

On Optimising the Editing Algorithms for Evaluating Similarity Between Monophonic Musical Sequences

Pierre HANNA, Pascal FERRARO and Matthias ROBINE

LaBRI - Université Bordeaux 1
F-33405 Talence Cedex, France
firstname.name@labri.fr

Keywords: symbolic melodic similarity, edition, local alignment, musical sequences.

Abstract

Melody is an important property for the perceptual description of Western musical pieces. In the monophonic context, retrieval systems based on melodic similarity generally consider sequences of pitches and durations. Algorithms that have been proposed for measuring melodic similarity rely on geometric representations, string matching techniques, etc. Adaptations of editing algorithms, mainly applied in bioinformatic applications, to the musical domain have already been proposed. However, we present in this paper several experiments in order to optimize these methods. The different possible representations for pitches and durations are discussed and evaluated. Optimizations specific to musical applications are proposed and imply significant improvements of the editing algorithm studied. Evaluation of this algorithm led to the best results during the MIREX 2006 symbolic melodic similarity contest.

1 Introduction

Research works in the domain of the musical information retrieval generally concern Western music. Melody is one of the most important property for the perceptual description of such music [Selfridge-Field, 1998]. In this paper we thus focus on the melodic characteristics of musical pieces.

The problem of melodic similarity evaluation has been raised with the development of musical applications such as query-by-humming. However, the notion of similarity between melodies is very difficult to precisely define. From a computational point of view, it consists of determining algorithms calculating a measure which indicates the degree of similarity between two melodic segments. For particular applications like query-by-humming, some properties of the retrieval system are expected. For instance, since a query can be transposed, played faster or slower,

without degrading the melody, retrieval systems have to be transposition invariant and tempo invariant.

Several techniques for evaluating melodic similarities have been introduced during the last few years. Geometric algorithms consider geometric representations of melodies and compute the distance between objects. Some systems [Ukkonen et al., 2003, Lubiw and Tanur, 2004] are closely linked to the well-known piano-roll representation, where notes are represented by horizontal line segments whose length corresponds to the length of the note, and whose coordinates correspond to the onset time and the pitch of the note. Other geometric systems represent notes by weighted points [Typke et al., 2004]. The weight is related to the duration of the note. Distances between such geometric representations are calculated according to the Earth Mover's Distance.

Another algorithm adapted from string matching domain is proposed in [Doraisamy and R ger, 2003, Uitdenbogerd, 2002]. N-grams techniques involve counting the distinct terms that the query and a potential answer have in common. This approach is very simple but appears to be very efficient (see Sec. 3). Nevertheless, this similarity measure (counting the matching subsequences) does not take into account the perceptual properties of the music: only two cases are assumed, the subsequence does match or not. However, this assumption is not as simple concerning the complex perceptual process of the music. This limitation has significant consequences on the accuracy of retrieval systems based on N-gram techniques.

In this paper, we propose a detailed study of editing algorithms mainly developed in the context of DNA sequence recognition [Gusfield, 1997] and their adaptation to the measurement of melodic similarity in the monophonic context [Cambouropoulos et al., 2005]. In monophonic music, no more than one note is sounded at any given time. Editing algorithms have already been presented for application in the monophonic musical context [Mongeau and Sankoff, 1990, Lemstr m, 2000, Crawford et al., 1998]. Retrieval systems based on melodic similarity relies on such algorithms. In this paper, we deal with algorithms that compare monophonic musical sequences. One of the main applications is the comparison of melodies.

Editing algorithms determine the score of operations that are necessary to transform one sequence into another one. We present in the following several experiments to adapt and optimize these editing algorithms to the musical context. The general editing algorithm is presented in Sec. 2. Experiments about the parameters of the algorithms are detailed in Sec. 3 and lead to conclusions that are proposed in Sec. 4.

2 Problem Formalization

Algorithms for retrieval systems based on melodic similarity consist of two main steps. The first one transforms a symbolic monophonic musical piece into a symbolic sequence. The second one computes a similarity score between two representations. These two steps are presented in this section.

2.1 Representation of Monophonic Musical Pieces as Sequences

Monophonic musical pieces can be represented by trees of pitches [Rizo and Iñesta-Quereda, 2002]. This representation implies a hierarchy relying on bars induced by the time signature of the score notation. However different trees can represent the same melody (same sequence of pitches and durations). For example, two melodies with two different time signatures are represented by two different musical scores. In this case, these two melodies sound similar but are represented in a different way.

Following Mongeau and Sankoff's model [Mongeau and Sankoff, 1990], any monophonic score can be represented as a sequence of ordered pairs with the pitch of the note as the first component and its length as the second. Thus, the sequence

$$(B4 \ B4 \ r4 \ C4 \ G4 \ E2 \ A2 \ G8)$$

represents the example illustrated in Fig. 1.



Figure 1: Example of monophonic melody.

Several alphabets of characters and set of number have been proposed to represent pitches and durations [Uitdenbogerd, 2002, Lemström, 2000]. We present only a few ones that we think are the most pertinent in this context.

The melodic contour indicates the variation between successive notes. Only three values are possible: Up, Down, Same. Therefore, the sequence corresponding to Fig. 1 is:

$$SUDUDD.$$

The absolute pitch simply indicates the exact pitch (MIDI notation). For example, the melody of Fig. 1 is represented by:

$$71, 71, 72, 67, 76, 69, 67.$$

In order to reduce the vocabulary, this exact pitch can be represented by their modulo-12 values. The melodic contour can also be taken into account by using positive values when the melody moves up and negative values when it moves down. The *directed modulo-12 absolute pitch* sequence corresponding to the melody represented by Fig. 1 is:

$$11, 11, +0, -7, +4, -9, -7.$$

In the context of query by humming applications, this representation present the huge disadvantage to be not transposition invariant.

At the contrary of the *absolute* pitch representations, the *interval* and *key relative* representations are transposition invariant. The *exact interval* representation is simply the number of semitones between two successive notes. The *exact interval* sequence corresponding to Fig. 1 is:

$$0, 1, 5, 9, 7, 2.$$

This representation can also be limited with modulo-12. Information about melodic direction can also be indicated:

$$0, +1, -5, +9, -7, -2.$$

The *key relative* representations indicate the difference in semitones between notes and the key of the melody. In the case of Fig. 1, the key signature corresponds to C major. Therefore the associated sequence is:

$$11, 11, 0, 7, 4, 9, 7.$$

This representation can also be limited according to modulo-12 and the information about melodic contour can be indicated:

$$11, 11, +0, -7, +4, -9, -7.$$

The limitations of the *key relative* representation is closely linked to the choice of the key. The correct key has to be known in order to compute the correct representation.

Concerning the note durations, the same representations are possible. The duration contour (Shorter, same, Longer) indicates the general variation of duration between successive notes. Therefore the duration representation of the melody of Fig. 1 is:

$$ssssSsL.$$

The *absolute representation* simply indicates the length of the note in sixteenth notes:

$$4, 4, 4, 4, 4, 2, 2, 8.$$

It is important to note that this representation is not tempo invariant, while the *relative* representation is tempo invariant. The difference of durations between successive notes can be expressed as duration subtraction:

$$0, 0, 0, 0, 2, 0, 6$$

or duration ratio:

$$1, 1, 1, 1, \frac{1}{2}, 1, 4.$$

According to these representations, each element of a sequence can thus be formally represented by a symbol belonging to an infinite set Σ of labels. We consider an edit score function s on this set of labels. It assigns a real number $s(x, y)$ to each pairs of labels (x, y) in $\Sigma \cup \{\lambda\}$ where λ represents the empty symbol¹ such that:

$$\begin{aligned} s(x, x) &> 0 && \forall x \in \Sigma, \\ s(x, y) &< 0 && \forall x \neq y, (x, y) \in \{\Sigma \cup \{\lambda\}\}^2. \end{aligned}$$

This means that the score between two symbols x and y becomes higher with their similarity.

¹ $s(x, \lambda)$ is the score of the deletion of symbol x in Σ and $s(\lambda, y)$ is the score of the insertion of y .

2.2 Local Similarity Problem

Measuring similarity between sequences is a well-known problem in computer science which has applications in many fields [Gusfield, 1997, Sankoff and Kruskal, 1983] such as computational biology, text processing, optical character recognition, image and signal processing, error correction, pattern recognition, etc.

In the early seventies, [Needleman and Wunsch, 1970] and then [Wagner and Fisher, 1974] proposed algorithms which compute a similarity measure between two strings of symbols as the maximum score sequence of elementary operations needed to transform one of the strings into the other. Given two strings of symbols S_1 and S_2 of respective lengths $|S_1|$ and $|S_2|$, a set of elementary operators on strings, called edit operations, and a score associated to each edit operation, a score between these two strings is defined as the score of the sequence of edit operations that transforms S_1 into S_2 with maximum score. This similarity measure makes use of the dynamic programming principle to achieve an algorithm with quadratic complexity, *i.e.* in $O(|S_1| \times |S_2|)$.

Let us consider only the three edit operations that are usually used to compare musical sequences: substitution, deletion and insertion. Let e be an edit operation, a score s is assigned to each edit operation as follows:

- if e substitutes x_i (the i th character of S_1) into y_j (the j th character of S_2) then $s(e) = s(x_i, y_j)$
- if e deletes x_i then $s(e) = s(x_i, \lambda)$
- if e inserts y_j then $s(e) = s(\lambda, y_j)$.

The score s is extended to a sequence of edit operation $E = (e_1, e_2, \dots, e_n)$ by letting $s(E) = \sum_{k=1}^n s(e_k)$. This makes it possible to define a score $\sigma(S_1, S_2)$ between sequences S_1 and S_2 as the maximum score of edit operation sequences transforming S_1 into S_2 , namely:

$$\sigma(S_1, S_2) = \max_{E \in \mathcal{E}} \{s(E)\}$$

where \mathcal{E} represents the set of sequences of edit operations transforming S_1 into S_2 .

In many applications, two strings may not be highly similar in their entirety but may contain *regions that are highly similar*. This is particularly true when long stretches of anonymous sequences are compared, since only some internal sections of those strings may be related. In this case, the task is to find and extract a pair of regions, one from each of the two given strings, that exhibits high similarity. This is called *local alignment* or *local similarity problem* [Smith and Waterman, 1981] and is defined as : given two strings S_1 and S_2 , find substrings ρ_1 and ρ_2 of S_1 and S_2 , respectively, whose similarity is maximum over all pairs of substrings from S_1 and S_2 .

The computation of a local similarity allows us to detect local conserved areas between both sequences. The solution of such a problem is based on the notion of suffix mapping between sequences. The local suffix mapping problem for a given pair x_i, y_j of symbols is to find a (possibly empty) suffix ρ_1 of the subsequence $S_1[x_i]$ (defined from the first symbol of string S_1

to x_i) and a (possibly empty) suffix ρ_2 of the subsequence $S_2[y_j]$ of S_2 such that the score of the optimal sequence of edit operations transforming ρ_1 into ρ_2 is the maximum over all scores of sequences of edit operations between suffixes of $S_1[x_i]$ and $S_2[y_j]$.

The score of the sequence solving the optimal local suffix mapping problem (called local score) for a given pair x_i, y_j of symbols is denoted by $LS(x_i, y_j)$:

$$LS(x_i, y_j) = \max\{\sigma(\rho_1, \rho_2), (\rho_1, \rho_2) \text{ suffixes of } S_1 \text{ and } S_2\}.$$

Local similarity between two sequences is then defined as the score of the best pair of local suffixes in trees S_1 and S_2 :

$$LS(S_1, S_2) = \max\{LS(x_i, y_j), (x_i, y_j) \in S_1 \times S_2\}.$$

So, in order to evaluate local similarity, the algorithm needs to find maximum similarity between suffixes of $S_1[x_i]$ and $S_2[y_j]$, for any pair (x_i, y_j) of $S_1 \times S_2$, and then to determine the best pair x_1^{\max}, y_2^{\max} of S_1 and S_2 .

Since we can always choose an empty suffix, $LS(x_i, \theta) = 0$ and $LS(\theta, y_j) = 0$, where θ is an empty sequence. And finally, for any (x_i, y_j) , the proper recurrence for $LS(x_i, y_j)$ is:

$$LS(x_i, y_j) = \max \begin{cases} 0 \\ LS(x_{i-1}, y_j) + s(x_i, \lambda) \\ LS(x_i, y_{j-1}) + s(\lambda, y_j) \\ LS(x_{i-1}, y_{j-1}) + s(x_i, y_j) \end{cases}$$

where x_{i-1} and y_{j-1} respectively represent symbols before x_i and y_j in sequences S_1 and S_2 . Note that if the query sequence S_1 has only one symbol x , then the local score between S_1 and S_2 is obtained from an empty sequence (*ie.* there is no matching) or from a unique matching between x and the most similar symbol of S_2 .

3 Experiments and Results

In this section, we detail the editing algorithm by proposing a detailed study of the different choices of possible settings. All these possibilities have been experimented.

3.1 Evaluation

One of the main problem in the music information retrieval domain is the problem of the evaluation of the system proposed. The first Music Information Retrieval Evaluation eXchange (MIREX 2005) [Downie et al., 2005] is a contest whose goal is to compare state-of-the-art algorithms and systems relevant for Music Information Retrieval. During this first contest, an evaluation topic about symbolic melodic similarity has been performed. Participants have discussed the process of evaluation and proposed an evaluation procedure. The experiments presented in this paper are based on these procedures.

The RISM A/II (International inventory of musical sources) collection is composed of one half-million notated real world compositions. The incipits are symbolically encoded music. They are monophonic and contain between 10 and 40 notes. 11 incipits have been randomly chosen from this collection. A ground truth has been established [Typke et al., 2005] by combining ranked lists that were created by 35 music experts. The resulting ground truth has the form of ranked groups of incipits. The groups contain incipits whose differences in rankings were not statistically significant, but the ranking of the groups is statistically significant.

A tested system returns a ranked list of incipits estimated melodically similar to the query proposed. A few measures are then used to compute a score according to the corresponding ground truth. A specific measure has been proposed: the Average Dynamic Recall (ADR) [Typke et al., 2006a]. It takes into account the ranked groups of the ground truth by indicating how many of the documents that should have appeared before or at a given position in the result list actually have appeared. ADR takes values in the range $[0, 1]$. The higher the ADR measure is, the more accurate the tested system is.

In the following sections, the local editing algorithm proposed has been tested with the MIREX 2005 data training, according to the ADR measure. These data are composed of 11 queries and 580 incipits for the database collection.

3.2 Melody Standardisation

The first part of the experiments proposed concern the melody standardisation. Several approaches can be chosen to represent a symbolic melody. We restrain these possibilities by considering only the pitch and the note and rest durations that compose melodies.

3.2.1 Pitch Representation

Several different representations of pitches have been described in Sec. 2. We only consider during our experiments 4 different representations: *contour*, *absolute*, *interval* and *key relative*. We also study the influence of the contour information by taking into account the information related to the variations between successive notes. Thus we also propose results with two other representations: *directed interval* and *directed key relative*. The results of these experiments are presented in Tab. 1. The editing algorithm tested is in its simplest form: it does not consider any information about duration, substitution scores are constant, etc. . . .

The results clearly underlines that the *contour* representation leads to the worst results. The difference of accuracy is very significant: 0.38 whereas the other results are greater than 0.50. It is mainly justified by the lack of information contained in this representation. It is obvious that several melodies can be represented by the same melodic contour. Thus, the vocabulary proposed appears to be too limited for musical applications [Uitdenbogerd and Zobel, 1999].

The *absolute* and *key relative* representations approximately lead to the same average ADR measures. However, the *key relative* representation presents the great advantage to propose a similarity score that is transposition invariant. The condition of transposition invariance is the before-hand knowledge of the key of the musical pieces studied. Nevertheless a false estimation of the key leads to high errors in the similarity measurement. In the MIREX 2005 training data,

Pitch representation	average ADR	min ADR	max ADR
contour	0.39	0.03	0.68
absolute	0.55	0.37	0.88
key relative	0.56	0.00	0.90
directed key relative	0.59	0.00	0.91
interval	0.60	0.32	0.93
directed interval	0.62	0.30	0.93

Table 1: ADR measures (average, minimum and maximum values) obtained by the retrieval system considering different pitch representations.



Figure 2: Example of limitations of the *key relative* representation: the query (top) is very similar to the incipit tested (bottom), but the difference of keys C major for the query, G major for the incipit) leads to a very low similarity score.

one query permits to show the limitations of this representation. Fig. 2 shows this query and the incipit that has been judged as the most similar by the musical experts. With the *key relative* representation, editing algorithm considers these two incipits as very different, whereas they are the same at the exception of one note. Another choice for the key signature (C major instead of G major) leads to an estimation of high similarity by the same algorithm. The small difference between the average ADR with the two representations *absolute* and *key relative* is mainly due to the bad results obtained with this query (minimum ADR 0.0 with the *key relative* representation), resulting from a bad choice for the musical key of the query.

Tab. 1 also underlines the average ADR obtained with the *interval* representation. By considering only the pitch difference between successive notes, this representation is transposition invariant. The lowest similarity score computed is thus higher than the one obtained with the *key relative* representation (0.32 instead of 0.0). Nevertheless, this representation also presents some drawbacks. Fig. 3 exhibits an example of the limitation of this representation. The query is represented by the sequence (0, 1, 5, 9, 7, 2), whereas the incipit tested is represented by the sequence (0, 1, 3, 7, 7, 2). Thus, one difference between two musical pieces results in two differences in the musical sequences [Lemström and Ukkonen, 2000]. The score measuring the similarity between pieces may slightly be affected by this error.

Moreover, taking into account the melodic contour slightly improves the results obtained with the MIREX 2005 training data. The average ADR 0.56 obtained with *key relative* representation increases to reach 0.59, whereas it is 0.62 instead of 0.59 for the *interval* representation. Therefore we choose in the next experiments to apply a *directed* representation.

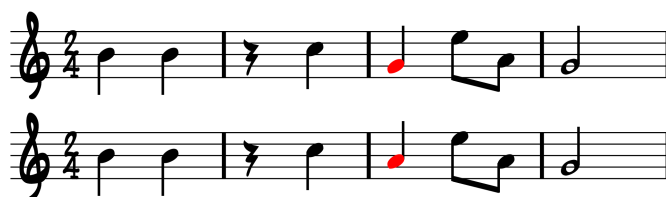


Figure 3: Illustration of the limitations of the *interval* representation: only one note (A instead of G) differs between the query (top) and the musical piece (bottom), but it leads to two differences between the corresponding musical sequences.

Furthermore, because of the limitations of these representations, we propose to experiment a new algorithm that takes into account both representations *directed interval* and *directed key relative*. The simple editing algorithm consists of computing two scores obtained with the two different representations. The maximum score is then considered to compare the melodic similarity between musical pieces. This algorithm presents the advantage to take into account both *key relative* representation, which seems to be the pitch representation that gives the best results if the key signature is correct, and *interval* representation, which permits to correct possible errors due to the false choice of key signature. This algorithm has been tested with the parameters corresponding to the results of Tab. 1: it appears that the results are not better in this case. But experiments with the parameters that gives the best results (see next sections) indicate that this hybrid technique slightly improves the results obtained with a single representation. In tab. 2, the method described is denoted *directed maximum* (*maximum* if the contour is not taken into account). The average ADR obtained by applying this method reaches 0.79 whereas it is only 0.76 or 0.74 with a single representation. Nevertheless, in this case, the time computation of the similarity algorithm is multiplied by two.

Pitch representation	average ADR	min ADR	max ADR
key relative	0.65	0.00	0.95
directed key relative	0.67	0.00	0.96
interval	0.74	0.62	0.88
directed interval	0.76	0.64	0.98
maximum	0.76	0.61	0.92
directed maximum	0.79	0.63	0.96

Table 2: ADR measures (average, minimum and maximum values) obtained by the retrieval system considering different pitch representations and the *maximum* and *directed maximum* techniques. The parameters of our editing algorithm chosen are the ones that give the best similarity results.

Following the different results described in this section, if the musical pieces are tonal music and if the key signature is precisely defined, the *directed key relative* representation is certainly the most appropriate. However, in the general case, in order to prevent rare but high errors, the

directed interval representation has to be chosen. The *maximum* algorithm taking into account the two different representations has to be chosen only when considering applications which require high precision but which do not have any constraint concerning time computation.

3.2.2 Note Duration

Musical pieces are assumed to be represented by sequences of pitches and duration. We propose experiments concerning the duration information and its relevance for similarity comparisons. Experiments performed in [Suyoto and Uitdenbogerd, 2005b] conclude that the combination of pitch and duration does not improve retrieval effectiveness over the use of pitch on its own. The goal of the experiments presented in this section is to check these results.

The duration information can be taken into account when calculating the substitution score between two notes. When substituting a half note by a quarter note, the substitution score should be more important than the score corresponding to a substitution between two notes with the same duration. Therefore, we propose that the score s depends on the pitch range and on the duration difference between two notes x_i and y_j . This score is computed according to existing works [Mongeau and Sankoff, 1990]:

$$s(x_i, y_j) = s_{\text{pitch}}(x_i, y_j) + ks_{\text{duration}}(x_i, y_j)$$

where s_{pitch} is the score due to the pitch difference, and s_{duration} is the score due to the duration difference. The parameter k determines the relative weight of the pitch difference with the duration difference ($k \in \mathbb{R}^+$). If k is null, no information concerning duration is used in the computation of the substitution score. If k is very high, only the information concerning duration is used.

Tab. 3 presents the results (ADR) obtained by taking into account the duration information or not. The *directed maximum* technique has been applied for the pitch representation, and the parameter k is set to 0.20.

Note representation	average ADR	min ADR	max ADR
with duration	0.69	0.33	0.92
without duration	0.61	0.38	0.91

Table 3: ADR measures (average, minimum and maximum values) obtained by the retrieval system considering the duration information or not.

The results clearly show that the average ADR increases when duration information is taken into account. The difference is significant: 0.69 instead of 0.60. The results are thus opposed to the conclusions proposed by [Suyoto and Uitdenbogerd, 2005b], and lead us to suggest that the representation of notes integrating duration improves the quality of melody retrieval systems. Another problem is now raised: computing the substitution score by considering note duration. It relies on the problem of the duration representation. This representation is discussed in the next section whereas the problem of the substitution score is discussed in Sec. 3.3.1.

3.2.3 Duration Representation

Different representations of duration have been described in Sec. 2. The problem of duration representation is similar to the problem of pitch representation choice. We consider both *absolute* and *interval* representations, and the possibility of taking into account the global variation (contour) of the duration by indicating + or – before the duration value (*directed* duration). The results obtained with four representations are presented in Tab. 4.

Duration representation	average ADR	min ADR	max ADR
absolute	0.69	0.35	0.92
interval	0.65	0.36	0.92
directed absolute	0.69	0.33	0.92
directed interval	0.68	0.35	0.92

Table 4: ADR measures (average, minimum and maximum values) obtained by the retrieval system considering different duration representations.

Results presented confirm the improvement due to the consideration of the variation of the duration: *directed* representation slightly increases the average ADR. But some other results are more surprising. The best average ADR is obtained with the *absolute* representation. However, it is important to remember that this duration representation does not permit the retrieval algorithm to be tempo invariant. This drawback should have an impact on the quality of the results. With this representation, each duration difference for a substitution is penalized. For example, a melody represented with half notes is measured as very different from the same melody represented with quarter notes. We therefore expect the *interval* representation to lead to the best results, but this limitation is justified by the MIREX data set.

These results are partly explained when observing the most similar musical pieces extracted from the database. Fig. 4 shows an example of a query and a piece from the database that are estimated very similar by our algorithm. This piece is not present in the ground truth established by music experts, whereas it is obvious that it is very similar with the query. In this case, our algorithm has a low ADR measure whereas the retrieved piece is correct. Fig. 4 also shows an incipit present in the third group of the ground truth [Typke et al., 2005], whereas the similarity is very high if the absolute duration of notes is not taken into account. Our algorithm considers this piece very similar to the query, which is also correct.

The conclusion of these experiments is that the duration representation mainly depends on the application considered. If the application is query-by-humming, similarity measures should be tempo invariant. That's why, the *directed interval* representation appears to be the best choice in this case. Otherwise, the *directed absolute* has to be chosen for applications that do not require tempo invariance. For example, music experts who established the ground truth for the MIREX 2005 training data consider the difference of tempo as an important difference to discriminate melodies. In order to be compared to these experts, we consider in the following the *directed absolute* representation.



Figure 4: Example of differences between our algorithm and the MIREX 2005 ground truth: the query (top) is very similar to the two incipits, but only the bottom one appears in the ground truth.

3.2.4 Weights for Each Component

In Sec. 3.2.2, we introduce the parameter k which determines the relative weight of the pitch difference with the duration difference. Experiments have been performed concerning the influence of this parameter. The average ADR obtained with different values for the parameter k is presented in Tab. 5.

k	average ADR	min ADR	max ADR
0.0	0.61	0.38	0.91
0.1	0.69	0.39	0.92
0.2	0.69	0.33	0.92
0.25	0.68	0.32	0.92
0.3	0.69	0.29	0.92
0.5	0.67	0.21	0.92
5.0	0.58	0.19	0.92

Table 5: ADR measures (average, minimum and maximum values) obtained by the retrieval system considering different values for the parameter k .

Results show that the influence of the parameter k is very small: the average ADR varies between 0.68 and 0.69. If k is null, no information about duration is taken into account (see Sec. 3.2.2). Thus, the average ADR highly decreases. At the opposite, if k takes high values, the duration information is considered as highly more important than the pitch information. This consideration results in lower average ADR because the pitch information is experimented as the main information for any melody. In the following, we set k to 0.25.

3.3 Retrieval Algorithm

In this section, we present experiments about the parametrization of editing algorithms.

3.3.1 Substitution Scores

In Sec. 2.2, edit operations have been presented. Substitution is the main operation and mainly determines the accuracy of the retrieval algorithm. For some applications, the substitution score is assumed to be constant. However, in the musical context, this assumption must be discussed [Uitdenbogerd, 2002]. Obviously, substituting one pitch with another one has more or less influence on the general melody. For instance, substituting a *C* note with a *G* note (fifth) may slightly modify a melody in comparison with substituting with a *D* note. As introduced by [Mongeau and Sankoff, 1990] the substitution score may be correlated to the consonance interval. We propose here to confirm the influence of this choice on the accuracy of the retrieval system. Note that determining the substitution score according to the consonance interval is totally different than determining substitution score according to the difference (in semi-tones for example). However, some models only consider such differences, for example [Typke et al., 2004].

The score due to the pitch difference is determined according to consonance: the fifth (7 semitones) and the third major or minor (3 or 4 semitones) are the most consonant intervals in Western music [Horwood, 1944]. Tab. 6 shows score values chosen during our experiments. These values slightly differ from the ones indicated by [Mongeau and Sankoff, 1990] because we choose to preserve the symmetry properties of the score: the score for a substitution between two identical notes is equal to 2.850. The score between a note and a rest has been fixed to -0.5 .

Pitch difference in semitones	Associated score
0	+2.850
1	-2.850
2	-2.475
3	-0.825
4	-0.825
5	+0.000
6	-1.800
rest	-0.500

Table 6: Scores associated to the substitution of two notes as a function of the interval between notes (in semitones), according principally to consonance.

Tab. 7 shows the ADR measures obtained by considering substitution scores according to consonance intervals. Experiments have been done by considering the parameters that give the best results (higher average ADR in both cases). These results clearly show that considering different values for the substitution scores according to the consonance interval significantly improves the retrieval system. The average ADR is 0.79, whereas it is only 0.71 when the substitution score is constant. This improvement is important because taking into account the consonance interval between notes is only possible with a few retrieval systems ([Lubiw and Tanur, 2004] for example), especially the edit-based systems.

Substitution scores	average ADR	min ADR	max ADR
constant	0.71	0.55	0.89
consonance	0.79	0.63	0.96

Table 7: ADR measures (average, minimum and maximum values) obtained by the retrieval system considering fixed substitution score or substitution scores related to consonance interval.

Substitution score can also depend on the note duration. We propose to experiment two different approaches. The first one assumes the substitution score to be dependent on the duration subtraction, estimated in sixteenth note values. This choice has been proposed by [Mongeau and Sankoff, 1990]. The second possibility considers the ratio between two consecutive note durations. Moreover, the variation between durations may also be taken into account: a score is defined according to the duration contour.

Tab. 8 shows the results obtained by the retrieval system when considering the substitution score depending on the duration subtraction or the duration ratio. Here, the substitution scores only depend on the duration information.

Duration	average ADR	min ADR	max ADR
subtraction	0.62	0.19	0.92
directed subtraction	0.63	0.19	0.92
ratio	0.68	0.35	0.92
directed ratio	0.69	0.33	0.92

Table 8: ADR measures (average, minimum and maximum values) obtained by the retrieval system considering duration subtraction or ratio, directed or not, for the calculation of the substitution scores.

These results clearly show that the duration ratio is more significant than the duration subtraction. The average ADR is 0.68 when considering duration ratio, whereas it is only 0.62 when considering duration subtraction. This observation can be easily justified with a few examples. Let us consider a melody M_1 composed of half notes, and the same melody M'_1 at the exception of one note which is substituted by a quarter note (same pitch). Thus, the duration subtraction is $16 - 8 = 8$ sixteenth notes. Now, consider the same melody M_2 with two times faster tempo, so that each note are represented by quarter notes. If this melody is compared to a melody M'_2 composed of the same notes at the exception of one note which is substituted by a eighth note, the duration subtraction is $8 - 4 = 4$ sixteenth notes. The differences between M_1 and M'_1 and the one between M_2 and M'_2 are not similar whereas, in each case, the mismatching note in M'_i is two times shorter than in M_i ($i \in \{1, 2\}$). We would expect the difference of substitution score to be the same in both cases. When considering the duration ratio, the substitution scores are the same.

These experiments indicate that the choice of considering duration ratio for the calculation of the substitution scores greatly improves the retrieval system. Existing systems like

[Mongeau and Sankoff, 1990] consider the duration subtraction, or cannot be parametrized for considering the difference ratio. Here again, the flexibility of the edit-based systems induces great improvements for musical applications, since it is possible to take into account the rhythmic information and to preserve the tempo invariance property.

Some other experiments concerning the insertion and deletion scores have been performed. One could think that considering the note duration for the calculation of the insertion/deletion scores may improve the quality of the retrieval system. Indeed, the insertion of a half note may disturb more significantly a melody than the insertion of a sixteenth note. Tab. 9 exhibits the average ADR computed by the retrieval system when the insertion/deletion score depends on the note duration. These results show that the improvement exists but is not significant. Nevertheless, in the following, the insertion/deletion scores depend on the note duration.

Duration	average ADR	min ADR	max ADR
dependent	0.72	0.43	0.94
independent	0.71	0.39	0.93

Table 9: ADR measures (average, minimum and maximum values) obtained by the retrieval system when considering the insertion/deletion operation score related to the note duration.

3.3.2 Global or Local Alignment

In Sec. 2, we discuss the different existing algorithms based on edit operations. Mainly two different techniques are proposed. The first one assumes that the pieces compared have very different lengths: similarity between local portions of pieces is estimated. The second one considers the two pieces compared in their entirety. It is obvious that considering two pieces with approximately the same length considerably retains the applications of global alignment algorithms. We performed experiments to measure the improvement due to the local alignment algorithm.

Tab. 10 presents the ADR measures computed by our retrieval system with the local and global alignment algorithms. Results show that the improvement due to the local alignment algorithm is very important, even if the MIREX training database is composed of musical incipits whose lengths are similar. In query-by-humming applications, the improvement may certainly be far more important. These results prove that the local alignment algorithm is better than the global alignment algorithm in this context.

Alignment	average ADR	min ADR	max ADR
local	0.79	0.63	0.96
global	0.55	0.34	0.91

Table 10: ADR measures (average, minimum and maximum values) obtained by the retrieval system when considering global or local alignment.

3.3.3 Query Length Weighting

In query-by-humming applications, the length of the query is assumed to be small compared to the length of the musical pieces of the database, thus the local alignment technique is generally more accurate. Nevertheless, the case of a short musical piece in the database may be investigated. Fig. 5 shows incipits from the MIREX 2005 training database. Two musical pieces (b) and (c) are estimated as very similar to the query (a). But (c) is shorter than the query and (b) is longer. Editing algorithm based on local alignment computes a small similarity score if the query is longer than the piece tested, compared to the score obtained with a long piece. Indeed, fewer notes are compared, and as the score is limited by the number of matching scores, it is thus limited by the number of notes. For example, in Fig. 5, the similarity score is higher when comparing the query (a) with the second piece (b), because the length of the piece tested is higher than the length of the query. The score is thus limited by the length of the query. At the opposite, the similarity score obtained when comparing the query with the third piece (c) is smaller.



Figure 5: Examples of query (a) and related musical pieces (b) and (c) with different lengths.

Therefore we propose to improve the editing algorithm presented by taking into account the length of the pieces. We assume that the length of the matching sequence is the minimum between the query length and the tested piece length. The score computed is then weighted by the inverse of this minimum length. Nevertheless, if a piece is composed of only a very few notes (less than five for example), we propose to ignore the piece tested.

Tab. 11 shows the results obtained by applying our editing algorithm. These results clearly underline that taking into account the length of the incipits tested improves the system. The average ADR computed is 0.79 instead of 0.72. Nevertheless, this improvement clearly depends on the application. The length of the query is generally small compared to the length of the musical pieces of the database. Therefore, in this case, weighting all the computed scores by the length of the query would not improve the retrieval system.

3.3.4 Complexity, Implementation and Time Computation

The editing algorithms we detailed in this paper have been implemented in C++ language. As explained in Sec. 2, the algorithm complexity is quadratic, in $O(|S_1| \times |S_2|)$, where $|S_1|$ and $|S_2|$

Query length weighting	average ADR	min ADR	max ADR
with	0.79	0.63	0.96
without	0.72	0.55	0.94

Table 11: ADR measures (average, minimum and maximum values) obtained by the retrieval system when considering or not the lengths of the query and the incipits.

are the lengths of the musical sequences compared. However, considering musical applications such as query-by-humming retrieval systems implies that the length of the query S_1 is very small compared to the length S_2 of the musical pieces of the database.

Tab. 12 presents computation times spent by the retrieval systems evaluated during MIREX 2005. It also indicates the computation time spent by our system. The lower computation times are generally obtained by the simplest system (N-grams, edit-distance or geometric), whereas they may be evaluated as very accurate (see next section), like the simple N-grams system proposed by [Suyoto and Uitdenbogerd, 2005a]. Furthermore, the slowest system in MIREX 2005 has been highly improved and is now very fast [Typke et al., 2006b]. These results underline that the computation time spent by our algorithm is very high compared to the other ones. It is mainly due to the score of the *maximum* algorithm presented for taking into account two pitch representations (see Sec. 3.2.1). Furthermore, the implementation can be optimised and we soon expect time computation reaches the same values as the fastest other systems.

Algorithm	Author(s)	Computation time (s)
Improved editing		110
Edit distance I/R	Grachten	80
N-grams	Orio	24
Simple N-grams	Uitdenbogerd	48
Geometric	Typke	50000
Geometric	Lemström	10
Edit distance	Lemström	10
Hybrid	Frieler	54

Table 12: Comparison of the computation times for the retrieval system proposed (top) and retrieval systems evaluated during MIREX 2005.

Improvements in editing algorithm may induce a significant increase of the time computation, whereas the accuracy due to the improvement may be limited. For example, [Mongeau and Sankoff, 1990] proposes two new operations *consolidation* and *fragmentation*. In this case, the complexity becomes cubic.

3.4 Comparisons with Existing Systems

In this section, we present the results obtained by our improved system and the results of the evaluations obtained during MIREX 2005. It is important to note that our system has not been evaluated during MIREX 2005. In order to evaluate our system without training, we then participated to MIREX 2006 symbolic melodic similarity contest: these results are presented in Sec. 3.5.

Results obtained with the MIREX 2005 training database are quite good : the average ADR obtained with our improved algorithm is 0.79 (the minimum ADR is 0.63 and the maximum ADR is 0.96). Tab. 13 shows the results obtained by the algorithms participating to the MIREX 2005. The best average ADR obtained during MIREX 2005 was 0.66.

Algorithm	Author	Rank	average ADR	min ADR	max ADR
Edit distance I/R	Grachten	1	0.66	0.29	0.88
N-grams	Orio	2	0.65	0.31	0.91
Simple N-grams	Uitdenbogerd	3	0.64	0.31	0.91
Geometric	Typke	4	0.57	0.29	0.86
Geometric	Lemström	5	0.56	0.34	0.72
Edit distance	Lemström	6	0.54	0.29	0.84
Hybrid	Frieler	7	0.52	0.34	0.81

Table 13: Results of the evaluation of retrieval systems during MIREX 2005.

3.5 Robustness to Pitch and Tempo Variations

During the 2nd Music Information Retrieval Evaluation eXchange (MIREX 2006), a new symbolic melodic similarity contest has been proposed. It was composed of two tasks. The first task consisted of retrieving the most similar incipits from the UK subset of the RISM A/II collection (about 15,000 incipits), given one of the incipits as a query. Both the query and the collection were monophonic. Half the queries were hummed or whistled queries that have been converted to MIDI, thus with slight rhythmic and pitch imperfections, and half the queries were quantized in pitch and rhythm.

At the opposite of the MIREX 2005 contest, no ground truth was established in advance. Algorithms proposed by participants computed the similarity measures and indicated the most similar musical pieces in the RISM database. These relevances of the matches were judged by experts, according to a general evaluation – not similar, somewhat similar or very similar – and one fine score from 0 to 10. Each system was then evaluated with several measures (Average Dynamic Recall for example).

Results are presented in Fig. 6. Our algorithm denoted FH obtains the best results, similar to the ones obtained by the improved geometric algorithm submitted by [Typke et al., 2006b]. The difference with the results obtained by [Uitdenbogerd, 2006], based on edit-distance, are significant and can be partly explained by the optimization of editing algorithms.

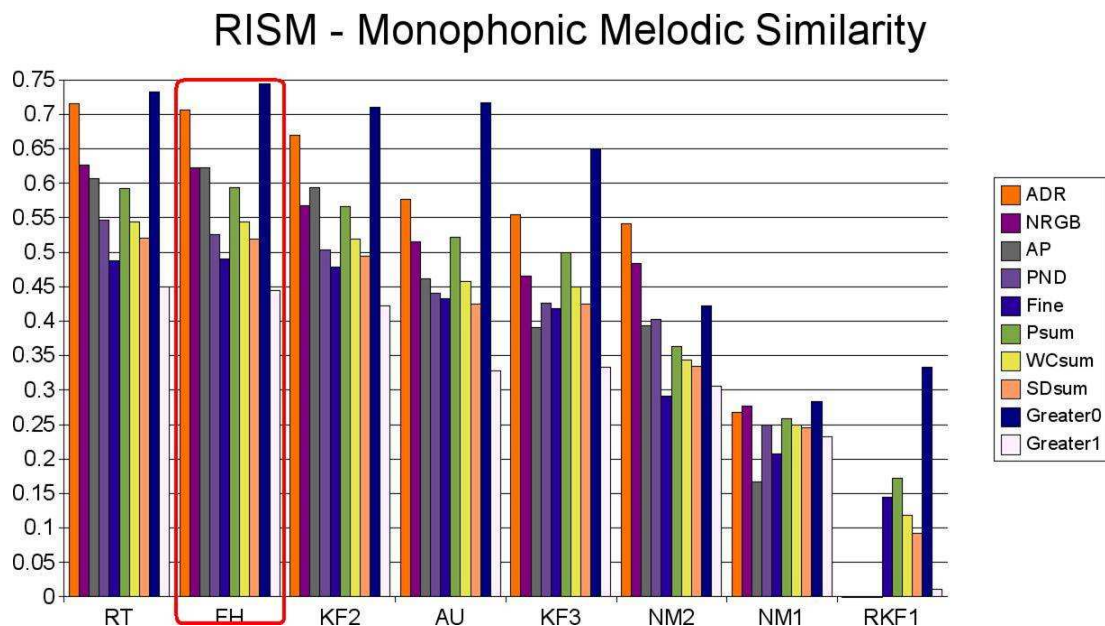


Figure 6: Results of the MIREX 2006 Symbolic Melodic Similarity task²: FH denotes the results obtained by the algorithm presented here.

Although our algorithm has been optimized according to MIREX 2005 training data, the results with the MIREX 2006 database are very good. It clearly underlines that our algorithm performs similarly whatever the database. These results also show that editing algorithms are robust to pitch and time variations. Results with hummed or whistled queries are comparable to results obtained with quantized queries. Finally, the differences with other editing algorithms show that the experiments detailed in this paper significantly improve such algorithms.

4 Conclusion and future works

Editing algorithms have already been presented for applications in the musical context. Nevertheless, to our knowledge, no complete evaluation of such algorithms has been proposed. We present in this paper several experiments in order to optimise and to adapt these editing algorithms for computing similarity measures between monophonic melodies. The representation of melodies and the editing algorithms are discussed and experimented. All the optimisations proposed lead to an algorithm which obtains the best results with MIREX 2005 training database. It also participated to the MIREX 2006 contest² and obtained the best results in the monophonic context. Differences with other editing algorithms show that the experiments detailed in this paper permit to significantly improve such algorithms. Thus, the editing algorithm that has been improved according to MIREX 2005 training data is robust to pitch and time variation and performs similarly whatever the database. Furthermore, results with hummed or whistled queries

²http://www.music-ir.org/mirex2006/index.php/MIREX2006_Results

are comparable to results obtained with quantized queries.

We can now raise the questions of improvements of such algorithms dedicated to monophonic musical pieces. Indeed, the evaluations indicate high accuracy for our retrieval system. The comparison between the results obtained with MIREX 2005 training database and the results of the similarity evaluation established by musical experts underlines that our algorithm generally computes better results than one single expert. Since the attribute of the similarity measure is subjective, it is difficult to propose a retrieval system satisfying every musical expert.

And it is certainly impossible to propose a retrieval system which satisfies every musical expert, because of the subjective attribute of the similarity measure.

The main goal of all these experiments performed is to show the flexibility of the editing algorithms. Edit operations can perfectly be adapted to the musical context. Musical elements such as tonality or rhythm are particularly important in the perception of music. Edit operations allow us to take into account musical properties of the Western music by including weights according to the musical importance of each note [Robine et al., 2007]. This flexibility also encourages us to adapt such algorithms to the polyphonic context [Hanna and Ferraro, 2007]. Indeed, new operations related to the nature of chords may be considered in the future in order to take into account all the notes that sound at the same time. This extension is necessary to propose a system which computes a measure from any kind of queries with any kind of musical pieces (monophonic or polyphonic). In particular, the system should be improved in the case of a polyphonic query with a polyphonic database. Other chord-specific edit operations are thus necessary and can be included in the system.

From an algorithmic point of view, we propose in this paper a direct application of the algorithm of [Smith and Waterman, 1981]. However, heuristics to approximate local alignment can be applied to the musical context and may lead to the reduction of time computation of local score. For instance, the *Basic Local Alignment Tool* (BLAST) [Altschul et al., 1990] is a well known searching engine for biological sequence databases. We aim to extend BLAST heuristic to musical sequences.

References

- [Altschul et al., 1990] Altschul, S., Gish, W., Miller, W., Myers, E., and Lipman, D. (1990). Basic Local Alignment Search Tool. *Journal of Molecular Biology*, 215:403–410.
- [Cambouropoulos et al., 2005] Cambouropoulos, E., Crochemore, M., Iliopoulos, C. S., Mohamed, M., and Sagot, M.-F. (2005). A Pattern Extraction Algorithm for Abstract Melodic Representations that Allow Partial Overlapping of Intervalllic Categories. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 167–174, London, UK.
- [Crawford et al., 1998] Crawford, T., Iliopoulos, C. S., and Raman, R. (1998). String Matching Techniques for Musical Similarity and Melodic Recognition. *Melodic Comparison: Concepts, Procedures, and Applications, Computing in Musicology*, 11:73–100.

- [Doraisamy and Rüger, 2003] Doraisamy, S. and Rüger, S. (2003). Robust Polyphonic Music Retrieval with N-grams. *Journal of Intelligent Information Systems*, 21(1):53–70.
- [Downie et al., 2005] Downie, J. S., West, K., Ehmann, A. F., and Vincent, E. (2005). The 2005 Music Information retrieval Evaluation Exchange (MIREX'05): Preliminary Overview. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR'05)*, pages 320–323, London, UK.
- [Gusfield, 1997] Gusfield, D. (1997). *Algorithms on Strings, Trees and Sequences - Computer Science and Computational Biology*. Cambridge University Press.
- [Hanna and Ferraro, 2007] Hanna, P. and Ferraro, P. (2007). Polyphonic Music Retrieval by Local Edition of Quotiented Sequences. In *Proceedings of the 5th International Workshop on Content-Based Multimedia Indexing (CBMI'07)*, pages 61–68, Bordeaux, France.
- [Horwood, 1944] Horwood, F. J. (1944). *The Basis of Music*. Gordon V. Thompson Limited, Toronto, Canada.
- [Lemström, 2000] Lemström, K. (2000). *String Matching Techniques for Music Retrieval*. PhD thesis, University of Helsinki, Dept. of Computer Science.
- [Lemström and Ukkonen, 2000] Lemström, K. and Ukkonen, E. (2000). Including Interval Encoding Into Edit Distance Based Music Comparison and Retrieval. In *Proceedings of the Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science (AISB'00)*, pages 53–60, Birmingham, UK.
- [Lubiw and Tanur, 2004] Lubiw, A. and Tanur, L. (2004). Pattern Matching in Polyphonic Music as a Weighted Geometric Translation Problem. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04)*, pages 289–296, Barcelona, Spain.
- [Mongeau and Sankoff, 1990] Mongeau, M. and Sankoff, D. (1990). Comparison of Musical Sequences. *Computers and the Humanities*, 24(3):161–175.
- [Needleman and Wunsch, 1970] Needleman, S. and Wunsch, C. (1970). A General Method Applicable to the Search for Similarities in the Amino Acid Sequences of Two Proteins. *Journal of Molecular Biology*, 48:443–453.
- [Rizo and Iñesta-Quereda, 2002] Rizo, D. and Iñesta-Quereda, J. (2002). Tree-Structured Representation of Melodies for Comparison and Retrieval. In *Proceedings of the 2nd International Workshop on Pattern Recognition in Information Systems (PRIS'02)*, pages 140–155, Alicante, Spain.
- [Robine et al., 2007] Robine, M., Hanna, P., and Ferraro, P. (2007). Music Similarity: Improvements of Edit-based Algorithms by Considering Music Theory. In *Proceedings of the ACM SIGMM International Workshop on Multimedia Information Retrieval (MIR)*, pages 135–141, Augsburg, Germany.

- [Sankoff and Kruskal, 1983] Sankoff, D. and Kruskal, J. B., editors (1983). *Time Wraps, Strings Edits, and Macromolecules: the Theory and Practice of Sequence Comparison*. Addison-Wesley Publishing Company Inc, University of Montreal, Canada.
- [Selfridge-Field, 1998] Selfridge-Field, E. (1998). Conceptual and Representational Issues in Melodic Comparison. *Melodic Comparison: Concepts, Procedures, and Applications, Computing in Musicology*, 11:3–64.
- [Smith and Waterman, 1981] Smith, T. and Waterman, M. (1981). Identification of Common Molecular Subsequences. *Journal of Molecular Biology*, 147:195–197.
- [Suyoto and Uitdenbogerd, 2005a] Suyoto, I. and Uitdenbogerd, A. (2005a). Simple Efficient N-gram Indexing for Effective Melody Retrieval. In *1st Music Information Retrieval Evaluation eXchange (MIREX'05)*, London, UK.
- [Suyoto and Uitdenbogerd, 2005b] Suyoto, I. S. H. and Uitdenbogerd, A. L. (2005b). Effectiveness of Note Duration Information for Music Retrieval. In *Proceedings of the 10th Database Systems for Advanced Applications Conference (DASFAA'05)*, pages 265–275, Beijing, China.
- [Typke et al., 2005] Typke, R., den Hoed, M., de Nooijer, J., Wiering, F., and Veltkamp, R. C. (2005). A Ground Truth For Half A Million Musical Incipits. *Journal of Digital Information Management*, 3(1):34–39.
- [Typke et al., 2004] Typke, R., Veltkamp, R. C., and Wiering, F. (2004). Searching Notated Polyphonic Music Using Transportation Distances. In *Proceedings of the 12th ACM Multimedia Conference (MM'04)*, pages 128–135, New-York, USA.
- [Typke et al., 2006a] Typke, R., Veltkamp, R. C., and Wiering, F. (2006a). A Measure for Evaluating Retrieval Techniques Based on Partially Ordered Ground Truth Lists. In *Proceedings of the 7th International Conference on Multimedia and Expo (ICME'06)*, pages 128–135, Toronto, Canada.
- [Typke et al., 2006b] Typke, R., Veltkamp, R. C., and Wiering, F. (2006b). MIREX Symbolic Melodic Similarity and Query by Singing/Humming. In *2nd Music Information Retrieval Evaluation eXchange (MIREX'06)*, Victoria, Canada.
- [Uitdenbogerd, 2006] Uitdenbogerd, A. (2006). Variations on Local Alignment for Specific Query Types. In *2nd Music Information Retrieval Evaluation eXchange (MIREX'06)*, Victoria, Canada.
- [Uitdenbogerd, 2002] Uitdenbogerd, A. L. (2002). *Music Information Retrieval Technology*. PhD thesis, RMIT University, Melbourne, Australia.
- [Uitdenbogerd and Zobel, 1999] Uitdenbogerd, A. L. and Zobel, J. (1999). Matching Techniques for Large Music Database. In *Proceedings of the 7th ACM International Conference on Multimedia (MM'99)*, pages 56–66, Orlando, USA.

[Ukkonen et al., 2003] Ukkonen, E., Lemström, K., and Mäkinen, V. (2003). Geometric Algorithms for Transposition Invariant Content-Based Music Retrieval. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR'03)*, pages 193–199, Baltimore, USA.

[Wagner and Fisher, 1974] Wagner, R. A. and Fisher, M. J. (1974). The String-to-String Correction Problem. *Journal of the association for computing machinery*, 21:168–173.