalent to a process where the piano-roll corresponding to the pattern is shifted on top of the piano-roll corresponding to the score in different alignments. In each alignment it is examined whether the pattern matches, using some appropriate definition of the similarity, with the aligned part of the score.

So, in order to retrieve music, we need an underlying concept of similarity to be able to discriminate which locations within the score are occurrences of the pattern. Music similarity is a very subtle, complex and subjective issue. To measure the similarity between two short monophonic excerpts is already a difficult task. In the literature one can find sophisticated similarity measures based on musicological findings that, for instance, given the tonic, takes into account the consonance/dissonance of the interval at hand (Mongeau and Sankoff, 1990), or accommodates to the prevalent musical key (Shmulevich et al., 2001). A recent successful method (Grachten et al., 2005) is based on machine learning and Narmour's well known implication-realisation model (Narmour, 1990).

When the score at hand is very large, one often needs to compromise over the similarity and use a formulation that is simpler and that can be computed faster. However, this alone will not suffice, but also careful algorithmic planning is required, especially when dealing with polyphonic (or homophonic) music. Consider, for instance, a short piece of homophonic, unanalysed music comprising 500 musical events, such that 5 notes always appear simultaneously (i.e. there are 100 distinct onset times). This short excerpt already generates $100^{\wedge 5}$ = 10,000,000,000 distinct monophonic lines for a naïve method to be considered. So we obviously need something more sophisticated to avoid the combinatorial explosion of the naïve method when dealing with very large databases.

Changing the musical framework from monophonic to polyphonic affects our perception of similarity. Therefore, simplifying the similarity measure may not necessarily do any harm; there is no guarantee that the aforementioned, sophisticated similarity measures would serve any better in this case.

In this paper we investigate CBMR methods applicable to symbolic, polyphonic music and large scores. Without losing generality, we limit our discussion to two categories of CBMR methods: the edit distance framework (EDF) is often used in CBMR, since it is easily adapted to the problem and there is a variety of available solutions for different variations of the problem, see e.g. (Mongeau and Sankoff, 1990; Ghias et al., 1995; Lemström, 2000). EDF methods model music as linear strings and define similarity by using the edit distance concept that relies heavily on the consecution of the elements within the strings. Having encoded music as strings, the (dis-)similarity of any two strings is simply achieved by calculating the number of operations needed to convert one string to the other.

The geometric framework (GF), on the other hand, models music as two-dimensional geometric plotting of objects, without a built-in, rigid linear ordering. The framework was first introduced by Clausen et al. (2000), later Meredith et al. (2002) and Typke et al. (2003) presented alternative formulations for it. Within this framework there is no common definition of the similarity.

For monophonic CBMR, there is yet another possible framework that is the one containing probabilistic methods. To our knowledge, however, no algorithms based on this approach have been developed for the polyphonic case.

Before introducing how these two frameworks work with the polyphonic CBMR problem, let us first briefly review the theoretical basics of the edit distance framework.

*Edit Distance Framework*

*Strings* are sequences of symbols that belong to a finite ordered set called an alphabet and denoted by $\Sigma$. String $A$ is denoted as $a_1 a_2 \ldots a_n$, where $a_i$ is drawn from $\Sigma$ and the index $i$ denotes the position of symbol $a_i$ in $A$. The length of $A$ is denoted by $|A| = n$. The set of all sequences over $\Sigma$ is denoted by $\Sigma^*$. If string $A$ is of form $A = \alpha\beta$, where $\alpha, \beta \in \Sigma^*$, we say that $\alpha$ is a *prefix* and $\beta$ a *suffix* of $A$. Moreover, string $S$ is a *subsequence* of string $A$, denoted as $S \subseteq A$, if it can be obtained from $A$ by deleting zero or more symbols.

Strings can be rewritten by using *editing operations*, such as insertion, deletion and substitution. Editing operations are equipped with weight function $w$. The *edit distance* between any two strings is defined as the minimum cost of allowed editing operations needed for converting one string to the other. For instance, given two strings $A$ and $B$, the well-known *unit cost edit distance* (or Levenshtein distance) uses the editing operations just mentioned, each associated with a unit cost; the so-called *indel distance*, $d_{ID}(A,B)$, allows the use of unit-weighted insertion and deletion operations.

Another, related similarity measure is given by the concept of longest common subsequence. The *longest common subsequence*, $lcs(A,B)$, is the longest sequence that is a subsequence of both $A$ and $B$. The length of the longest common subsequence, denoted $LCS(A,B)$, is a dual operation to $d_{ID}(A,B)$:

$$d_{ID}(A,B) = |A| + |B| - 2^* \; LCS(A,B).$$

Thus, algorithms computing $d_{ID}(A,B)$ can be used to compute $LCS(A,B)$ and vice versa.

Edit distances $d(A,B)$ are usually evaluated by using dynamic programming. Such a procedure tabulates the distances between the prefixes of $A$ and $B$: each cell $d_{ij}$ of the table stores the edit distance between $a_1 \ldots a_i$ and $b_1 \ldots b_j$, $(0 \leq i \leq m, 0 \leq j \leq n)$. The table is evaluated row by row or column by column by using the initialisations $d_{i0} = i$, $d_{0j} = j$ and the recurrence:

$$d_{ij} = \min \begin{cases} d_{i-1,j} + 1, \\ d_{i,j-1} + 1, \\ d_{i-1,j-1} + (\text{if } a_i = b_j \text{ then } 0 \text{ else } 1). \end{cases}$$

Finally, $d_{mn}$ equals the distance $d(A,B)$.

The pattern-matching problem is solved analogously, first changing the initialisations to $d_{i0} = i$, $d_{0j} = 0$. In this case, the distance in the cell $d_{mj}$ of the bottom row of the table corresponds to the minimum distance between the pattern and any suffix of the prefix of the score ending at position $j$.

Naturally filling in such a table of $m \times n$ cells requires computational time proportional to $mn$ (i.e. time $O(mn)$, for short). This quadratic behaviour is rather characteristic of CBMR methods based on the edit distance framework. In some cases the columns of the table can be replaced by bit vectors. Then instead of a single cell, possibly a whole column of cells can be processed in a single step, thus introducing a speed-up of a constant factor (see (Holub et al. 2001; Lemström and Tarhio, 2003; Lemström et al. 2005) how to apply this technique to CBMR).

## 2. String representations for polyphonic music

Because of their ordered linear nature, string representations can be used rather naturally to represent monophonic melodies. In polyphonic music, however, the musical events are not in a strict sequential order, as many of them appear simultaneously. Let us examine the excerpt depicted in Figure 1, where there is a monophonic vocal line and polyphonic accompaniment part. Although the vocal part dominates the perception of this excerpt, there are several different themes in the accompaniment part that can also be of interest.

### 2.1. Non-Interleaved Representation

A straightforward way to form a string representation for polyphonic music is to divide polyphonic texture into several monophonic lines according to composed or perceived voices. In the *non-interleaved representation* each monophonic line is represented by an associated (monophonic) string. The resulting strings are stored sequentially, separated by an extra symbol not belonging to the alphabet (Figure 2).

Together with a musically motivated edit distance measure (recall the ones mentioned in the introduction), this representation is effective when the pattern to be searched for resembles an excerpt within the melody (or within some other voice) of the score. Such an occurrence within one voice is called a *solid occurrence* of the query pattern (see Figure 2.a).

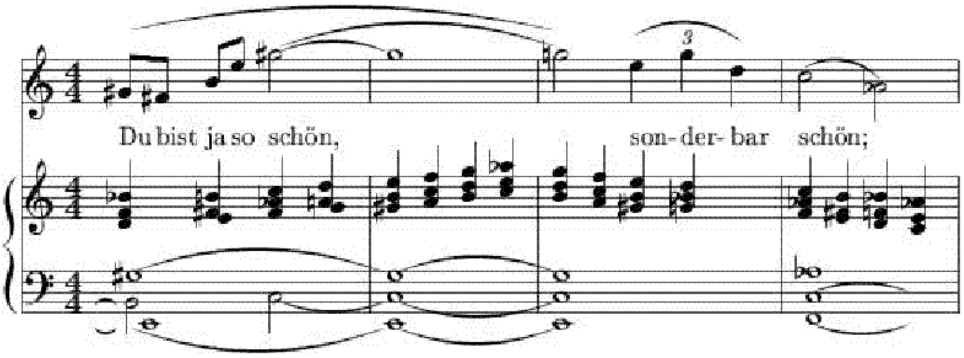


*Figure 1.*
*An excerpt of polyphonic music from Einojuhani Rautavaara's opera Thomas (1985). Printed with the permission of the publisher Warner/Chappell Music Finland Oy.*

When the music at hand does not contain clear voicing, but it is rather perceived as a progression of chords, forcing the musical data into monophonic voices may result in a musically meaningless representation. Moreover, a pragmatic problem is that the representation does not support the problem depicted in Figure 2.b, where one wishes to find occurrences of the query pattern that may be distributed across the voices of the score.

a) (0,68) (1,66) (2,71) (3,76) (4,80) %
(0,70) (2,71) (4,72) (6,74) %
(0,65) (2,66) (4,68) (6,69) %
(0,62) (2,64) (4,66) (6,67) %
(0,56) % (0,47) (4,48) % (0,40) %

b) (0,68) (1,66) (2,71) (3,76) (4,80) %
(0,70) (2,71) (4,72) (6,74) %
(0,65) (2,66) (4,68) (6,69) %
(0,62) (2,64) (4,66) (6,67) %
(0,56) % (0,47) (4,48) % (0,40) %

*Figure 2.*
*Non-interleaved representation of the first bar of the excerpt in Figure 1. Notes are represented by ordered pairs (o, p) where o represents the onset time and p the pitch information. We have emphasised a) a solid occurrence, and b) a distributed occurrence of an imaginary query pattern.*

## 2.2. Interleaved Representation

Another monophonic reduction of the polyphonic data is to sort the notes in *lexicographical order*: the notes are firstly ordered by their onset times. Then, to achieve a total order, the simultaneous notes are ordered using the associated pitch values (see Figure 3). We call this *the interleaved representation*.

The interleaved representation (see e.g. Pienimäki, 2002) is more flexible than non-interleaved, as regards handling polyphonic musical structures. Nevertheless, the forced strict ordering is somewhat artificial by nature; simultaneously played notes are rarely perceived to be in any particular order. The crucial problem, once again, is a pragmatic one. By definition, within the score there is a large number of "extra" intervening elements in between any two notes belonging to an occurrence. Therefore, when using an edit distance measure with an error threshold, either no occurrence is to be found (the case of a small threshold value) or a large number of false positives are to be reported (the case of a large threshold value).

As the representation preserves the overall order of the notes, one could use the longest common subsequence measure. However, the approach faces a deep efficiency problem if transposition invariance is required.

(0,40) (0,47) (0,56) (0,62) (0,65)
(0,68) (0,70) (1,66) (2,64) (2,66)
(2,71) (3,76) (4,48) (4,66) (4,68)
(4,72) (4,80) (6,67) (6,69) (6,74)

*Figure 3.*
*The first bar of the excerpt of Figure 1 in interleaved representation.*

### 2.3. Onset-Based Representation

In the onset-based approach all notes having their onsets simultaneously are also to be considered at the same time, thus making it possible to find distributed occurrences, as well. If the voicing information is available, the monophonic voices are aligned based on the onsets. Another possibility is to use the representation depicted in Figure 4, where notes having simultaneous onsets are grouped together. The onset-based approach is more flexible than the previous ones, but careful algorithmic planning is required to avoid the combinatorial explosion.

If transposition invariance is not needed the edit distance framework can be used straightforwardly; see e.g. Bloch and Dannenberg (1985), Dovey (2001), and Holub et al. (2001). Doraisamy and Rüger (2001) avoided the combinatorial explosion in the transposition invariant case by chopping the possibilities in *n*-grams (strings of length *n*). When the degree of polyphony is high, however, this may prove not to be an efficient solution because of the large number of generated *n*-grams. Another possibility, introduced by Lemström and Tarhio (2003), is to split the pattern to bare intervals that are searched individually. This requires a two-pronged search: the candidates found in the first phase have to be examined in another phase to discard *discontinuous patterns* (the subsequent interval does not begin in the note where the former ends (see Figure 5)). As a major drawback, the method does not allow any extra intervening elements to appear in the score.

Recently, Lemström and Mäkinen (2005) introduced an algorithm finding

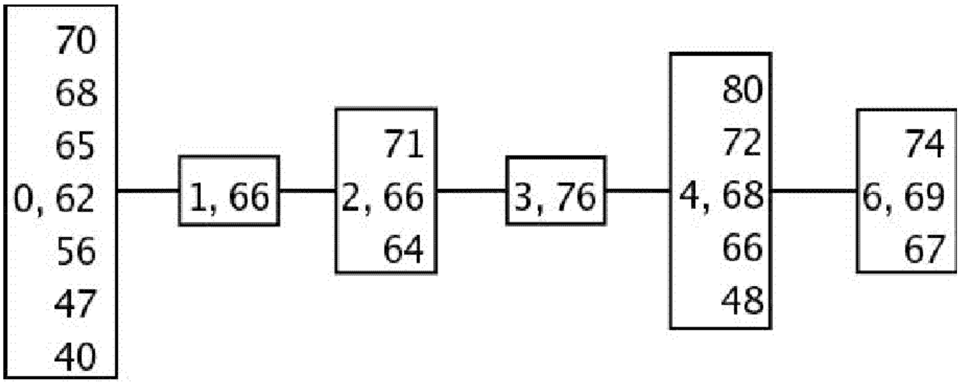


*Figure 4.*
*The first bar of the excerpt in Figure 1 in an onset-based representation.*

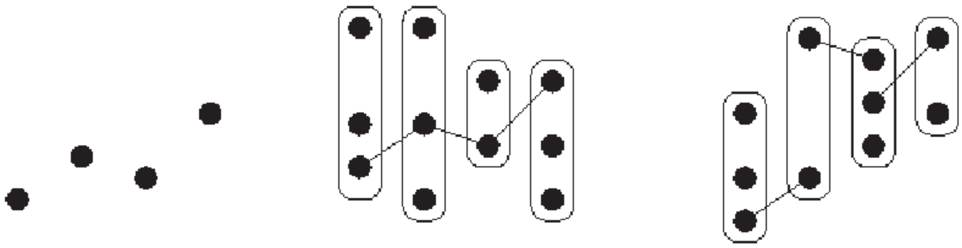

*Figure 5.*
*A pattern (left) and its continuous (middle) and discontinuous (right) occurrences in polyphonic scores.*

occurrences that minimise the number of jumps across the voices (consider for instance the example depicted in Figure 2.b of two jumps).

### 2.4. Note Duration Problem

The above representations, in their basic forms, do not consider the durations of the notes — only pitch and (the order of) onset time information are used. For the similarity judgements, however, it is often crucial to know the durations of the notes, as well. One may wish, for instance, to calculate the longest common time (LCT) of the query pattern with any possible excerpt of the score.

In non-interleaved and interleaved representations note length information is trivially added in the representation in which case the strings would consist of pairs (as in Figures 2 and 3). In onset-based representation, however, including durational information causes problems.

Recall that the sets are formed by notes beginning at the same time. Such notes are in many cases perceived as a chord, thus having a common function in the polyphony. The melodic line may often have longer notes during which the notes belonging to the accompaniment may change. Every such change forces a new set to take place in the representation, which complicates the search process. This leads to the idea of fragmenting the long notes: each note change induces a new set that stores, not only the notes beginning at that particular time, but also the notes still playing. Mongeau and Sankoff (1990) introduced the well-known variation of the edit distance measure that deals with such fragmentations. However, their measure is not directly applicable to the transposition invariant, polyphonic search problem. Moreover, fragmentation caused by polyphony will have a remarkable effect on the length of the score, and thus also on the efficiency of the search algorithm.

## 3. Geometric representations for music

Let us now consider another approach that is intuitive, allows a natural handling of polyphony and is easily visualised. Here the objects inhabit some multidimensional (usually the two-dimensional pitch-against-time) space. In this case the content-based searching becomes closely related to the classical problems considered in computational geometry; thus the designation of geometric representation.

The geometric representations resemble the well-known piano-roll representation that can be visualised by a canvas on which the musical elements (corresponding to the notes) are plotted. The elements are formed of pairs *(o, p)* where *o* and *p* correspond to the onset time and the pitch, as above, but this time the pair is interpreted as a co-ordinate in the corresponding two-dimensional space, see Figure 6.

### 3.1. Indexing via Metrical Position Reduction

Clausen et al. (2000) suggested an indexing schema to be used with the geometric representation. Their aim was to achieve query times that are sublinear in the length of the score. To this end, the onset times are quantized to a pre-selected resolution in order to discretise the time dimension, thus limiting the search space. Note that in traditional western music the pitch dimension is readily discrete. Moreover, the timing information *o* is represented relatively to its metrical position within the measure. So, for instance, two consecutive quarter notes with (midi) pitch values 60 and 64 in a sixteenth-note resolution could be encoded as pairs (1,60) and (5,64), respectively.

In a preprocessing phase, all pairs *(o, p)* within the score are collected together with the information where they occur. The latter piece of information is encoded as pair *(i, M)* where *M* refers to the *M*th measure within the *i*th piece. The inverted file contains, for each pair *(o, p)*, a list of its occurrences (i.e. a list of pairs *(i, M)*) within the score.

Having constructed the index, exact occurrences of a query can be found

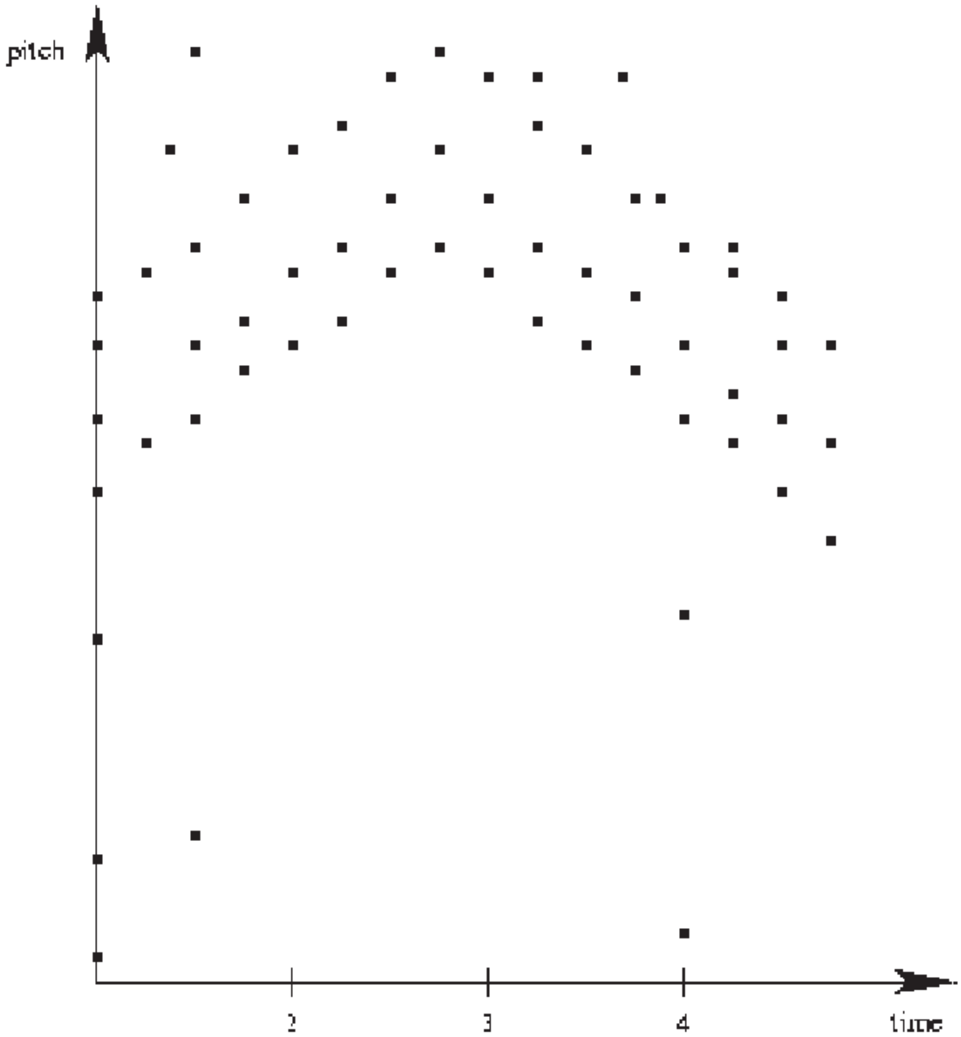


*Figure 6.*
*The example of Figure 1 in piano-roll representation.*

straightforwardly: the query is first analysed and encoded as described above. Then the occurrence lists for each of the query notes are retrieved. Finally the exact occurrences are revealed by computing the intersection of the occurrence lists.

The efficiency is achieved with the cost of robustness. The method finds only occurrences (total and partial) that have a similar metrical position as the query. Moreover, local time and pitch fluctuations cannot be dealt with. By definition the method is neither time scaling (tempo) nor transposition invariant. However, they obtain transposition invariance by using a mathematical trick that outperforms a brute-force solution.

### 3.2. TRANSPOSITION INVARIANCE VIA TRANSLATION VECTORS

Using the piano-roll representation, finding a transposition invariant, exact occurrence of any pattern becomes straightforward when working on translation vectors between the points rather than on the points themselves. Here "translation" is used in the mathematical sense: the pattern may be slid both in the horizontal and in the vertical direction. A vertical move corresponds to transposition, while a horizontal move corresponds to aligning the pattern (or one point in the pattern) time-wise. Being able to use vertical and horizontal moves give us transposition and starting position invariances, respectively. Thus, the original task becomes finding a translation vector that translates each of the points in the pattern to some point within the score (Wiggins et al. 2002). Moreover, because the method does not work on the translation vectors within the pattern (and try to find whether such a translation vector pattern occurs within the database), it avoids the problem depicted in Figure 5.

Now, if you think of this case in terms of the edit distance framework, the method allows an indefinite number of deletions (without paying the cost of using the operation and thus avoiding EDF's problem with a fixed threshold). Moreover, processing the translation vectors in a suitable order, total, transposed occurrences of the pattern are found in a linear expected time (in the length of the score) (Ukkonen et al. 2003; problem PI). In the worst case, however, the computational complexity is quadratic.

The above solution can be modified also to the case of finding all the partial occurrences of the pattern, i.e. it suffices that only some of the points in the pattern are translatable to points in the score (Wiggins et al. 2002). With careful planning of the processing order of the translation vectors, the problem can be solved in *O(nmlogm)* time (where $n$ and $m$ refer to the length of the score and the pattern, respectively) (Ukkonen et al. 2003; algorithm PII). Once again, thinking this problem in terms of the edit distance framework, also (free) insertions are now allowed.

These algorithms find all transposed (total and partial) occurrences of the pattern. Still they cannot deal with local time and pitch fluctuations, and they are not time scaling invariant.

### 3.3. CONSIDERING DURATIONS

#### 3.3.1. Durations via Horizontal Line Segments

An obvious way to include durational information is to replace all the points in the piano-roll representation by horizontal line segments where the length of any such line segment corresponds to the length of the note (Figure 7). In this setting, an intuitive similarity measure would be to calculate the longest common time: find translations such that the total common length of the horizontal line segments of the pattern and of those of the score is maximised.

A natural visualisation of this problem is such that the line segments of the pattern are drawn on a transparency and the line segments of the score on a paper.

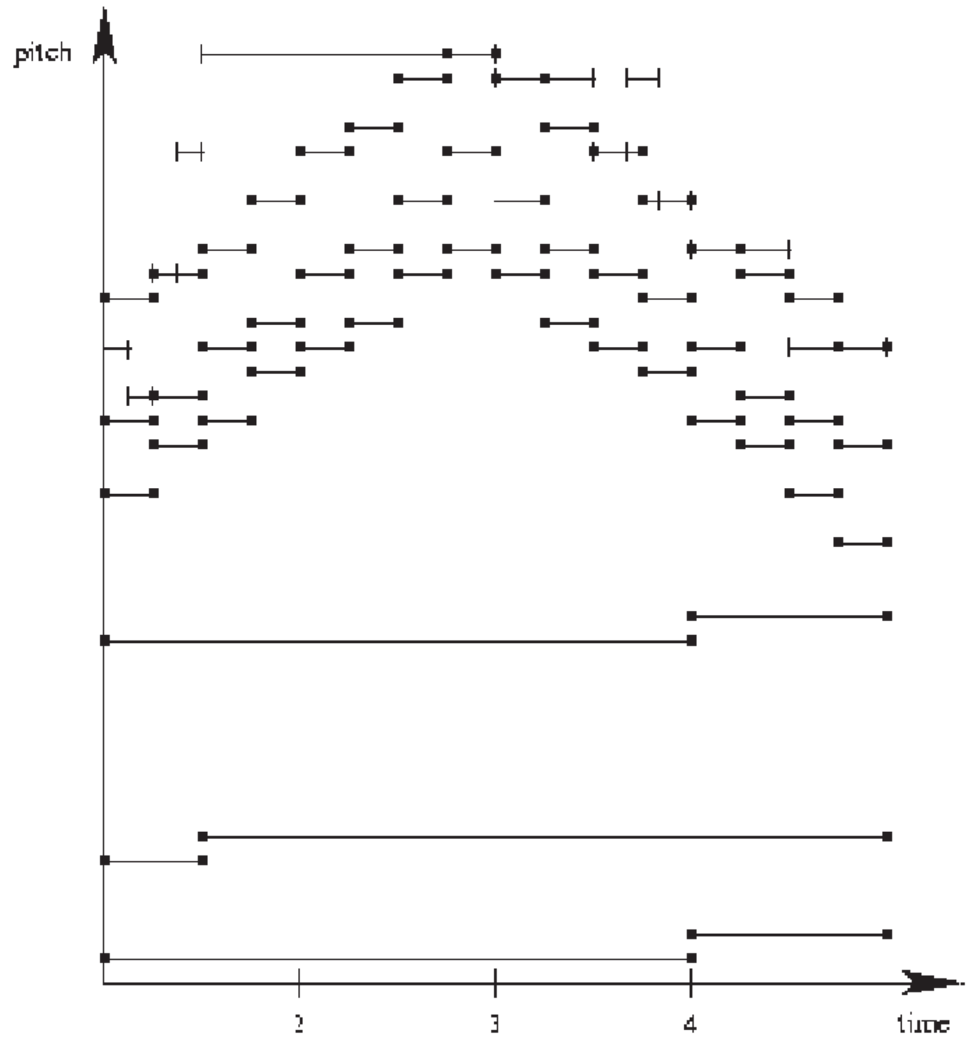


*Figure 7.*
*The example of Figure 1 in horizontal line segment representation.*

Then the transparency is slid on top of the paper. The location, where the segments on the transparency and on the paper intersect as much as possible, gives the answer to our problem.

Ukkonen et al's (2003) method (recall the PII algorithm) can be modified also to this problem. In the PIII algorithm some extra data structures are required, but interestingly it is computationally as efficient as PII.

Obviously the setting takes care of tempo fluctuations, but not of pitch fluctuations. One attempt in this direction was presented by Lubiw and Tanur (2004). They used a more general weight function that measures the area between the translated pattern and the score (and search for the minimum such setting). However, the method cannot yet deal with global time scalings.

### 3.3.2. Durations via Weighted Points

Typke et al. (2003) suggested another possibility to take the durations into account. In their representation the size of the point in the piano-roll reflects the duration of the corresponding note (see Figure 8). The intuition in their similarity measure is that the points in the pattern are piles of soil (or of earth) and the points in the score are holes. Then the similarity between two such point patterns is measured by the amount of earth needed to be carried to the holes in order to fill them, weighting the measure by the distance the soil has to be carried. Therefore, the measure is called the Earth Mover's Distance (EMD).

The EMD is robust against local pitch and tempo fluctuations. To achieve transposition invariance, either one of the two excerpts is transposed so that the weighted average pitch becomes equal or all possible transpositions need to be calculated. In (Typke et al., 2004), the method was generalised to the polyphonic case. Moreover, to achieve time scaling invariance, they use an idea resembling the aforementioned Doraisamy and Rüger's (2001) *n*-gram method. They have also built an indexing schema applicable to their method, to avoid the very expensive online calculation of the transportation distance.

## 4. Conclusions

In this paper, we have overviewed two frameworks that have been used to solve one of the focal tasks in music information retrieval, i.e., the problem of content-based music retrieval (CBMR) of symbolically encoded polyphonic music. Methods using the edit distance framework model music by using linear strings and define the similarity by adapting the concept of edit distance. The recently introduced geometric framework was developed just for this particular matter. In literature one can find several CBMR methods that fall in one of these two categories. In describing the frameworks, we have included several representative methods as illustrative examples.

Both frameworks have their advantages and disadvantages. Combinatorial pattern matching is a mature field of computer science. Thus, basing the solution on the edit distance framework, one will easily obtain a reasonable and very efficient solution for the simplest CBMR problems. Interestingly, despite its one-dimensional linearity, the framework has also been used for polyphonic CBMR with respectable results. The poverty of the representation becomes salient when trying to combine transposition invariance and polyphony. In such a case something needs to be compromised. The edit distance framework offers robustness against local tempo and pitch fluctuations. Pitch fluctuation can be dealt with by using a reduced version of the underlying alphabet such as music contour (up, down, same), Narmour's

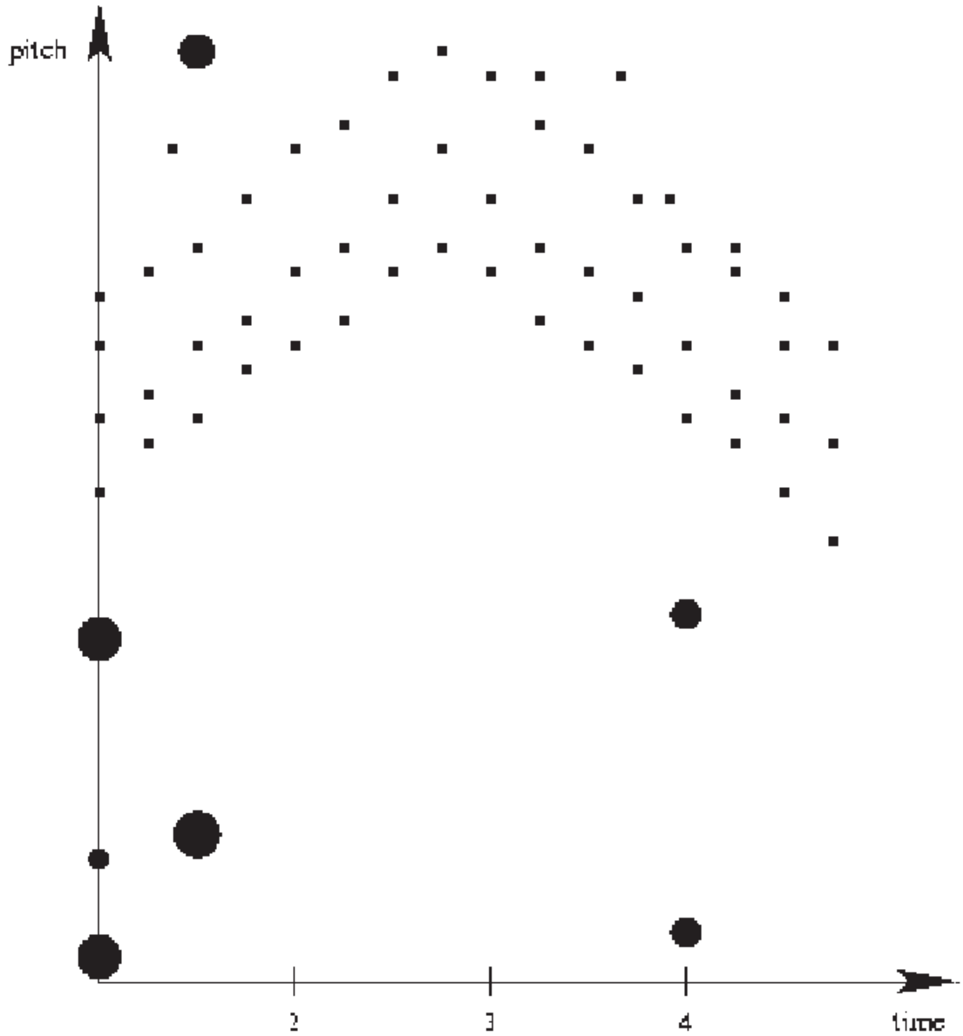


*Figure 8.*
*The example of Figure 1 in sized point representation.*

(1990) small, medium size and large intervals or Lemström and Laine's (1998) quantised partially overlapping intervals. What seems a major disadvantage of the edit distance framework at first glance is that it preserves only the order of the notes out of the temporal information. However, this proves to be a powerful property for practical purposes. It does not only provide robustness against local tempo fluctuation, but it makes the representation time scaling invariant.

The geometric framework is novel and there is still much to be studied. However, it seems promising: the piano-roll-like representation is intuitive, and the rich representation deals naturally with polyphony, transposition invariance and durational information. Although it does not yet provide as efficient and robust tools as the edit distance framework, major steps towards this have already been taken. Transposition invariance is a built-in feature in most of the geometric methods. Pitch and time fluctuation can also be dealt with naturally either by using Earth Mover's Distance, or by replacing the above presented horizontal line segments by horizontal box segments (this is a study on progress). Moreover, Typke et al. have also been able to deal with time scaling invariance when using the geometric framework.

When dealing with very large databases, the efficiency of the algorithms becomes a major issue. Online algorithms of the edit distance framework are somewhat more efficient than those of the geometric framework. Moreover, Lubiw and Tanur (2004) showed that using the geometric framework, subquadratic online algorithms cannot be obtained without a major breakthrough in the theory of computation. Nevertheless, very large databases need to be processed faster, using offline algorithms and indexing structures. Edit distance framework provides well-known and very efficient indexing structures. Recently, the first steps towards indexing piano-rolls have also been taken (Clausen et al. 2000, Typke et al. 2004).

Although the edit distance framework provides somewhat more efficient and robust tools, the clearly more effective representation for polyphonic music of the geometric framework compensates that. Moreover, as the edit distance framework faces severe problems in combining polyphony and transposition invariance, which both are very important and intrinsic features of the musical task at hand, we conclude that the geometric framework is the choice for successful polyphonic CBMR.

**Address for correspondence:**
**Kjell Lemström, Anna Pienimäki**
**Dept. of Computer Science**
**University of Helsinki**
**Helsinki, Finland**
**e-mail: {Kjell.Lemstrom, Anna.Pienimaki}@cs.Helsinki.Fi**

## • REFERENCES


Bloch, J. J. & Dannenberg, R. B. (1985). Real-time accompaniment of polyphonic keyboard performance. *Communications of the ACM, 35 (10)*, 74-82.

Clausen, M., Engelbrecht, R., Meyer D., & Schmitz, J. (2000). PROMS: A web-based tool for searching in polyphonic music. In *Proceedings of the 1st International Symposium on Music Information Retrieval*. Plymouth, Massachusetts, USA.

Doraisamy, S., & Rüger, S. M. (2001). An approach towards a polyphonic music retrieval system. In J. S. Downie and D. Bainbridge (eds) *Proceedings of the 2nd International Symposium on Music Information Retrieval* (pp. 187-93). Bloomington, Indiana, USA.

Dovey, M. J. (2001). A technique for regular expression style searching in polyphonic music. In J. S. Downie & D. Bainbridge (eds) *Proceedings of the 2nd International Symposium on Music Information Retrieval* (pp. 179-85). Bloomington, Indiana, USA.

Ghias, A., Logan, J., Chamberlin D., & Smith B. C. (1995). Query by humming – musical information retrieval in an audio database. In *Proceedings of the 1995 ACM Multimedia Conference* (pp. 231-36). San Francisco, California, USA.

Grachten, M., Arcos, J. L., & de Mántaras, R. L (2005). Melody retrieval using the implication/realization model. *Electronic proceedings of the MIREX 2005 competition*. http://www.music-ir.org/evaluation/mirex-results/articles/similarity/grachten.pdf [07.11.2006]

Holub, J., Iliopoulos, C. S., & Mouchard, L. (2001). Distributed string matching using finite automata. *Journal of Automata, Languages and Combinatorics, 6 (2)*, 191-204.

Lemström, K. (2000). *String Matching Techniques for Music Retrieval* (PhD thesis), Helsinki: University of Helsinki.

Lemström, K. & Laine, P. (1998). Musical information retrieval using musical parameters. In *Proceedings of the 1998 International Computer Music Conference* (pp.341-48). Ann Arbor, Michigan, USA.

Lemström, K. & Mäkinen, V. (2005). On minimizing pattern splitting in multi-track string matching. *Journal of Discrete Algorithms, 3 (2-4)* 48-66.

Lemström, K., Navarro, K. & Pinzon, Y. (2005). Bit-parallel algorithms for transposition-invariant multi-track string-matching. *Journal of Discrete Algorithms, 3 (2-4)*, 267-92.

Lemström, K. & Tarhio, J. (2003). Transposition invariant pattern matching for multi-track strings. *Nordic Journal of Computing, 10 (3)*, 185-205.

Lubiw, A. & Tanur, L. (2004). Pattern matching in polyphonic music as a weighted geometric translation problem. In C. L. Buyoli & R. Loureiro (eds) *Proceedings of the 5th International Conference on Music Information Retrieval* (pp. 289-96). Barcelona, Spain.

Meredith, D., Lemström, K., & Wiggins, G. A. (2002). Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research, 31 (4)*, 321-45.

Mongeau, M. & Sankoff, D. (1990). Comparison of musical sequences. *Computers and the Humanities, 24 (3)*, 161-75.

Narmour, E. (1990). *The Analysis and Cognition of Basic Melodic Structures: The Implication-Realization Model*. Chicago: The University of Chicago Press.

Pienimäki, A. (2002). Indexing music databases using automatic extraction of frequent phrases. In M. Fingerhut (ed.), *Proceedings of the 3rd International Conference on Music Information Retrieval* (pp. 25-30). Paris, France.

Shmulevich, I., Yli-Harja, O., Coyle, E., Povel, D.-J., & Lemström, K. (2001). Perceptual issues in music pattern recognition - complexity of rhythm and key finding. *Computers and the Humanities*, *35 (1)*, 23-35.

Typke, R., Giannopoulos, P., Veltkamp, R.C., Wiering, F. & van Oostrum, R. (2003). Using transportation distances for measuring melodic similarity. In H. Hoos & D. Bainbridge (eds) *Proceedings of the 4th International Conference on Music Information Retrieval* (pp. 107-14). Baltimore, Maryland, USA.

Typke, R., Wiering, F. & Veltcamp, R. C. (2004). A search method for notated polyphonic music with pitch and tempo fluctuations. In C. L. Buyoli & R. Loureiro (eds) *Proceedings of the 5th International Conference on Music Information Retrieval* (pp. 281-88). Barcelona, Spain.

Ukkonen, E., Lemström, K. & Mäkinen, V. (2003). Geometric algorithms for transposition invariant content-based music retrieval. In H. Hoos & D. Bainbridge (eds) *Proceedings of the 4th International Conference on Music Information Retrieval* (pp. 193-99). Baltimore, Maryland, USA.

Wiggins, G. A., Lemström, K. & Meredith, D. (2002). SIA(M)ESE: An algorithm for transposition invariant, polyphonic content-based music retrieval. In M. Fingerhut (ed), *Proceedings of the 3rd International Conference on Music Information Retrieval* (pp. 283-4). Paris, France.

## • Comparando la "edit-distance" y los marcos geométricos para la recuperación del contenido simbólico codificado de la música polifónica

Este trabajo trata de la recuperación del contenido musical (CBMR) de música polifónica codificada simbólicamente. Éste es uno de los asuntos claves en el campo de la recuperación de la información de la música. Gracias a amplias investigaciones, contamos ya con métodos satisfactorios para la recuperación del contenido musical (CBMR) de la monodía. Lamentablemente, éste no es el caso de la polifonía. El problema ha sufrido una aproximación desde varios puntos de vista ; la mayoría de los métodos sugieren abordar el tema partiendo de dos marcos diversos. El primero presenta la música como cadenas lineales y la similitud está basada en el concepto bien conocido de "edit-distance". El segundo presenta la música como conjuntos de objetos geométricos bidimensionales — véase la representación del rollo de pianola —, pero la definición de similitud varía considerablemente dentro de dicho marco. Nosotros examinamos estos marcos intentando encontrar propiedades comunes relevantes que inhiban o estimulen la efectividad de los métodos. Aunque el marco de "edit-distance" ofrece soluciones más eficaces, concluimos que el marco geométrico es la opción adecuada para la tarea de CBMR, debido a que es la forma más natural de desarrollar modelos musicales preservando los hechos intrínsecos a dicha tarea.

## • Confronto fra *edit distance* e strutture geometriche nel recupero su base contenutistica di musica polifonica codificata simbolicamente

Il presente articolo tratta il recupero su base contenutistica (*content-based music retrieval*, CBMR) di musica polifonica codificata simbolicamente. Si tratta di uno degli aspetti chiave nel campo del recupero dell'informazione musicale. Grazie ad un'ampia attività di ricerca, vi sono già metodi soddisfacenti per il CBMR monodico. Sfortunatamente, non si può dire lo stesso dell'impresa polifonica. Il problema è stato affrontato in vari modi ; la maggior parte dei metodi suggeriva di ricadere in due strutture. La prima struttura modella la musica come stringhe lineari, e la similarità si basa sul ben noto concetto di *edit distance*. La seconda struttura modella la musica come serie di oggetti geometrici bidimensionali (vedi la rappresentazione *piano roll*), ma la definizione di similarità varia considerevolmente all'interno della stessa struttura. Abbiamo esaminato a fondo queste strutture nel tentativo di trovare proprietà comuni e rilevanti che inibiscano o favoriscano l'efficacia dei metodi. Sebbene la struttura *edit distance* offra soluzioni più valide, concludiamo che quella geometrica sia preferibile ai fini del CBMR per la maniera assai naturale di modellare la musica pur conservando gli aspetti intrinseci al compito.

### • Comparaison de l'edit-distance et des cadres géométriques dans l'extraction fondée sur le contenu de musique polyphonique encodée par symboles

Nous étudions ici l'extraction fondée sur le contenu (CBMR : content- based music retrieval) de musique polyphonique encodée par symboles. C'est un problème-clé du domaine de l'extraction d'informations musicales. Grâce à de nombreuses recherches, il existe déjà des méthodes satisfaisantes de CBMR monophonique. Malheureusement, ce n'est pas le cas de la polyphonie. On a déjà abordé le problème de différentes manières et la plupart des méthodes proposées entrent dans deux cadres. Le premier cadre modélise la musique comme ensemble de cordes linéaires et la similarité est fondée sur le concept bien connu d'édit-distance. Le deuxième modélise la musique comme un ensemble d'objets géométriques à deux dimensions (songez à une représentation du rouleau de piano mécanique), mais la définition de la similarité varie énormément dans ce cadre. Nous avons étudié de près ces cadres pour trouver des propriétés communes et pertinentes qui puissent soit faire obstacle, soit améliorer l'efficacité de ces méthodes. Bien que le cadre d'edit-distance présente des solutions plus efficaces, nous en avons conclu que le cadre géométrique est le meilleur pour le travail CBMR car il modélise la musique de manière très naturelle tout en préservant les caractéristiques intrinsèques de ce travail.

### • Über den Vergleich von Ausgabedistanzen und geometrischen Systemen beim inhaltsbasierten Abruf von symbolisch verschlüsselter polyphoner Musik

Dieser Aufsatz behandelt den inhaltsbasierten Musikabruf („Content-based music retrieval" - CBMR) von symbolisch verschlüsselter polyphoner Musik. Das ist einer der zentralen Punkte im Feld des musikalischen Informationsabrufs. Dank extensiver Forschung gibt es bereits befriedigende Methoden für einstimmiges CBMR. Leider trifft dies nicht für polyphone Aufgaben zu. Das Problem wurde verschiedentlich angegangen, die Mehrheit der vorgeschlagenen Methoden lässt sich in zwei Systeme einteilen. Das erste System modelliert Musik als lineare Folge, die Ähnlichkeit basiert dabei auf der Edit-Distanz (oder Editierdistanz). Das zweite System modelliert Musik als ein Set von zweidimensionalen geometrischen Objekten (siehe Klavierrollen-Darstellung), allerdings gibt es beträchtliche Unterschiede in der Definition von Ähnlichkeit innerhalb dieses Systems. Wir untersuchten diese Systeme, um Gemeinsamkeiten Eigenheiten zu finden, die entweder die Effektivität der Methoden verringern oder steigern würden. Obwohl das Editierdistanz-System effizientere Lösungen bietet, schlussfolgern wir, dass das geometrische System für CBMR-Aufgaben vorzuziehen ist, da auf diese Weise Musik sehr natürlich modelliert werden kann und außerdem die aufgabenspezifischen Merkmale beibehalten werden.