Melodic Similarity through Shape Similarity

Julián Urbano, Juan Lloréns, Jorge Morato, and Sonia Sánchez-Cuadrado

University Carlos III of Madrid
Department of Computer Science
Avda. Universidad, 30
28911 Leganés, Madrid, Spain
{jurbano,llorens}@inf.uc3m.es,
{jorge,ssanchec}@ie.inf.uc3m.es

Abstract. We present a new geometric model to compute the melodic similarity of symbolic musical pieces. Melodies are represented as splines in the pitch-time plane, and their similarity is computed as the similarity of their shape. The model is very intuitive and it is transposition and time scale invariant. We have implemented it with a local alignment algorithm over sequences of n-grams that define spline spans. An evaluation with the MIREX 2005 collections shows that the model performs very well, obtaining the best effectiveness scores ever reported for these collections. Three systems based on this new model were evaluated in MIREX 2010, and the three systems obtained the best results.

Keywords: Music information retrieval, melodic similarity, interpolation.

1 Introduction

The problem of Symbolic Melodic Similarity, where musical pieces similar to a query should be retrieved, has been approached from very different points of view [24][6]. Some techniques are based on string representations of music and editing distance algorithms to measure the similarity between two pieces[17]. Later work has extended this approach with other dynamic programming algorithms to compute global- or local-alignments between the two musical pieces [19][11][12]. Other methods rely on music representations based on n-grams [25][8][2], and other methods represent music pieces as geometric objects, using different techniques to calculate the melodic similarity based on the geometric similarity of the two objects. Some of these geometric methods represent music pieces as sets of points in the pitch-time plane, and then compute geometric similarities between these sets [26][23][7]. Others represent music pieces as orthogonal polynomial chains crossing the set of pitch-time points, and then measure the similarity as the minimum area between the two chains [30][1][15].

In this paper we present a new model to compare melodic pieces. We adapted the local alignment approach to work with n-grams instead of with single notes, and the corresponding substitution score function between n-grams was also adapted to take into consideration a new geometric representation of musical sequences. In this geometric representation, we model music pieces as curves in the pitch-time plane, and compare them in terms of their shape similarity.

K. Jensen et al. (Eds.): CMMR 2010, LNCS 6684, pp. 338–355, 2011.

In the next section we outline several problems that a symbolic music retrieval system should address, and then we discuss the general solutions given in the literature to these requirements. Next, we introduce our geometric representation model, which compares two musical pieces by their shape, and see how this model addresses the requirements discussed. In section 5 we describe how we have implemented our model, and in section 6 we evaluate it with the training and evaluation test collections used in the MIREX 2005 Symbolic Melodic Similarity task (for short, we will refer to these collections as *Train05* and *Eval05*) [10][21][28]. Finally, we finish with conclusions and lines for further research. An appendix reports more evaluation results at the end.

2 Melodic Similarity Requirements

Due to the nature of the information treated in Symbolic Melodic Similarity[18], there are some requirements that have to be considered from the very beginning when devising a retrieval system. Byrd and Crawford identified some requirements that they consider every MIR system should meet, such as the need of cross-voice matching, polyphonic queries or the clear necessity of taking into account both the horizontal and vertical dimensions of music[5].Selfridge-Field identified three elements that may confound both the users when they specify the query and the actual retrieval systems at the time of computing the similarity between two music pieces: rests, repeated notes and grace notes [18]. In terms of cross-voice and polyphonic material, she found five types of melody considered difficult to handle: compound, self-accompanying, submerged, roving and distributed melodies. Mongeau and Sankoff addressed repeated notes and refer to these situations as fragmentation and consolidation [17].

We list here some more general requirements that should be common to any Symbolic Melodic Similarity system, as we consider them basic for the general user needs. These requirements are divided in two categories: vertical (i.e. pitch) and horizontal (i.e. time).

2.1 Vertical Requirements

Vertical requirements regard the pitch dimension of music: octave equivalence, degree equality, note equality, pitch variation, harmonic similarity and voice separation. A retrieval model that meets the first three requirements is usually regarded as transposition invariant.

2.1.1 Octave Equivalence

When two pieces differ only in the octave they are written in, they should be considered the same one in terms of melodic similarity. Such a case is shown in Fig. 1, with simple versions of the main riff in *Layla*, by *Dereck and the Dominos*.

It has been pointed out that faculty or music students may want to retrieve pieces within some certain pitch range such as C5 up to F#3, or every work above A5 [13].However, this type of information need should be easily handled with

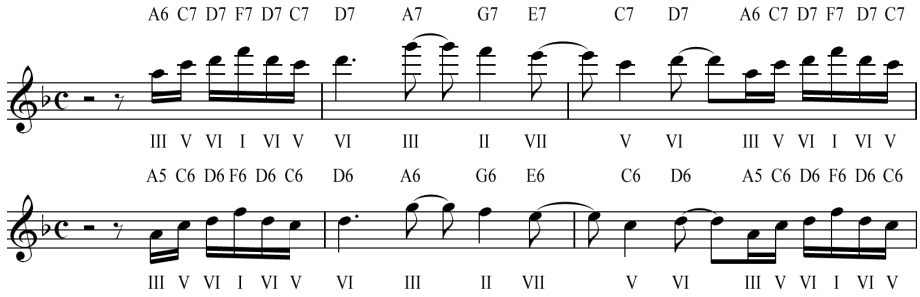

Fig. 1. Octave equivalence

metadata or a simple traverse through the sequence. We argue that users without such a strong musical background will be interested in the recognition of a certain pitch contour, and such cases are much more troublesome because some measure of melodic similarity has to be calculated. This is the case of query by humming applications.

2.1.2 Degree Equality

The score at the top of Fig. 1 shows a melody in the F major tonality, as well as the corresponding pitch and tonality-degree for each note. Below, Fig. 2 shows exactly the same melody shifted 7 semitones downwards to the B^b major tonality.

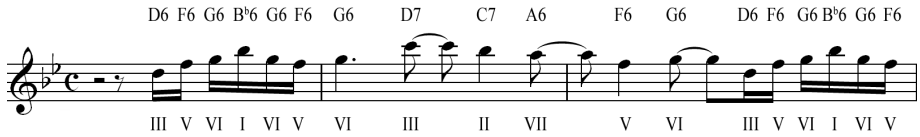

Fig. 2. Degree equality

The tonality-degrees used in both cases are the same, but the resultant notes are not. Nonetheless, one would consider the second melody a version of the first one, because they are the same in terms of pitch contour. Therefore, they should be considered the same one by a retrieval system, which should also consider possible modulations where the key changes somewhere throughout the song.

2.1.3 Note Equality

We could also consider the case where exactly the same melodies, with exactly the same notes, are written in different tonalities and, therefore, each note corresponds to a different tonality-degree in each case. Fig. 3 shows such a case, with the same melody as in Fig. 1, but in the C major tonality.

Although the degrees do not correspond one to each other, the actual notes do, so both pieces should be considered the same one in terms of melodic similarity.

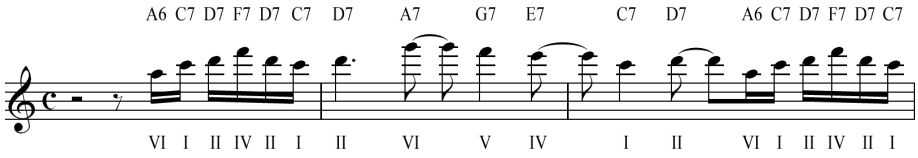

Fig. 3. Note equality

2.1.4 Pitch Variation

Sometimes, a melody is altered by changing only the pitch of a few particular notes. For instance, the first melody in Fig. 1 might be changed by shifting the 12th note from D7 to A6 (which actually happens in the original song). Such a change should not make a retrieval system disregard that result, but simply rank it lower, after the exactly-equal ones. Thus, the retrieval process should not consider only exact matching, where the query is part of a piece in the repository (or the other way around). Approximate matching, where documents can be considered similar to a query to some degree, should be the way to go. This is of particular interest for scenarios like query by humming, where it is expected to have slight variations in pitch in the melody hummed by the user.

2.1.5 Harmonic Similarity

Another desired feature would be to match harmonic pieces, both with harmonic and melodic counterparts. For instance, in a triad chord (made up by the root note and its major third and perfect fifth intervals), one might recognize only two notes (typically the root and the perfect fifth). However, some other might recognize the root and the major third, or just the root, or even consider them as part of a 4-note chord such as a major seventh chord (which adds a major seventh interval). Fig. 4 shows the same piece as in the top of Fig. 1, but with some intervals added to make the song more harmonic. These two pieces have basically the same pitch progression, but with some ornamentation, and they should be regarded as very similar by a retrieval system.

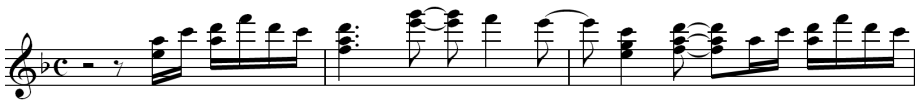

Fig. 4. Harmonic similarity

Thus, a system should be able to compare harmony wholly and partially, considering again the Pitch Variation problem as a basis to establish differences between songs.

2.1.6 Voice Separation

Fig. 5 below depicts a piano piece with 3 voices, which work together as a whole, but could also be treated individually.

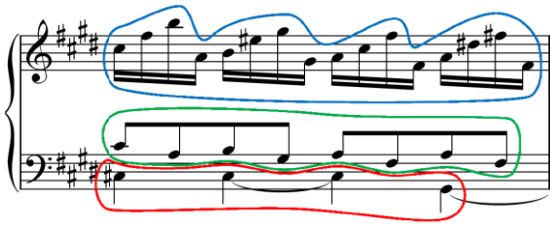

Fig. 5. Voice separation

Indeed, if this piece were played with a flute only one voice could be performed, even if some streaming effect were produced by changing tempo and timbre for two voices to be perceived by a listener [16]. Therefore, a query containing only one voice should match with this piece in case that voice is similar enough to any of the three marked in the figure.

2.2 Horizontal Requirements

Horizontal requirements regard the time dimension of music: time signature equivalence, tempo equivalence, duration equality and duration variation. A retrieval model that meets the second and third requirements is usually regarded as time scale invariant.

2.2.1 Time Signature Equivalence

The top of Fig. 6 depicts a simplified version of the beginning of *op. 81 no. 10* by *S. Heller,* with its original 2/4 time signature. If a 4/4 time signature were used, like in the bottom of Fig. 6, the piece would be split into bars of duration 4 crotchets each.

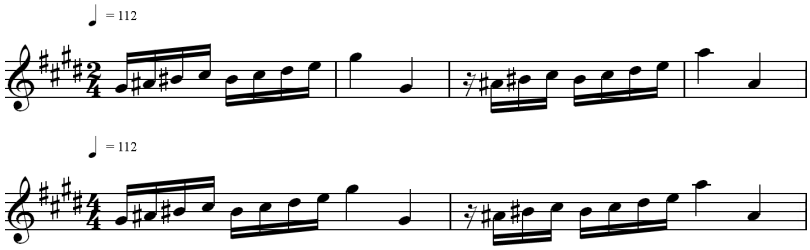

Fig. 6. Time signature equivalence

The only difference between these two pieces is actually how intense some notes should be played. However, they are in essence the same piece, and no regular listener would tell the difference. Therefore, we believe the time signature should not be considered when comparing musical performances in terms of melodic similarity.

2.2.2 Tempo Equivalence

For most people, the piece at the top of Fig. 6, with a tempo of 112 crotchets per minute, would sound like the one in Fig. 7, where notes have twice the length but the

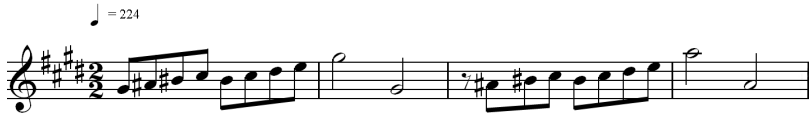

Fig. 7. Tempo equivalence

whole score is played twice as fast, at 224 crotchets per minute. This two changes result in exactly the same actual time.

On the other hand, it might also be considered a tempo of 56 crotchets per minute and notes with half the duration. Moreover, the tempo can change somewhere in the middle of the melody, and therefore change the actual time of each note afterwards. Therefore, actual note lengths cannot be considered as the only horizontal measure, because these three pieces would sound the same to any listener.

2.2.3 Duration Equality

If the melody at the top of Fig. 6 were played slower or quicker by means of a tempo variation, but maintaining the rhythm, an example of the result would be like the score in Fig. 8.

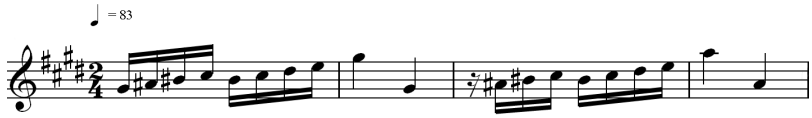

Fig. 8. Duration equality

Even though the melodic perception does actually change, the rhythm does not, and neither does the pitch contour. Therefore, they should be considered as virtually the same, maybe with some degree of dissimilarity based on the tempo variation.

2.2.4 Duration Variation

As with the Pitch Variation problem, sometimes a melody is altered by changing only the rhythm of a few notes. For instance, the melody in Fig. 9 maintains the same pitch contour as in Fig. 6, but changes the duration of some notes.

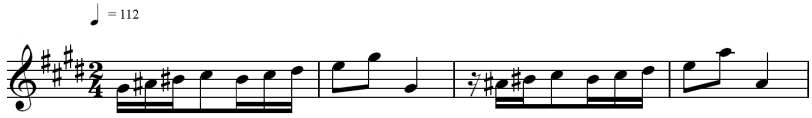

Fig. 9. Duration variation

Variations like these are common and they should be considered as well, just like the Pitch Variation problem, allowing approximate matches instead of just exact ones.

3 General Solutions to the Requirements

Most of these problems have been already addressed in the literature. Next, we describe and evaluate the most used and accepted solutions.

3.1 Vertical Requirements

The immediate solution to the Octave Equivalence problem is to consider octave numbers with their relative variation within the piece. Surely, a progression from G5 to C6 is not the same as a progression from G5 to C5. For the Degree Equality problem it seems to be clear that tonality degrees must be used, rather than actual pitch values, in order to compare two melodies. However, the Note Equality problem suggests the opposite.

The accepted solution for these three vertical problems seems to be the use of relative pitch differences as the units for the comparison, instead of the actual pitch or degree values. Some approaches consider pitch intervals between two successive notes [11][8][15], between each note and the tonic (assuming the key is known and failing to meet the Note Equality problem) [17], or a mixture of both [11]. Others compute similarities without pitch intervals, but allowing vertical translations in the time dimension [1][19][30]. The Voice Separation problem is usually assumed to be solved in a previous stage, as the input to these systems uses to be a single melodic sequence. There are approximations to solve this problem [25][14].

3.2 Horizontal Requirements

Although the time signature of a performance is worth for other purposes such as pattern search or score alignment, it seems to us that it should not be considered at all when comparing two pieces melodically.

According to the Tempo Equivalence problem, actual time should be considered rather than score time, since it would be probably easier for a regular user to provide actual rhythm information. On the other hand, the Duration Equality problem requires the score time to be used instead. Thus, it seems that both measures have to be taken into account. The actual time is valuable for most users without a musical background, while the score time might be more valuable for people who do have it.

However, when facing the Duration Variation problem it seems necessary to use some sort of timeless model. The solution could be to compare both actual and score time [11], or to use relative differences between notes, in this case with the ratio between two notes' durations [8]. Other approaches use a rhythmical framework to represent note durations as multiples of a base score duration [2][19][23], which does not meet the Tempo Equivalence problem and hence is not time scale invariant.

4 A Model Based on Interpolation

We developed a new geometric model that represents musical pieces with curves in the pitch-time plane, extending the model with orthogonal polynomial chains [30][1][15]. Notes are represented as points in the pitch-time plane, with positions

relative to their pitch and duration differences. Then, we define the curve *C(t)* as the interpolating curve passing through each point (see Fig. 10). Should the song have multiple voices, each one would be placed in a different pitch-time plane, sharing the same time dimension, but with a different curve $C_i(t)$ (where the subscript *i* indicates the voice number). Note that we thus assume the voices are already separated.

With this representation, the similarity between two songs could be thought of as the similarity in shape between the two curves they define. Every vertical requirement identified in section 2.1would be met with this representation: a song with an octave shift would keep the same shape; if the tonality changed the shape of the curve would not be affected either; and if the notes remained the same after a tonality change, so would the curve do. The Pitch Variation problem can be addressed analytically measuring the curvature difference, and different voices can be compared individually in the same way because they are in different planes.

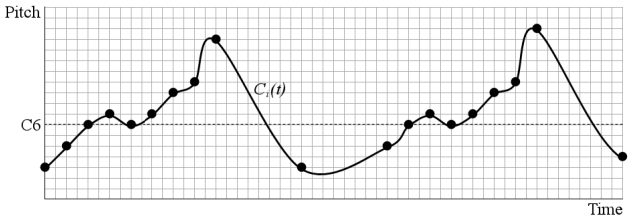

Fig. 10. Melody represented as a curve in a pitch-time plane

Same thing happens with the horizontal requirements: the Tempo Equivalence and Duration Equality problems can be solved analytically, because they imply just a linear transformation in the time dimension. For example, if the melody at the top of Fig. 6 is defined with curve *C(t)* and the one in Fig. 7 is denoted with curve *D(t)*, it can be easily proved that $C(2t)=D(t)$. Moreover, the Duration Variation problem could be addressed analytically as the Pitch Variation problem, and the Time Signature Equivalence problem is not an issue because the shape of the curve is independent of the time signature.

4.1 Measuring Dissimilarity with the Change in Shape

Having musical pieces represented with curves, each one of them could be defined with a polynomial of the form $C(t)=a_nt^n+a_{n-1}t^{n-1}+\ldots+a_1t+a_0$. The first derivative of this polynomial measures how much the shape of the curve is changing at a particular point in time (i.e. how the song changes). To measure the change of one curve with respect to another, the area between the first derivatives could be used.

Note that a shift in pitch would mean just a shift in the a_0 term. As it turns out, when calculating the first derivative of the curves this term is canceled, which is why the vertical requirements are met: shifts in pitch are not reflected in the shape of the curve, so they are not reflected in the first derivative either. Therefore, this representation is transposition invariant.

The song is actually defined by the first derivative of its interpolating curve, *C'(t)*. The dissimilarity between two songs, say *C(t)* and *D(t)*, would be defined as the area

between their first derivatives (measured with the integral over the absolute value of their difference):

$$diff(C, D) = \int |C'(t)-D'(t)|dt \quad (1)$$

The representation with orthogonal polynomial chains also led to the measurement of dissimilarity as the area between the curves [30][1]. However, such representation is not directly transposition invariant unless it used pitch intervals instead of absolute pitch values, and a more complex algorithm is needed to overcome this problem[15]. As orthogonal chains are not differentiable, this would be the indirect equivalent to calculating the first derivative as we do.

This dissimilarity measurement based on the area between curves turns out to be a metric function, because it has the following properties:

- Non-negativity, $diff(C, D) \geq 0$: because the absolute value is never negative.
- Identity of indiscernibles, $diff(C, D) = 0 \Leftrightarrow C = D$: because calculating the absolute value the only way to have no difference is with the same exact curve[1].
- Symmetry, $diff(C, D) = diff(D, C)$: again, because the integral is over the absolute value of the difference.
- Triangle inequality, $diff(C, E) \leq diff(C, D) + diff(D, E)$:

$$\int |C'(t) - E'(t)|dt \leq \int |C'(t) - D'(t)|dt + \int |D'(t) - E'(t)|dt$$

$$\int |C'(t) - D'(t)|dt + \int |D'(t) - E'(t)|dt = \int |C'(t) - D'(t)| + |D'(t) - E'(t)|dt$$

$$\int |C'(t) - D'(t)| + |D'(t) - E'(t)|dt \geq \int |C'(t) - E'(t)|\, dt$$

Therefore, many indexing and retrieval techniques, like vantage objects[4], could be exploited if using this metric.

4.2 Interpolation with Splines

The next issue to address is the interpolation method to use. The standard Lagrange interpolation method, though simple, is known to suffer the Runge's Phenomenon [3]. As the number of points increases, the interpolating curve wiggles a lot, especially at the beginning and the end of the curve. As such, one curve would be very different from another one having just one more point at the end, the shape would be different and so the dissimilarity metric would result in a difference when the two curves are practically identical. Moreover, a very small difference in one of the points could translate into an extreme variation in the overall curve, which would make virtually impossible to handle the Pitch and Duration Variation problems properly (see top of Fig. 11).

[1] Actually, this means that the first derivatives are the same, the actual curves could still be shifted. Nonetheless, this is the behavior we want.

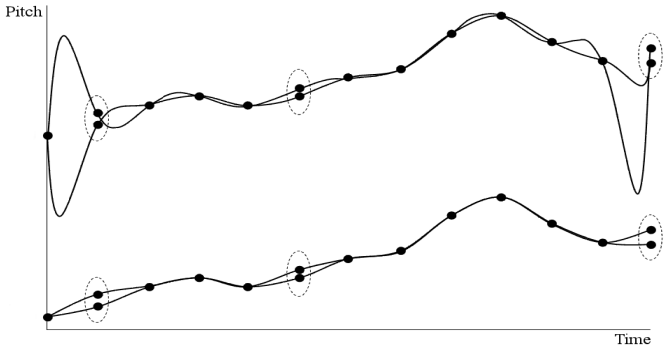

Fig. 11. Runge's Phenomenon

A way around Runge's Phenomenon is the use of splines (see bottom of Fig. 11). Besides, splines are also easy to calculate and they are defined as piece-wise functions, which comes in handy when addressing the horizontal requirements. We saw above that the horizontal problems could be solved, as they implied just a linear transformation of the form $D(t) \Rightarrow D(kt)$ in one of the curves. However, the calculation of the term k is anything but straightforward, and the transformation would apply to the whole curve, complicating the measurement of differences for the Duration Variation problem. The solution would be to split the curve into spans, and define it as

$$C_i(t) = \begin{cases} c_{i,1}(t) & t_{i,1} \leq t \leq t_{i,k_n} \\ c_{i,2}(t) & t_{i,2} \leq t \leq t_{i,k_n+1} \\ \vdots & \vdots \\ c_{i,m_i-k_n+1}(t) & t_{i,m_i-k_n+1} \leq t \leq t_{i,m_i} \end{cases} \quad (2)$$

where $t_{i,j}$ denotes the onset time of the *j-th* note in the *i-th* voice, m_i is the length of the *i-th* voice, and k_n is the span length. With this representation, linear transformations would be applied only to a single span without affecting the whole curve. Moreover, the duration of the spans could be normalized from *0* to *1*, making it easy to calculate the term k and comply with the time scale invariance requirements.

Most spline interpolation methods define the curve in parametric form (i.e. with one function per dimension). In this case, it results in one function for the pitch and one function for the time. This means that the two musical dimensions could be compared separately, giving more weight to one or the other. Therefore, the dissimilarity between two spans $c(t)$ and $d(t)$ would be the sum of the pitch and time dissimilarities as measured by (1):

$$diff(c, d) = k_p diff_p(c, d) + k_t diff_t(c, d) \quad (3)$$

where $diff_p$ and $diff_t$ are functions as in (1) that consider only the pitch and time dimensions, respectively, and k_p and k_t are fine tuning constants. Different works suggest that pitch is much more important than time for comparing melodic similarity, so more weight should be given to k_p [19][5][8][23][11].

5 Implementation

Geometric representations of music pieces are very intuitive, but they are not necessarily easy to implement. We could follow the approach of moving one curve towards the other looking for the minimum area between them [1][15]. However, this approach is very sensitive to small differences in the middle of a song, such as repeated notes: if a single note were added or removed from a melody, it would be impossible to fully match the original melody from that note to the end. Instead, we follow a dynamic programming approach to find an alignment between the two melodies [19].

Various approaches for melodic similarity have applied editing distance algorithms upon textual representations of musical sequences that assign one character to each interval or each n-gram [8]. This dissimilarity measure has been improved in recent years, and sequence alignment algorithms have proved to perform better than simple editing distance algorithms [11][12]. Next, we describe the representation and alignment method we use.

5.1 Melody Representation

To practically apply our model, we followed a basic n-gram approach, where each n-gram represents one span of the spline. The pitch of each note was represented as the relative difference to the pitch of the first note in the n-gram, and the duration was represented as the ratio to the duration of the whole n-gram. For example, an n-gram of length 4 with absolute pitches ⟨74, 81, 72, 76⟩ and absolute durations ⟨240, 480, 240, 720⟩, would be modeled as ⟨81-74, 72-74, 76-74⟩ = ⟨7, -2, 2⟩ in terms of pitch and ⟨240, 480, 240, 720⟩/1680 = ⟨0.1429, 0.2857, 0.1429, 0.4286⟩ in terms of duration. Note that the first note is omitted in the pitch representation as it is always 0.

This representation is transposition invariant because a melody shifted in the pitch dimension maintains the same relative pitch intervals. It is also time scale invariant because the durations are expressed as their relative duration within the span, and so they remain the same in the face of tempo and actual or score duration changes. This is of particular interest for query by humming applications and unquantized pieces, as small variations in duration would have negligible effects on the ratios.

We used Uniform B-Splines as interpolation method [3]. This results in a parametric polynomial function for each n-gram. In particular, an n-gram of length k_n results in a polynomial of degree k_n-1 for the pitch dimension and a polynomial of degree k_n-1 for the time dimension. Because the actual representation uses the first derivatives, each polynomial is actually of degree k_n-2.

5.2 Melody Alignment

We used the Smith-Waterman local alignment algorithm [20], with the two sequences of overlapping spans as input, defined as in (2). Therefore, the input symbols to the alignment algorithm are actually the parametric pitch and time functions of a span,

based on the above representation of n-grams. The edit operations we define for the Smith-Waterman algorithm are as follows:

- Insertion: $s(-, c)$.Adding a span c is penalized with the score–$diff(c, \phi(c))$.
- Deletion: $s(c, -)$. Deleting a span c is penalized with the score–$diff(c, \phi(c))$.
- Substitution: $s(c, d)$. Substituting a span c with d is penalized with $-diff(c, d)$.
- Match: $s(c, c)$. Matching a span c is rewarded with the score $2(k_p\mu_p+k_t\mu_t)$.

where $\phi(\bullet)$ returns the null n-gram of • (i.e. an n-gram equal to • but with all pitch intervals set to 0), and μ_p and μ_t are the mean differences calculated by $diff_p$ and $diff_t$ respectively over a random sample of 100,000 pairs of n-grams sampled from the set of incipits in the *Train05* collection.

We also normalized the dissimilarity scores returned by $diff_t$. From the results in Table 1 it can be seen that pitch dissimilarity scores are between 5 and 7 times larger than time dissimilarity scores. Therefore, the choice of k_p and k_t does not intuitively reflect the actual weight given to the pitch and time dimensions. For instance, the selection of k_t=0.25, chosen in studies like [11], would result in an actual weight between 0.05 and 0.0357. To avoid this effect, we normalized every time dissimilarity score multiplying it by a factor $\lambda = \mu_p / \mu_t$. As such, the score of the match operation is actually defined as $s(c, c) = 2\mu_p(k_p+k_t)$, and the dissimilarity function defined in (3) is actually calculated as $diff(c, d) = k_p\, diff_p(c, d) + \lambda k_t diff_t(c, d)$.

6 Experimental Results[2]

We evaluated the model proposed with the *Train05* and *Eval05* test collections used in the MIREX 2005 Symbolic Melodic Similarity Task [21][10], measuring the mean Average Dynamic Recall score across queries [22]. Both collections consist of about 580 incipits and 11 queries each, with their corresponding ground truths. Each ground truth is a list of all incipits similar to each query, according to a panel of experts, and with groups of incipits considered equally similar to the query.

However, we have recently showed that these lists have inconsistencies whereby incipits judged as equally similar by the experts are not in the same similarity group and vice versa [28]. All these inconsistencies result in a very permissive evaluation where a system could return incipits not similar to the query and still be rewarded for it. Thus, results reported with these lists are actually overestimated, by as much as 12% in the case of the MIREX 2005 evaluation. We have proposed alternatives to arrange the similarity groups for each query, proving that the new arrangements are significantly more consistent than the original one, leading to a more robust evaluation. The most consistent ground truth lists were those called *Any-1* [28]. Therefore, we will use these *Any-1* ground truth lists from this point on to evaluate our model, as they offer more reliable results. Nonetheless, all results are reported in an appendix as if using the original ground truths employed in MIREX 2005, called *All-2*, for the sake of comparison with previous results.

[2] All system outputs and ground truth lists used in this paper can be downloaded from http://julian-urbano.info/publications/

To determine the value of the k_n and k_t parameters, we used a full factorial experimental design. We tested our model with n-gram lengths in the range $k_n \in \{3, 4, 5, 6, 7\}$, which result in Uniform B-Spline polynomials of degrees 1 to 5. The value of k_p was kept to 1, and k_t was converted to nominal with levels $k_t \in \{0, 0.1, 0.2, \ldots, 1\}$.

6.1 Normalization Factor λ

First, we calculated the mean dissimilarity scores μ_p and μ_t for each n-gram length k_n, according to $diff_p$ and $diff_t$ over a random sample of 100,000 pairs of n-grams. Table 1 lists the results. As mentioned, the pitch dissimilarity scores are between 5 and 7 times larger than the time dissimilarity scores, suggesting the use of the normalization factor λ defined above.

Table 1. Mean and standard deviation of the $diff_p$ and $diff_t$ functions applied upon a random sample of 100,000 pairs of n-grams of different sizes

k_n	μ_p	σ_p	μ_t	σ_t	$\lambda = \mu_p / \mu_t$
3	2.8082	1.6406	0.5653	0.6074	4.9676
4	2.5019	1.6873	0.494	0.5417	5.0646
5	2.2901	1.4568	0.4325	0.458	5.2950
6	2.1347	1.4278	0.3799	0.3897	5.6191
7	2.0223	1.3303	0.2863	0.2908	7.0636

There also appears to be a negative correlation between the n-gram length and the dissimilarity scores. This is caused by the degree of the polynomials defining the splines: high-degree polynomials fit the points more smoothly than low-degree ones. Polynomials of low degree tend to wiggle more, and so their derivatives are more pronounced and lead to larger areas between curves.

6.2 Evaluation with the *Train05* Test Collection, *Any-1* Ground Truth Lists

The experimental design results in 55 trials for the 5 different levels of k_n and the 11 different levels of k_t. All these trials were performed with the *Train05* test collection, ground truths aggregated with the *Any-1* function [28]. Table 2 shows the results.

In general, large n-grams tend to perform worse. This could probably be explained by the fact that large n-grams define the splines with smoother functions, and the differences in shape may be too small to discriminate musically perceptual differences. However, k_n=3 seems to be the exception (see Fig. 12). This is probably caused by the extremely low degree of the derivative polynomials. N-grams of length k_n=3 result in splines defined with polynomials of degree 2, which are then differentiated and result in polynomials of degree 1. That is, they are just straight lines, and so a small difference in shape can turn into a relatively large dissimilarity score when measuring the area.

Overall, k_n=4 and k_n=5 seem to perform the best, although k_n=4 is more stable across levels of k_t. In fact, k_n=4 and k_t=0.6 obtain the best score, 0.7215. This result agrees with other studies where n-grams of length 4 and 5 were also found to perform

Table 2. Mean ADR scores for each combination of k_n and k_t with the *Train05* test collection, ground truth lists aggregated with the *Any-1* function. k_p is kept to 1. Bold face for largest scores per row and italics for largest scores per column.

k_n	k_t=0	k_t=0.1	k_t=0.2	k_t=0.3	k_t=0.4	k_t=0.5	k_t=0.6	k_t=0.7	k_t=0.8	k_t=0.9	k_t=1
3	0.6961	0.7067	0.7107	0.7106	0.7102	0.7109	**0.7148**	0.711	0.7089	0.7045	0.6962
4	0.7046	0.7126	0.7153	0.7147	0.7133	*0.72*	***0.7215***	*0.7202*	*0.7128*	*0.7136*	*0.709*
5	0.7093	0.7125	*0.7191*	***0.72***	*0.7173*	0.7108	0.704	0.6978	0.6963	0.6973	0.6866
6	***0.714***	*0.7132*	0.7115	0.7088	0.7008	0.693	0.6915	0.6874	0.682	0.6765	0.6763
7	0.6823	**0.6867**	0.6806	0.6747	0.6538	0.6544	0.6529	0.6517	0.6484	0.6465	0.6432

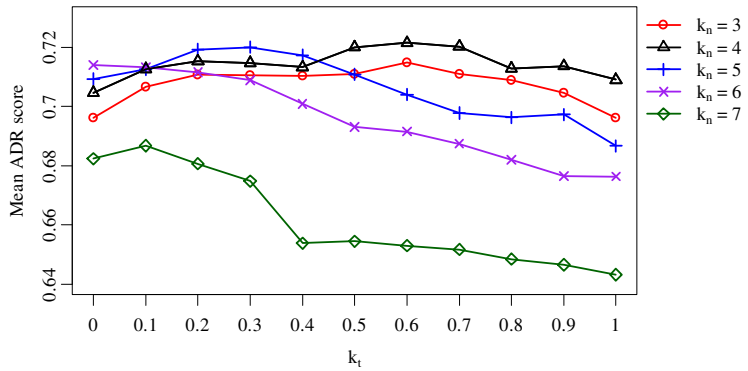

Fig. 12. Mean ADR scores for each combination of k_n and k_t with the *Train05* test collection, ground truth lists aggregated with the *Any-1* function. k_p is kept to 1

better [8]. Moreover, this combination of parameters obtains a mean ADR score of 0.8039 when evaluated with the original *All-2* ground truths (see Appendix). This is the best score ever reported for this collection.

6.3 Evaluation with the *Eval05* Test Collection, *Any-1* Ground Truth Lists

In a fair evaluation scenario, we would use the previous experiment to train our system and choose the values of k_n and k_t that seem to perform the best (in particular, k_n=4 and k_t=0.6). Then, the system would be run and evaluated with a different collection to assess the external validity of the results and try to avoid overfitting to the training collection. For the sake of completeness, here we show the results for all 55 combinations of the parameters with the *Eval05* test collection used in MIREX 2005, again aggregated with the *Any-1* function [28].Table 3 shows the results.

Unlike the previous experiment with the *Train05* test collection, in this case the variation across levels of k_t is smaller (the mean standard deviation is twice as much in *Train05*), indicating that the use of the time dimension does not provide better results overall (see Fig. 13). This is probably caused by the particular queries in each collection. Seven of the eleven queries in *Train05* start with long rests, while this

Table 3. Mean ADR scores for each combination of k_n and k_t with the *Eval05* test collection, ground truth lists aggregated with the *Any-1* function. k_p is kept to 1. Bold face for largest scores per row and italics for largest scores per column.

k_n	k_t=0	k_t=0.1	k_t=0.2	k_t=0.3	k_t=0.4	k_t=0.5	k_t=0.6	k_t=0.7	k_t=0.8	k_t=0.9	k_t=1
3	0.6522	*0.6601*	***0.6646***	0.6612	*0.664*	0.6539	0.6566	*0.6576*	*0.6591*	0.6606	*0.662*
4	*0.653*	0.653	0.6567	*0.6616*	0.6629	***0.6633***	*0.6617*	0.6569	0.65	*0.663*	0.6531
5	0.6413	0.6367	0.6327	0.6303	0.6284	0.6328	**0.6478**	0.6461	0.6419	0.6414	**0.6478**
6	0.6269	0.6251	0.6225	0.6168	0.6216	**0.6284**	0.6255	0.6192	0.6173	0.6144	0.6243
7	0.5958	**0.623**	0.6189	0.6163	0.6162	0.6192	0.6215	0.6174	0.6148	0.6112	0.6106

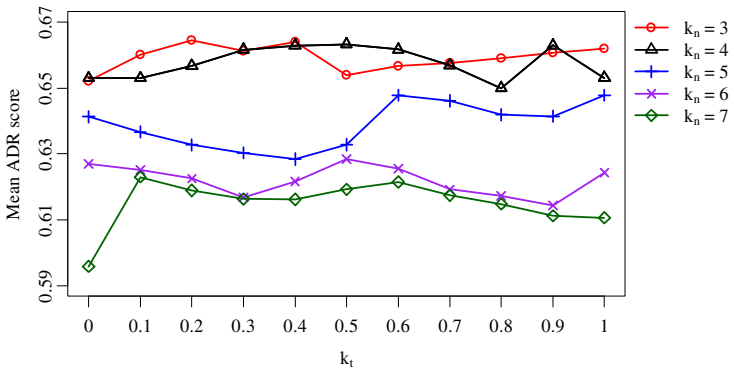

Fig. 13. Mean ADR scores for each combination of k_n and k_t with the *Eval05* test collection, ground truth lists aggregated with the *Any-1* function. k_p is kept to 1

happens for only three of the eleven queries in *Eval05*. In our model, rests are ignored, and so the effect of the time dimension is larger when the very queries have rests as their duration is added to the next note's.

Likewise, large n-grams tend to perform worse. In this case though, n-grams of length k_n=3 and k_n=4 perform the best. The most effective combination is k_n=3 and k_t=0.2, with a mean ADR score of 0.6646. However, k_n=4 and k_t=0.5 is very close, with a mean ADR score of 0.6633. Therefore, based on the results of the previous experiment and the results in this one, we believe that k_n=4 and $k_t \in [0.5, 0.6]$ are the best parameters overall.

It is also important to note that none of the 55 combinations ran result in a mean ADR score less than 0.594, which was the highest score achieved in the actual MIREX 2005 evaluation with the *Any-1* ground truths [28]. Therefore, our systems would have ranked first if participated.

7 Conclusions and Future Work

We have proposed a new transposition and time scale invariant model to represent musical pieces and compute their melodic similarity. Songs are considered as curves in the pitch-time plane, allowing us to compute their melodic similarity in terms of the shape similarity of the curves they define. We have implemented it with a local

alignment algorithm over sequences of spline spans, each of which is represented by one polynomial for the pitch dimension and another polynomial for the time dimension. This parametric representation of melodies permits the application of a weight scheme between pitch and time dissimilarities.

The MIREX 2005 test collections have been used to evaluate the model for several span lengths and weight schemes. Overall, spans 4 notes long seem to perform the best, with longer spans performing gradually worse. The optimal weigh scheme we found gives about twice as much importance to the pitch dimension than to the time dimension. However, time dissimilarities need to be normalized, as they are shown to be about five times smaller than pitch dissimilarities.

This model obtains the best mean ADR score ever reported for the MIREX 2005 training collection, and every span length and weight scheme evaluated would have ranked first in the actual evaluation of that edition. However, the use of the time dimension did not improve the results significantly for the evaluation collection. On the other hand, three systems derived from this model were submitted to the MIREX 2010 edition: *PitchDeriv*, *ParamDeriv* and *Shape* [27]. These systems obtained the best results, and they ranked the top three in this edition. Again, the use of the time dimension was not shown to improve the results.

A rough analysis of the MIREX 2005 and 2010 collections shows that the queries used in the 2005 training collection have significantly more rests than in the evaluation collection, and they are virtually absent in the 2010 collection. Because our model ignores rests, simply adding their durations to the next note's duration, the use of the time dimension is shown to improve the results only in the 2005 training collection. This evidences the need for larger and more heterogeneous test collections for the Symbolic Melodic Similarity task, for researchers to train and tune their systems properly and reduce overfitting to particular collections[9][29].

The results indicate that this line of work is certainly promising. Further research should address the interpolation method to use, different ways of splitting the curve into spans, extend the model to consider rests and polyphonic material, and evaluate on more heterogeneous collections.

References

1. Aloupis, G., Fevens, T., Langerman, S., Matsui, T., Mesa, A., Nuñez, Y., Rappaport, D., Toussaint, G.: Algorithms for Computing Geometric Measures of Melodic Similarity. Computer Music Journal 30(3), 67–76 (2006)
2. Bainbridge, D., Dewsnip, M., Witten, I.H.: Searching Digital Music Libraries. Information Processing and Management 41(1), 41–56 (2005)
3. de Boor, C.: A Practical guide to Splines. Springer, Heidelberg (2001)
4. Bozkaya, T., Ozsoyoglu, M.: Indexing Large Metric Spaces for Similarity Search Queries. ACM Transactions on Database Systems 24(3), 361–404 (1999)
5. Byrd, D., Crawford, T.: Problems of Music Information Retrieval in the Real World. Information Processing and Management 38(2), 249–272 (2002)
6. Casey, M.A., Veltkamp, R.C., Goto, M., Leman, M., Rhodes, C., Slaney, M.: Content-Based Music Information Retrieval: Current Directions and Future Challenges. Proceedings of the IEEE 96(4), 668–695 (2008)

7. Clifford, R., Christodoulakis, M., Crawford, T., Meredith, D., Wiggins, G.: A Fast, Randomised, Maximal Subset Matching Algorithm for Document-Level Music Retrieval. In: International Conference on Music Information Retrieval, pp. 150–155 (2006)
8. Doraisamy, S., Rüger, S.: Robust Polyphonic Music Retrieval with N-grams. Journal of Intelligent Systems 21(1), 53–70 (2003)
9. Downie, J.S.: The Scientific Evaluation of Music Information Retrieval Systems: Foundations and Future. Computer Music Journal 28(2), 12–23 (2004)
10. Downie, J.S., West, K., Ehmann, A.F., Vincent, E.: The 2005 Music Information Retrieval Evaluation Exchange (MIREX 2005): Preliminary Overview. In: International Conference on Music Information Retrieval, pp. 320–323 (2005)
11. Hanna, P., Ferraro, P., Robine, M.: On Optimizing the Editing Algorithms for Evaluating Similarity Between Monophonic Musical Sequences. Journal of New Music Research 36(4), 267–279 (2007)
12. Hanna, P., Robine, M., Ferraro, P., Allali, J.: Improvements of Alignment Algorithms for Polyphonic Music Retrieval. In: International Symposium on Computer Music Modeling and Retrieval, pp. 244–251 (2008)
13. Isaacson, E.U.: Music IR for Music Theory. In: The MIR/MDL Evaluation Project White paper Collection, 2nd edn., pp. 23–26 (2002)
14. Kilian, J., Hoos, H.H.: Voice Separation — A Local Optimisation Approach. In: International Symposium on Music Information Retrieval, pp. 39–46 (2002)
15. Lin, H.-J., Wu, H.-H.: Efficient Geometric Measure of Music Similarity. Information Processing Letters 109(2), 116–120 (2008)
16. McAdams, S., Bregman, A.S.: Hearing Musical Streams. In: Roads, C., Strawn, J. (eds.) Foundations of Computer Music, pp. 658–598. The MIT Press, Cambridge (1985)
17. Mongeau, M., Sankoff, D.: Comparison of Musical Sequences. Computers and the Humanities 24(3), 161–175 (1990)
18. Selfridge-Field, E.: Conceptual and Representational Issues in Melodic Comparison. Computing in Musicology 11, 3–64 (1998)
19. Smith, L.A., McNab, R.J., Witten, I.H.: Sequence-Based Melodic Comparison: A Dynamic Programming Approach. Computing in Musicology 11, 101–117 (1998)
20. Smith, T.F., Waterman, M.S.: Identification of Common Molecular Subsequences. Journal of Molecular Biology 147(1), 195–197 (1981)
21. Typke, R., den Hoed, M., de Nooijer, J., Wiering, F., Veltkamp, R.C.: A Ground Truth for Half a Million Musical Incipits. Journal of Digital Information Management 3(1), 34–39 (2005)
22. Typke, R., Veltkamp, R.C., Wiering, F.: A Measure for Evaluating Retrieval Techniques based on Partially Ordered Ground Truth Lists. In: IEEE International Conference on Multimedia and Expo., pp. 1793–1796 (2006)
23. Typke, R., Veltkamp, R.C., Wiering, F.: Searching Notated Polyphonic Music Using Transportation Distances. In: ACM International Conference on Multimedia, pp. 128–135 (2004)
24. Typke, R., Wiering, F., Veltkamp, R.C.: A Survey of Music Information Retrieval Systems. In: International Conference on Music Information Retrieval, pp. 153–160 (2005)
25. Uitdenbogerd, A., Zobel, J.: Melodic Matching Techniques for Large Music Databases. In: ACM International Conference on Multimedia, pp. 57–66 (1999)
26. Ukkonen, E., Lemström, K., Mäkinen, V.: Geometric Algorithms for Transposition Invariant Content-Based Music Retrieval. In: International Conference on Music Information Retrieval, pp. 193–199 (2003)

27. Urbano, J., Lloréns, J., Morato, J., Sánchez-Cuadrado, S.: MIREX 2010 Symbolic Melodic Similarity: Local Alignment with Geometric Representations. Music Information Retrieval Evaluation eXchange (2010)
28. Urbano, J., Marrero, M., Martín, D., Lloréns, J.: Improving the Generation of Ground Truths based on Partially Ordered Lists. In: International Society for Music Information Retrieval Conference, pp. 285–290 (2010)
29. Urbano, J., Morato, J., Marrero, M., Martín, D.: Crowdsourcing Preference Judgments for Evaluation of Music Similarity Tasks. In: ACM SIGIR Workshop on Crowdsourcing for Search Evaluation, pp. 9–16 (2010)
30. Ó Maidín, D.: A Geometrical Algorithm for Melodic Difference. Computing in Musicology 11, 65–72 (1998)

Appendix: Results with the Original *All-2* Ground Truth Lists

Here we list the results of all 55 combinations of k_n and k_t evaluated with the very original *Train05* (see Table 4) and *Eval05* (see Table 5) test collections, ground truth lists aggregated with the *All-2* function [21][28]. These numbers permit a direct comparison with previous studies that used these ground truth lists as well.

The qualitative results remain the same: k_n=4 seems to perform the best, and the effect of the time dimension is much larger in the *Train05* collection. Remarkably, in *Eval05* k_n=4 outperforms all other n-gram lengths for all but two levels of k_t.

Table 4. Mean ADR scores for each combination of k_n and k_t with the *Train05* test collection, ground truth lists aggregated with the original *All-2* function. k_p is kept to 1. Bold face for largest scores per row and italics for largest scores per column.

k_n	k_t=0	k_t=0.1	k_t=0.2	k_t=0.3	k_t=0.4	k_t=0.5	k_t=0.6	k_t=0.7	k_t=0.8	k_t=0.9	k_t=1
3	0.7743	0.7793	0.788	0.7899	0.7893	0.791	**0.7936**	0.7864	0.7824	0.777	0.7686
4	0.7836	*0.7899*	0.7913	*0.7955*	*0.7946*	*0.8012*	***0.8039***	*0.8007*	*0.791*	*0.7919*	*0.7841*
5	0.7844	0.7867	*0.7937*	**0.7951**	0.7944	0.7872	0.7799	0.7736	0.7692	0.7716	0.7605
6	*0.7885*	0.7842	**0.7891**	0.7851	0.7784	0.7682	0.7658	0.762	0.7572	0.7439	0.7388
7	**0.7598**	0.7573	0.7466	0.7409	0.7186	0.7205	0.7184	0.7168	0.711	0.7075	0.6997

Table 5. Mean ADR scores for each combination of k_n and k_t with the *Eval05* test collection, ground truth lists aggregated with the original *All-2* function. k_p is kept to 1. Bold face for largest scores per row and italics for largest scores per column.

k_n	k_t=0	k_t=0.1	k_t=0.2	k_t=0.3	k_t=0.4	k_t=0.5	k_t=0.6	k_t=0.7	k_t=0.8	k_t=0.9	k_t=1
3	**0.7185**	0.714	0.7147	0.7116	0.712	0.7024	0.7056	0.7067	*0.708*	0.7078	*0.7048*
4	*0.7242*	*0.7268*	*0.7291*	***0.7316***	*0.7279*	*0.7282*	*0.7263*	*0.7215*	0.7002	*0.7108*	0.7032
5	**0.7114**	0.7108	0.6988	0.6958	0.6942	0.6986	0.7109	0.7054	0.6959	0.6886	0.6914
6	**0.708**	0.7025	0.6887	0.6693	0.6701	0.6743	0.6727	0.6652	0.6612	0.6561	0.6636
7	0.6548	**0.6832**	0.6818	0.6735	0.6614	0.6594	0.6604	0.6552	0.6525	0.6484	0.6499

It can also be observed that the results would again be overestimated by as much as 11% in the case of *Train05* and as much as 13% in *Eval05*, in contrast with the maximum 12% observed with the systems that participated in the actual MIREX 2005 evaluation.