Musicae Scientiae
Discussion Forum 4A, 2007, 153-181

Transportation distances and human perception of melodic similarity

RAINER TYPKE*, FRANS WIERING**
AND REMCO C. VELTKAMP**
* The Austrian Research Institute for Artificial Intelligence (OFAI)
** Department of Information and Computing Sciences, Universiteit Utrecht

• **ABSTRACT**

This article describes how transportation distances such as the Earth Mover's Distance can be used for measuring melodic similarity for notated music. We represent music notation as weighted point sets in a two-dimensional space of onset time and pitch. The Earth Mover's Distance can then be used for comparing point sets by determining how much work it would take to convert one of the point sets into the other by moving weight between the point sets.

For evaluating how well this method and other methods agree with human perception of melodic similarity, we established a ground truth for the RISM A/II collection based on the opinions of human experts.

The RISM A/II collection contains about half a million musical incipits. For 22 queries, we filtered the collection so that about 50 candidates per query were left, each of which we then presented to about 30 human experts (out of a group of 37 experts) for a final ranking. We present our filtering methods, the experiment design, the resulting ground truth, and a new measure (called "Average Dynamic Recall") that can be used for comparing different similarity measures with the ground truth.

1. MODELLING MELODIES FOR RETRIEVAL

Music information retrieval (MIR) has only recently become a major research area, even though the concept originates in the 1960s (Byrd and Crawford, 2002). The aim of MIR is to develop methods for finding musical information from a collection of digitised instances of musical works, typically encoded scores or audio recordings. An especially challenging area of MIR is content-based music retrieval, where the user's query specifies the desired musical content of the works he wants to find. Usually, the query is a melody, and the user expects to find works that contain that melody or one that resembles it. An adequate model of melody is thus generally a prerequisite for a successful MIR system. Such a model may or may not be informed

by insights from music perception and cognition research: the decisive criterion is retrieval performance, not biological or cultural plausibility - although the latter may better satisfy the intellect.

Even when endowed with only moderate musical abilities, people seem to be able to recognize hundreds of musical works from their principal melodies. From this follows that melodies are rich in features that allow distinguishing between them. On the other hand, there are also features that seem to have little or no influence on the perception of melodic difference, such as transposition, tempo, ornamentation, and moving, splitting and merging individual tones. One can almost predict from this that retrieval methods based on a single feature are bound to be rather unsuccessful, for example methods that consider melodies as strings of pitches or intervals.

Our research proposes the use of weight flow distances, the Earth Mover's Distance in particular. Originally developed for graphics shape matching, these distances model the effort needed to transform one shape into another. When applied to the musical domain, they can be used to compare — among other things — melodic shapes.

1.1. Contribution

The following three sections present our transportation-based distance measure for notated music, the ground truth we built for the RISM A/II collection, and our proposed measure for comparing the result quality of various MIR methods against the ground truth.

2. Measuring melodic similarity with transportation distances

In this section, we describe the current version of our distance measure for melodies that is based on the Earth Mover's Distance (EMD). An earlier version is described in by Typke *et al.* (2004). The current version optimizes the alignment of point sets in a way that the dependance on the segmenting algorithm is lower.

Our distance measure can be used to find occurrences of melodies that are similar to a given query in a database of pieces of music. This algorithm takes onset times, note durations, and pitches into account at the same time. It also supports partial matching, that is, the notes in the short query are compared to the most similar group of notes somewhere in the piece of music, but the rest of this piece does not influence the comparison result.

2.1. Earth Mover's Distance

At the core of our method lies the EMD (Rubner *et al.*, 1998), which determines the minimum amount of work that is needed for converting one set of weighted points into another. The required work grows with the amount of weight that needs to be

moved to different positions, and with the distance over which the weight needs to be moved.

The EMD is defined as follows:

$$EMD\,(A,\,B) = \frac{\min_{F \in \mathcal{F}} \sum_{i=1}^{m} \sum_{j=1}^{n} f_{ij} d_{ij}}{min\,(W,\,U)}$$

A and B are sets of weighted points. $\mathcal{F}$ is the set of all possible flows that would convert A into B (constraints are: one set acts only as a supplier, the other one only as a receiver, no set supplies or receives more than its total weight, and the lighter of the two sets is completely matched). Every flow consists of one flow element for each pair of points out of the m points in A and the n points in B. Every flow element carries a weight of f_{ij} over a ground distance of d_{ij} from one point in A to one point in B. W and U are the sums of weights in set A and B, respectively. Therefore, the EMD is the sum of distances in the optimum flow, weighted with the corresponding weights, normalized with the total weight of the lighter point set.

2.2. Representing music as set of weighted points

To be able to apply the EMD, we need to represent music as weighted point sets. This is done by representing every note as a point in the two-dimensional space of onset time (X-coordinate) and pitch (Y-coordinate). The duration of the note can be represented with the weight that is assigned to the point. This representation works for both monophonic and polyphonic music. See Figure 1 for an example. It would probably be even better to use the weights for representing the notes' importance for human perception, which could be determined by using Lui's algorithm (Lui *et al.*, 2005). This way, pitch and onset time would also influence the weights. In our basic version of the algorithm, however, pitch and onset time are only reflected by the coordinates of points.

2.3. Applying the EMD to weighted point sets

See Figure 2 for an example of how the EMD can be used to compare two pieces of music. For making the flow better visible, the black point set was shifted a bit to the top so that points that lie on top of one another are visible as separate entities. In reality, the distance between the leftmost points in the two point sets is zero, therefore only points that represent notes which are different from their corresponding notes in the other point set increase the distance measure.

Note that the EMD can handle cases where one note has more than one corresponding note in the compared piece of music (the last note of the melody on the top is matched with two notes in the bottom melody). A shortened note that occurs later than its corresponding note still is partially matched with the right note, but since it is actually different, it contributes to the non-zero overall distance.

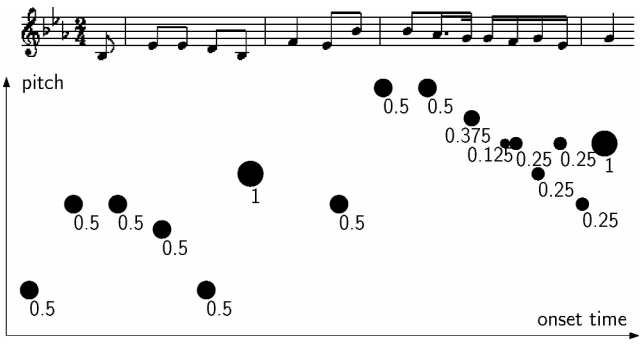


Figure 1.

A piece of music (top) and its representation as weighted point set in the two-dimensional space of onset time and pitch. The durations of the notes can be represented as the weights of the points, which are here shown as the area of the disks around the points and as numbers. The numbers are multiples of quarter notes.

In absolute numbers, the EMD depends on how one converts pitches and time into coordinates. If, for example, an eigth note covers 1/8 on the time axis, and we align the two point sets such that the first note has the time coordinate zero in both point sets, the B in measure 2 (the first note where the two point sets differ) would have a time coordinate of 1 in the top point set and 17/16 in the bottom one. Their pitch coordinate would be the same. The ground distance between these two notes would therefore be 1/16. Since the amount of weight moved from one B to the other is 0.25, this partial flow would contribute 1/16. 0.25 = 1/64 to the overall EMD distance.

Generally, the question what notes should be matched to one another is simply answered by determining which set of flows results in the lowest possible distance. Thus, our algorithm does not need specific rules for all possible constellations.

The EMD depends on what distance measure is used for the ground distance, the measure for distances between individual points. So far, we have simply used the Euclidean distance, but it is conceivable that a distance measure that is inspired by Krumhansl's work (Krumhansl, 1990) would work better. See Section 5 for details.

2.4. Segmenting and aligning point sets correctly before calculating the EMD

For melodic similarity as it is perceived by humans, transposition has almost no influence, and for the character of a melody, it also does not matter very much where in a piece the melody occurs. On the other hand, the coordinates of points have a large influence on the EMD. Therefore, we cannot simply calculate the EMD between two point sets representing two pieces of music and expect the result to have much to do with melodic similarity unless the point sets were properly aligned. In

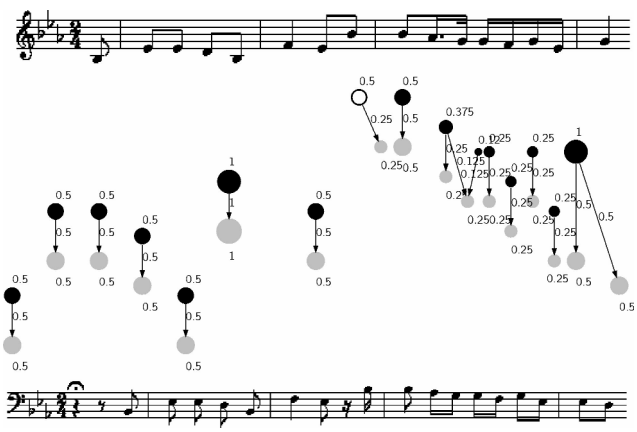


Figure 2

Two pieces of music and their representations as weighted point sets, along with the optimum flow that turns one of the point sets into the other. In this picture, the points representing the top melody are vertically shifted to make the flow more visible. The first seven points, for example, actually have the same coordinates and weights in both sets and therefore contribute nothing to the overall distance.

other words, we have to find a proper translation of the query in both the time and pitch dimension such that the EMD is minimized.

We would also like our method to support different tempi of the query, so the point set representing the query should also be scaled in the time dimension. Besides differences in the global tempo, we would also like to support local tempo fluctuations to make the algorithm suitable for queries that are entered by humans, for example for Query by Humming.

All this can be done by segmenting the query into overlapping, short point sets before translating and scaling them. Our segmenting algorithm counts consecutive notes and ignores additional notes that occur at the same time in different voices. The algorithm works as follows. First, we set a pointer to the onset time of the first note that is to become part of the next segment. This is the beginning time of a new segment. Then, we move the pointer to the next end of any note whose onset time lies within the current new segment, then to the next beginning of a note. We do this *n* times (for *n* consecutive notes in the segment). We include all notes with an onset time within the closed interval from the beginning of the segment to the current pointer position in the next segment.

Figure 3 illustrates the algorithm for overlapping segments of length six that are three notes apart.

Figure 4 illustrates the problem of minimizing the EMD by translating and scaling. The query is shown at the top. One could, for example, take the group of

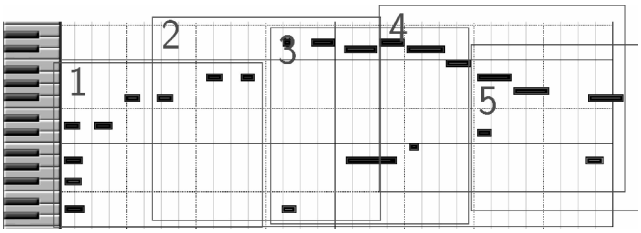


Figure 3

The segmenting algorithm for monophonic or polyphonic music counts consecutive notes while ignoring additional notes that occur at the same time.

notes from the fourth to the ninth note as one segment. The corresponding weighted point set is shown as a set of black dots. Before applying the EMD, we now need to find out that the position shown in Figure 5 would minimize the EMD.

We use the evolutionary optimization function "evofunc" (Min, 2005) of the package "Reusable Evolutionary Algorithms in Shape Matching" (REALISM), part of the Shape Matching Environment (SHAME) for this purpose. Its evolutionary algorithm creates a population of individuals, where every individual stands for a transformation of the source (here, the source is a weighted point set). We allow translation in both dimensions (to become independent of the position of a melody within a piece and also independent of transposition) and scaling in the time dimension (to become independent of tempo) for the transformations, so every individual is characterized by three numbers. The "evofunc" function then uses a series of mutations and selection steps among the individuals for finding a transformation that minimizes the distance of the transformed source from a target (here, the target is the other weighted point set). This optimization method does not guarantee to find the optimum, but it works well in practice. Many other optimization methods could be considered for this task of finding a good alignment. We have chosen the evolutionary optimization algorithm because of its straightforward application and good results.

2.4.1. Aggregating the results obtained for segments

After finding documents that contain groups of notes resembling query segments, we have several lists of possible matches (one list per segment), with the EMD distance between the match and the query segment for every match.

The user, however, would like to get just one list as a response to his query.

When combining the segment results, we need to take into consideration the distances of matches that were found for segments, but possibly also the absence of any matches for some segments. While we do not know the distance to the closest match for segments where we did not find any match (because the closest match is

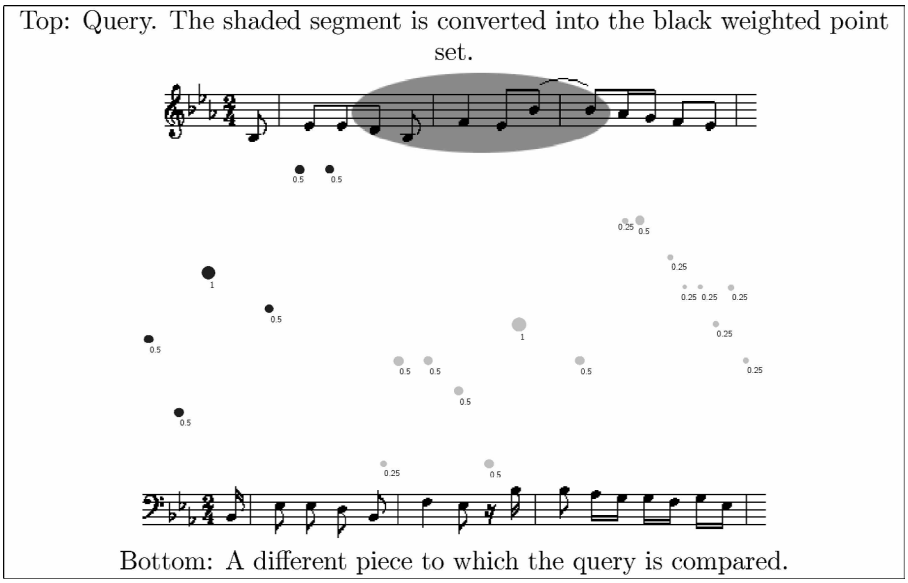


Figure 4

Problem: Before calculating the EMD, we need to somehow find out that the black point set should be moved near the middle of the grey one for minimizing the EMD (see Figure 5).

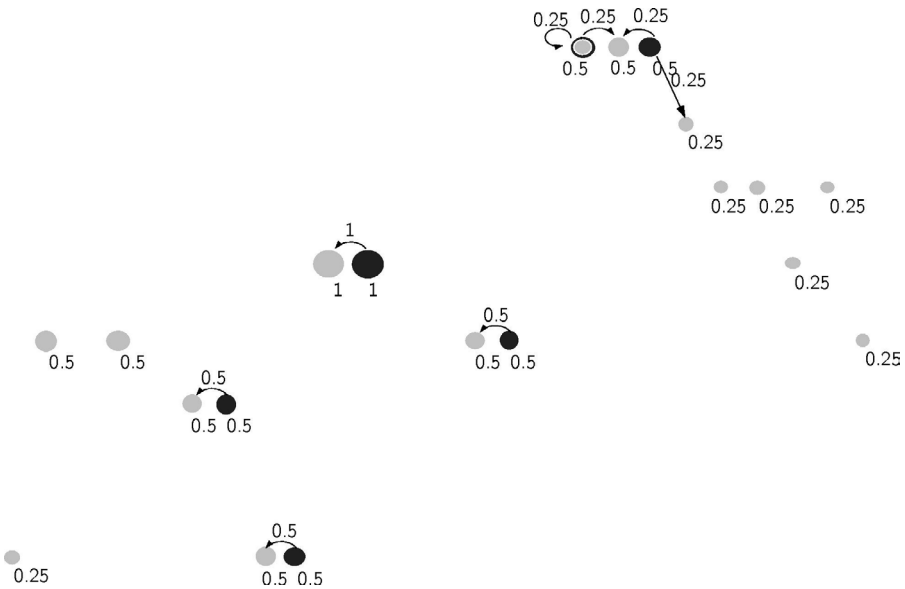


Figure 5

The optimum alignment of the two point sets from Figure 4 so that the EMD is minimized.

outside our search radius), we do know that it is larger than the largest distance that we encountered. Therefore, it seems reasonable to penalize cases where no match was found with that maximum distance.

We calculate one overall score based on the list of segments in the following way:
- For every document that occurs in any list of matches for a query segment ("candidate document"), construct a list that contains: Query segment number, distance to the document, and the onset times of the first and last query note within the matched document. This list might contain the same query number multiple times (if it matches at more than one place within the document), and it might contain different query segment numbers (if more than one query segment matches the document).
- For every candidate document, determine how many of the query segments would ideally match. This would be all query segments if the candidate document is longer than the query (contains more consecutive notes than the query), or, if the document is shorter than the query, as many query segments as would cover the entire document.
- For every candidate document, calculate an overall distance score by computing the sum of distances for a legal combination of segments that match with this document and adding the maximum segment distance for every query segment that should also match (based on the comparison of query length and document length) but fails to do so. We simply use a recursive brute-force algorithm for finding the minimum distance for all legal combinations, which is bearable if there are only very few segments to consider (queries tend to be very short). To make it even more bearable, we cut the number of possibilities by not further checking obviously non-optimal parts of the recursion tree. For example, if we have already determined that segments *a, b,* and *c* can follow one another, we no longer check any possibilities that start with segments *b, c* since that would never lead to a lower score than a combination that starts with *a, b, c.*

By "legal combination", we mean that:
- No segment number occurs twice.
- Segments with a higher number are matched at a later position within the document.
- For segments with consecutive numbers: the overlap of the matched areas corresponds to the overlap of the segments within the query (if there is any), and the matched areas are not too far apart.

3. A GROUND TRUTH FOR THE RISM A/II COLLECTION

For evaluating the performance of a music retrieval system, one needs a ground truth for its data collection and some given queries. In other words, for the given queries, it should be known what the ideal search result is.

The music retrieval systems we have in mind serve the information need for music that is melodically similar to a given query.

The RISM A/II collection (RISM, 2002) contains 476,600 incipits, short excerpts of notated music from the beginnings of manuscripts in libraries, archives, cloisters, schools, and private collections worldwide. This collection is useful for content-based music retrieval because of its size and the fact that it contains real music written by human composers. A music retrieval system that does not work well with this collection probably also does not perform well for real-world applications in general. Our ground truth can serve as a benchmark for deciding how well a music retrieval system works with the RISM A/II collection.

In TREC (Voorhees and Harman, 2000), relevance assessments are mostly binary ("relevant" or "not relevant"). Only in more recent TREC web tracks such as at TREC-9 (Hawking, 2001), this was extended to ternary ("irrelevant"/"relevant"/ "highly relevant").

Studies such as Selfridge-Field (1998) show that melodic similarity is continuous. Local melodic changes such as lengthening a note or moving it up or down a step are usually not perceived as changing the identity of a melody, and by applying more and more changes, the perceived relationship to the original becomes only gradually weaker. Also, melodies are generally quite resistant to the insertion of all sorts of ornamentation.

Because of the continuity of melodic similarity, there are no sensible criteria for assigning one out of a few distinct degrees of relevance to a melody, so any relevance assessment with a given scale length seems inappropriate. Instead, we asked human experts to rank all incipits where they saw any similarity to the query. Our ground truth therefore does not consist of sets of highly relevant, relevant and irrelevant documents, but of ranking lists of documents.

A valid way of establishing such a ground truth would be to ask a number of human experts to look at all possible matches for a given query (carefully making sure that they stay concentrated long enough) and order them by similarity. Since we cannot expect our human experts to sift through half a million melodies, we needed to filter out incipits of which we can be reasonably sure that they do not resemble the query.

3.1. Filtering Melodies

To be able to exclude incipits that are very different from our selected queries, we calculated some features for every incipit in the database. Filtering could then easily be done by issuing SQL statements with selections based on those features.

- **Pitch range:** the interval between the highest and lowest note in the incipit.
- **Duration ratio:** the duration of the shortest note (not rest), divided by the duration of the longest note (not rest). The result is a number in the interval (0,1], where 1 means that all notes have the same duration, while a very small number means a very high contrast in durations.
- **Maximum interval:** the largest interval between subsequent notes. Rests are ignored.

• **Editing distance between gross contours:** the editing distance between two character strings is the sum of the costs of the cheapest possible combination of character insertion, deletion, and replacement operations that transform one string into the other. We determined the gross contour as a string of characters from the alphabet U ("up"), D ("down"), and R ("repeat") and calculated the distance to every query for each incipit in the database, using the editing distance described by Prechelt and Typke (2001). They had optimized the costs for the insertion, deletion, and replacement operations for gross contour strings such that the resulting similarity measure corresponds well with human perception.
• **Editing distance between rhythm strings:** we also represented the incipits as rhythm strings with one character from a three-character alphabet for each pair of subsequent notes: longer, shorter, and same duration.
• **Interval histogram:** the number of occurrences for each interval between subsequent notes, normalized with the total number of intervals. With this feature, we can base selections on things like "incipits with many thirds".
• **Interval strings:** one string of diatonic intervals and one string of chromatic intervals for every incipit. This makes it possible to select incipits that contain a certain sequence of intervals.
• **Motive repetitions:** in order to be able to select things like "all incipits with at least three repeated notes in two different places", we collected sequences of intervals that were repeated at least once, along with their number of occurrences, for every incipit. The repetition detection algorithm maximizes the motive length, even if this means fewer repetitions.

We used different filtering steps and features for every query since every query has its own characteristic features. Every filtering step had the aim of reducing the number of candidates for matches for a given query by excluding incipits with features that make them very different from the query. As long as this holds for every filtering step, different people should arrive at similar candidate lists even if they apply different filtering steps. However, they need to have similar notions of melodic dissimilarity (also similar to those of the human experts whose input determines the actual ground truth).

For example, we used the following filtering steps for the "White Cockade" incipit whose ground truth is shown in Table 2:
• Exclude incipits whose pitch range is less than an octave or greater than a minor tenth. This excluded 78 % of the incipits in the database.
• Exclude incipits whose maximum interval between subsequent notes is less than a minor sixth or greater than a diminished seventh. This excluded 79 % of the remaining incipits.
• Exclude incipits with a duration ratio greater than 0.51, *i.e.* incipits where all notes have quite similar durations. This excluded a further 4 % of incipits.
• Exclude incipits that do not contain at least one of the two interval sequences "fifth up, third down, unison, sixth up" or "third up, unison, unison, sixth up". This left us with 88 incipits.

Because of the dangers of filtering too strictly and thereby accidentally excluding incipits that are similar to the query, we stopped the filtering process once the number of remaining incipits had fallen below 300. To arrive at the desired number of about 50 candidates, we manually excluded remaining incipits that were very different from the query.

As an additional measure to limit the error introduced by accidentally filtering out similar incipits, we used our prototype of a search engine based on transportation distances (see Section 2; we used the algorithm without segmenting) as well as two algorithms by Lemström *et al.* (2003) for finding incipits that are similar to the query. The latter two algorithms, called P2 and P3 by their authors, find incipits containing transpositions of the query where many onset time/pitch combinations match, and incipits containing transpositions of the query with maximum common duration with matching pitch. From these search results, we included candidates that we considered similar although they had been filtered out. Also, we used the metadata in the RISM A/II collection. For example, for "Roslin Castle" (see Table 1), we made sure that every incipit whose title contains the word "Roslin" was included.

With these methods, we found between 0 and about 8 additional candidates for each query, with an average of about 4. In a comparison of algorithms based on the ground truth, one needs to avoid favouring the algorithms that were used for finding additional candidates against other algorithms. If other search algorithms find more sensible matches for a query that were incorrectly excluded in the filtering steps, they need to be included in the ground truth, ideally by rebuilding the ground truth with a panel of human experts that is shown the more complete list of candidates for matches.

Once we had filtered out all incipits that are not similar to the query, we also removed incipits that were either identical to other incipits or to parts of other incipits. Including identical incipits multiple times in the candidate list would have amounted to asking our experts the same question multiple times, and we wanted to put their time to a more productive use. As a result, only 6 versions of "Roslin Castle" occur in our ground truth in Table 1 although we list 16 known occurrences of this melody in our paper about using transportation distances for measuring melodic similarity (Typke et al., 2003), for which we used the same 2002 version of the RISM database.

3.2. Experiment Design

3.2.1. Notated music, MIDI files

Our goal was to establish a ground truth for the incipits that are contained in the RISM A/II collection. These incipits can be exported from the database in the "Plaine & Easie" format (Howard, 1997) and then rendered in common music notation. In order to prevent differences in the rendition of the notated music from having an impact on the ground truth, we used the software that is included with

the RISM A/II database (RISM, 2002) for rendering the music notation bitmaps and took screen shots of the results. Only in cases where the RISM software fails to show the whole incipit because it is too long for fitting on the screen, we rendered the notated music ourselves by converting the Plaine & Easie data into the Lilypond format (Lilypond — see http://lilypond.org — is an open source music typesetter). In addition to the notated music, we also provided MIDI files generated from the Plaine & Easie data as an illustration of the incipits. However, we told the experiment subjects that the definitive source for similarity judgements is the notated music, and that the MIDI files only serve as an illustration.

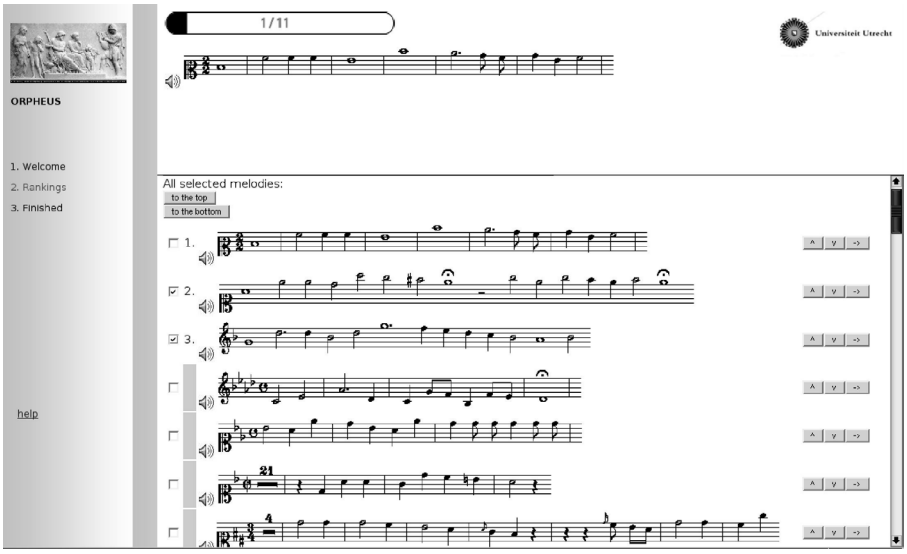


Figure 6

The user interface for the experiment. MIDI files are provided for listening to incipits. In the bottom half of the screen, the subjects can change the order of the candidate incipits, while the query always remains visible at the top.

The metadata from the RISM A/II collection (composer, work title, title of the movement, instrumentation etc.) was not shown to the human experts. They only saw the notated music of the incipits and could listen to a MIDI rendition, as can be seen in Figure 6.

3.2.2. Experts

Müllensiefen and Frieler (2004) point out that music experts tend to have stable similarity judgements, in other words, do not change their mind on what is melodically similar when asked to perform the same judgements a few weeks apart.

Subjects with stable similarity judgements, in turn, seem to have the same notion of melodic similarity (however, there also were some music experts with unstable notions of melodic similarity). In order to establish a meaningful ground truth, we therefore tried to recruit music experts as our experimental subjects. We asked people who either have completed a degree in a music-related field such as musicology or performance, who were still studying music theory, or who attended the International Conference on Music Information Retrieval Graduate School in Barcelona 2004 to participate in our experiment. For organizational and budgetary reasons, we did not test the stability of their notions of melodic similarity. Instead, we try to ignore outliers with statistical methods as described below.

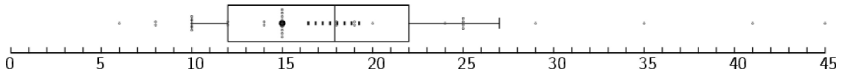


Figure 7

The experience of our experts (instrument playing or studying music), in years. The box extends from the first to the third quartile. The whiskers mark the bottom and top 10 percent. Every data point is shown as a little dot. The median is marked with a fat dot, the mean is shown as a vertical line. The dashed horizontal line around the mean marks one standard deviation below and above the mean.

All of our experts play at least one instrument or sing, most play several instruments. See Figure 7 for a box-and-whisker plot showing their musical experience in years.

3.2.3. Instructions, tasks

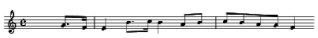
We asked the subjects to rank all candidates that resemble the query by their melodic similarity to the query. Candidates that seemed completely different from the query could be left unranked. The ranking was to be done by reordering the given candidates such that the candidate most similar to the query was at the top, followed by less and less similar candidates, and finally a number of candidates without any assigned ranks that did not resemble the query at all. By asking people to reorder a list instead of picking a rank from a scale, we avoided suggesting how long the ranked list should be, and we also made it easy for the experts to judge whether they ranked all candidates correctly by looking at a local ordering only.

It was sufficient to ensure that for each pair of consecutive candidates in the ranked part of their reordered list, the incipit that was ranked higher was more similar to the query than the other incipit of the pair.

We asked the experts to regard transpositions of a melody as identical, as well as melodies that are notated slightly differently, but in a way that does not affect the way they sound. For example, melodies that are notated with different clefs, but are otherwise the same, should not be viewed as different. In cases where two incipits

Table 1

Ground truth for "Roslin Castle". Table contents: median rank, incipit with title and RISM A/II signature, box-and-whisker plot showing the ranks assigned by our subjects, and a bar composed of squares visualizing the Wilcoxon rank sum test results for every preceding incipit. For details see Section 3.3.2



Query: Anonymus: Roslin Castle, RISM A/II signature: 800.000.193

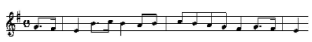
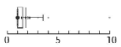
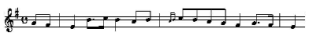
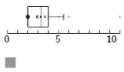
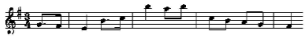
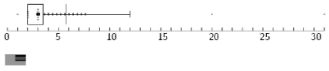
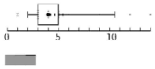
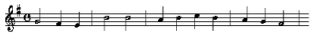
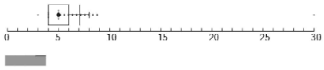
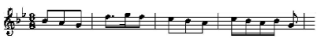
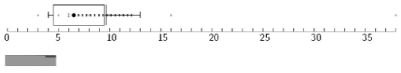
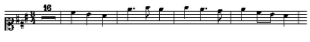
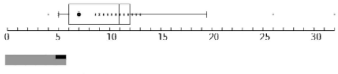
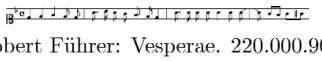
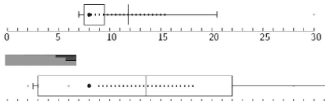
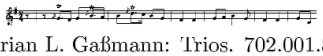
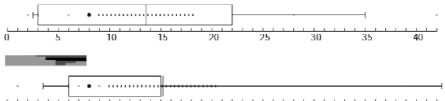
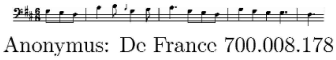
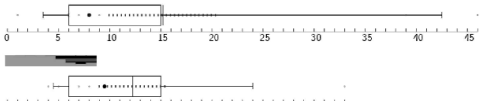
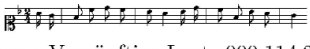
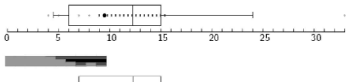
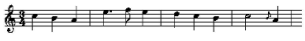
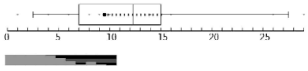
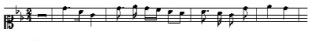
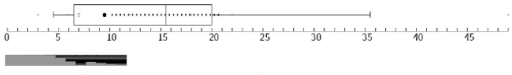
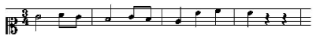
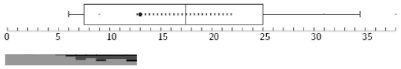
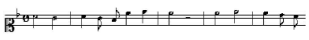
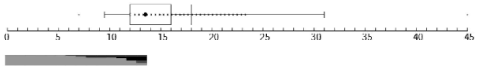
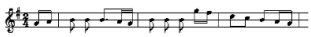
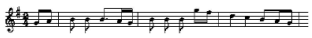
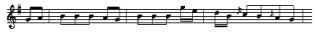
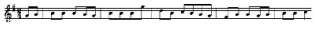
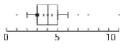
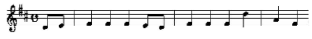
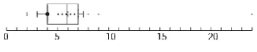
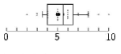
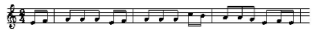
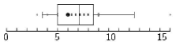
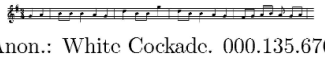
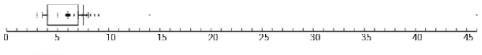
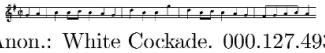
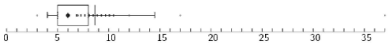
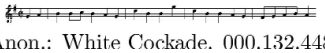
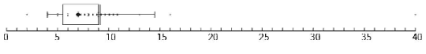
Median Rank	Candidate Incipit, Composer, Title, RISM A/II signature	Ranks, Wilcoxon Test Results (p-values: dark upper area)
1	 Anonymus: Roslin Castle. 000.109.446	
2	 Anonymus: Roslin Castle. 000.111.779	
3	 Anonymus: Roslin Castle. 000.112.692	
4	 Anonymus: Roslin Castle. 000.132.330	
5	 Anonymus: Roslin Castle. 000.112.625	
6.5	 Anonymus: Allegro. 704.000.704	
7	 Anonymus: Care Jesu. 400.196.546	
8	 Robert Führer: Vesperae. 220.000.909	
8	 Florian L. Gaßmann: Trios. 702.001.807	
8	 Anonymus: De France 700.008.178	
9.5	 Anon.: Vernünftige Lust. 000.114.257	
9.5	 Anon.: Pour me venger. 000.109.156	
9.5	 G. Molinari: Mecum enim 850.503.217	
13	 T. W. Fischer: Masses. 550.161.144	
13.5	 G. J. Werner: Puer natus. 530.004.292	

Table 2

**Ground truth for "The White Cockade" by J. F. Latour as query.
Only one out of the top nine pieces, "Cotillons", is not the same piece as the query.
As one should expect, the Wilcoxon rank sum test results warrant a separator
between the first nine incipits and the tenth, which is from a different piece
and at the same time clearly different from the preceding incipits.
For details see Section 3.4**



Query: J. F. Latour: The White Cockade, RISM A/II signature: 000.111.706

Median Rank	Candidate Incipit, Composer, Title, RISM A/II signature	Ranks, Wilcoxon Test Results (p-values: dark upper area)
1	 J. F. Latour: The White Cockade. 000.111.706	
2	 Anon.: White cockade. 000.113.506	
3	 J. F. Latour: The White C. 000.116.073	
4	 E. Hille: Der verurteilte Hochlandsmann. 451.503.814	
5	 J. F. Latour: The White C. 000.113.932	
6	 Anonymus: Cotillons. 190.018.612	
6	 Anon.: White Cockade. 000.135.676	
6	 Anon.: White Cockade. 000.127.493	
7	 Anon.: White Cockade. 000.132.448	

were taken from similar pieces, but covered different amounts of musical material, we asked the subjects to only consider the common parts of the two incipits for the comparison.

We asked every subject for about 2 hours of his time and presented up to 11 queries. We asked the experts to work carefully, even if that meant that they could not finish all 11 queries within two hours. After collecting 30 expert opinions for a query, we stopped showing it to other experts. For the next expert, we picked

11 queries from the set of queries for which we still had fewer than 30 expert opinions. Overall, 37 experts worked on these 11 queries.

3.2.4. Threats to the validity of results

• **Filtering errors.** It is possible that we filtered out some incipits although they are similar to the query. Our ground truth, therefore, could be incomplete. However, this does not threaten the validity of the ranking of those candidates that we did include.

• **Sequence effects.** The initial order of candidates as well as the order in which queries are presented to the experts could have an impact on the results. Experts could be tempted to leave the order similar to the initial order, and they get more tired and at the same time more skilled at using our interface over the course of the experiment. We addressed these problems by randomizing the order of queries for every participant, and we also put the candidates in a new random order whenever a new query appeared on the screen.

• **Carelessness of experts.** For some queries, such as the "White Cockade" shown in Table 2, we included the query itself among the candidates. Careful experts should put it at the very top of the ranked list. Not everybody did, but enough of the experts were careful. This query was recognized as most similar to itself with high statistical significance: the Wilcoxon rank sum test, which we used as described in Section 3.3.1, shows that for every candidate that was not identical to the query, the probability of the null hypothesis is less than or equal to 0.0001123.

3.3. Results

3.3.1. Evaluation methodology

For every query, the subjects were asked to choose and rank as many of the candidates for matches as they thought had some similarity to the query. Those candidates without any similarity could be left unranked. This gives us a list of ranks for every candidate. These lists tend to be longer for the candidates that are more similar to the query.

To obtain a ground truth, we ordered the candidates by their median rank and then by their mean rank. In addition, for every ranked candidate, we applied the Wilcoxon rank sum test to the ranked candidate and every incipit that was ranked higher. The Wilcoxon rank sum test, given two samples, determines the probability of the null hypothesis (p-value), that is, the hypothesis that the median values are the same for the whole two populations from which the samples were taken (here, the population would be the group of all Western music experts).

We used it to find out how likely it is that the differences in ranking observed by us are only caused by our choice of 37 people out of the whole population of music experts. A low p-value resulting from the Wilcoxon test means that the difference in medians is probably not a coincidence. A large p-value does not necessarily mean that the medians are the same, but just that we do not have compelling evidence for them being different.

3.3.2. The resulting ground truth tables

We visualize the ranks assigned to each candidate with a box-and-whisker plot. The box extends from the first to the third quartile. The whiskers mark the bottom and top 10 percent. Every data point is shown as a little dot. The median is marked with a fat dot, the mean is shown as a vertical line. The dashed horizontal line around the mean marks one standard deviation below and above the mean. The numbers on the scales reflect ranks.

Below every box-and-whisker plot except for the first one, we visualize the Wilcoxon rank sum test results with a horizontal bar that is composed of one square for every incipit which is ranked higher than the current one. Each of these squares has a dark upper area and a lower area with a lighter colour. The size of the dark upper area reflects the p-value (see Section 3.3.1 for an explanation of what the p-value means).

For incipits where every square in the Wilcoxon visualization is almost entirely light-coloured, we can be reasonably sure that all preceding incipits should indeed be ranked higher. Wherever this is the case, we draw a horizontal line immediately above the incipit. For Table 1, we set the threshold for the maximum probability for the null hypothesis at 0.25. In other words, we draw a horizontal line above every incipit where the p-value is less than 0.25 for every single incipit that appears higher in the list. Most actual probabilities are much lower than that, as the visualization of the Wilcoxon tests in Table 1 shows.

For "Roslin Castle" (Table 1), we find five clearly distinguishable groups that way. The incipit with median rank 1 is generally considered the most similar incipit to the query. For the incipit with median rank 2, the Wilcoxon test shows that the probability for the null hypothesis is $p = 0.00006722$. Therefore, we consider the difference in median values statistically significant and separate the second incipit from the first with a horizontal line. For the incipit with median rank 3, the difference in medians is statistically significant for the comparison with the first incipit ($p = 0.0002765$), but not for the comparison with the second incipit ($p = 0.6363$). This is reflected in the Wilcoxon visualization bar, which consists of one almost entirely light-coloured square on the left for the comparison of the third incipit with the first one, and one mostly dark square on the right for the comparison of the third incipit with the second one. Since there is no statistically significant difference between the second and third incipit, we group them together and do not separate them with a horizontal line.

The third group consists of the incipit with median rank 4. The highest of its three p-values resulting from the Wilcoxon tests for its three predecessors is 0.07633. The fourth group again consists of one single incipit, while for all other incipits, there are no statistically significant differences in median ranks. Either we did not have enough subjects who ranked these incipits, or people simply do not consider the dissimilarities between the remaining incipits and the query significantly different.

The tables shown in this paper are not complete. We cut them off a bit after the last detected border between clearly distinguishable groups because the ranking becomes less reliable and therefore less interesting towards the bottom of the tables. The complete data are available online at http://give-lab.cs.uu.nl/orpheus.

3.4. Musical properties of the identified groups

In Table 1 ("Roslin Castle"), the candidate with the highest rank looks as if it would begin with the query and therefore should, according to our instructions, be regarded as identical to the query since only the common part should be considered. If one looks more closely, however, one notices that the key signatures are different. The resulting differences in two notes, however, are not big enough for our experts to consider it very different from the query.

The incipits with median ranks 2 and 3 constitute the second group. Both begin differently from the query — the incipit with median rank 2 has slight differences in rhythm at the beginning and two grace notes added in the second measure, while the incipit with median rank 3 has its second measure transposed by an octave. Otherwise their beginnings are the same as the query. Our experts agree that these incipits are both less similar than the incipit with median rank 1, but they disagree on whether the transposition of a measure by an octave or the modified rhythm and added grace notes should be seen as a greater dissimilarity. Because of this, these two incipits are combined into one group.

The experts agree that the incipit with median rank 4 is significantly different from those preceding it. This is justified by a minor difference in rhythm in measure 1 and a major one in measure two — the first note is just a grace note, so there is no group of four descending eighth notes in that measure as in all preceding incipits.

The incipit with median rank 5 is again significantly different. The rhythm is changed in several ways, leading to a very noticeable difference in measure 3. The third note in this measure corresponds to the first note in measure 2 of all preceding incipits. Because here this note is not at the beginning of the measure, it is much less emphasized, which changes the character of the melody.

The last statistically significant border between groups is that between the incipits with median ranks 5 and 6.5. The latter is the first incipit of a different piece, and it also has a different time signature, so we would expect a border between groups here.

Another border could be expected between the second and third incipit with median rank 9.5 because the interval sequence at the beginning changes noticeably here. However, at this point in the ranked list, we do not have enough votes per incipit for finding a statistically significant difference.

4. A RESULT QUALITY MEASURE FOR USE WITH OUR GROUND TRUTH: AVERAGE DYNAMIC RECALL

Our ground truth that is described in Section 3 was used at the "1st Annual Music Information Retrieval Evaluation eXchange" (MIREX) 2005 for comparing various methods for measuring melodic similarity for notated music. In order to compare different algorithms, a measure was necessary that compares every algorithm's performance with the ground truth. The measure that was used for ranking the algorithms is described in this section.

Our ground truth does not give one single correct order of matches for every query. One reason is that limited numbers of experts do not allow statistically significant differences in ranks for every single item. Also, for some alternative ways of altering a melody, human experts simply do not agree on which one changes the melody more, so even increasing the number of experts might not always avoid situations where the ground truth contains only *groups* of matches whose correct order is reliably known, while the correct order of matches within the groups is not known.

Kekäläinen and Järvelin (2002a, 2002b) suggested graded relevance assessment measures based on cumulated gain, which are related to traditional measures such as expected search length (Cooper, 1968), average search length (Losee, 1998), and normalized recall (Rocchio, 1966, Salton and McGill, 1983).

We propose a measure (called "average dynamic recall") that measures, at any point in the result list, the recall among the documents that the user should have seen so far. Unlike Kekäläinen's and Järvelin's measures (Järvelin and Kekäläinen, 2002a), this measure only requires a partially ordered result list as ground truth, but no similarity scores, and it works without a binary relevance scale.

4.1. MOTIVATION

Because of the restrictions of binary scales, and also because the ground truth we used is not based on a finite relevance scale and does not contain relevance scores for the documents, we are proposing a new measure for our comparison. We try to meet the following criteria:

1. To make comparisons easy, the measure should deliver one number, for example in the range from 0 to 1, where 0 denotes a completely useless result and 1 a result that completely agrees with the ground truth.
2. In the ground truth, we know only the correct order of groups of matches, not necessarily of every single match. The measure should be able to use the existing information without requiring the ground truth to be completely ordered.
3. There are no relevance scores known for the documents in the ground truth, which only consists of a partially ordered list. The measure should therefore not depend on relevance scores.
4. The measure should not have any parameters one could use to dramatically alter

the results (such as a freely chosen discount function for the purpose of rewarding returning highly relevant matches early, arbitrarily chosen thresholds, and the like).
5. The measure should reward putting matches in the right order, as far as that order is known. Therefore, differences in the order within groups should not influence the result, but differences in the order across group boundaries should.
6. In a similar fashion, violations of the correct order should be punished if they happen across group boundaries.
7. False positives in the result should lead to a lower measure, even if the order of the true positives is correct.
8. Both true and false positives that occur close to the beginning of the result list should have a higher influence on the measure than those occurring closer to the end of the list.
9. Since the group sizes do not mean much (they are influenced, for example, by the threshold for statistical significance that was chosen when the groups were established (Typke *et al.*, 2005)), they should not have a high influence on the measure.

4.2. Definition

Our measure is the average recall over the first n documents, where n is the number of items in the ground truth, and the recall is calculated over a dynamic set of relevant documents. Because of this, we call it "average dynamic recall". At the beginning of the result list, only the most similar document is counted as relevant (or all documents of which it is not known that they are less similar than the most similar one). The set of relevant documents grows with the position in the result list. Since there are groups of documents in the ground truth where no differences in relevance are known, the dynamic set of relevant documents does not always grow just by one single new relevant document. Rather, at each group boundary it grows by all elements of the next group, and it does not grow between group boundaries. However, at each position in the result list, we still divide the number of found relevant items at that position by the position number, not by the number of all items that would count as relevant.

More formally, consider a result list result list

$$\langle R_1, R_2, \ldots \rangle$$

and a ground truth of g groups of items

$$\langle (G_1^1, G_2^1, \ldots, G_{m_1}^1), (G_1^2, \ldots, G_{m_2}^2), \ldots, (G_1^g, \ldots, G_{m_g}^g) \rangle$$

(with m_i denoting the number of members of group i) where we know that $rank\ (G_j^i) < rank\ (G_l^k)$ if and only if $i < k$, but we do not know whether *rank*

$(G_j^i) < rank\,(G_p^i)$ for any i (unless $j = p$)[1]. We propose to calculate the result quality as follows. Let $n = \sum_{i=1}^{g} m_i$ be the number of matches in the ground truth and c the number of the group that contains the ith item in the ground truth ($\sum_{v=1}^{c} m_v \geq \sum_{v=1}^{c-1} m_v < i$). Then we can define r_i, the recall after the item R_i, as:

$$r_i = \frac{\#\,\{R_w \mid w \leq i \wedge \exists\, j, k: j \leq c \wedge R_w = G_k^j\}}{i}$$

The result quality q is then defined as:

$$q = \frac{1}{n} \sum_{i=1}^{n} r_i$$

As an example, consider ⟨(1, 2), (3, 4, 5)⟩ as ground truth and the result list ⟨2, 3 , 1, 5, 7, 8, 9, 4⟩. That is, while we do not know whether item 1 or item 2 should be at the top of the list, we know that both should be ranked higher than any of the items 3, 4, and 5. In this case, the result quality q is calculated as follows:

Pos	encountered	relevant	#found	recall
1	2	1, 2	1	1
2	2, 3	1, 2	1	0.5
3	2, 3, 1	1, 2, 3, 4, 5	3	1
4	2, 3, 1, 5	1, 2, 3, 4, 5	4	1
5	2, 3, 1, 5, 7	1, 2, 3, 4, 5	4	0.8

The overall result quality here is (1+0.5+1+1+0.8)/5 = 0.86.

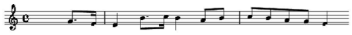
If there would be an additional false positive at position 2, say, ⟨2, 10, 3, 1, 5, 7, 8, 9, 4⟩, the result quality would be lower: 0.7433. False positives lower the result quality in two ways: by shifting subsequent true positives to lower ranks and possibly by shifting true positives out of the scope altogether. Both true and false positives have higher impacts if they occur closer to the beginning of the result list since they influence all subsequent recall values. This illustrates how the criteria number 7 and 8 are met. Criterion 1 is obviously met, and so are criteria 2, 3, and 4.

Criteria 5 and 6 are met because of the way r_i is defined: at every group boundary, the set of items that count as relevant is extended by all elements in the next group. Therefore, it does not matter in which order group members are found, as long as they are found before the group boundary.

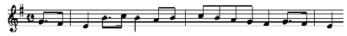
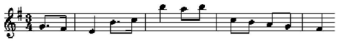
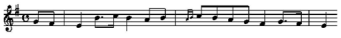
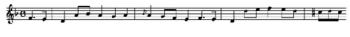
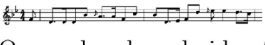
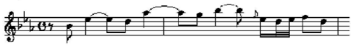
(1) The *rank* function determines the position of an item within the result list. It is 1 for the first element, 2 for the second one and so forth.

Table 3

A sample search result with the ADR calculation. In the "found" column, we list the number of relevant documents found so far. Note that although the second and third match are listed in the inverse order when compared to Table 1, they are still both counted as relevant since they belong to the same group



Query: Anonymus: Roslin Castle, RISM A/II signature: 800.000.193

Distance	Found Incipit, Composer, Title, RISM A/II signature	Found	Recall	ADR
0.0010	 Anonymus: Roslin Castle. 000.109.446	1	1	1
0.0012	 Anonymus: Roslin Castle. 000.112.692	2	1	1
0.0016	 Anonymus: Roslin Castle. 000.111.779	3	1	1
0.002	 Anonymus: Roslin Castle. 000.132.330	4	1	1
0.0102	 Anonymous: Quemadmodum desiderat cervus. 702.004.201	4	0.8	0.96
0.0107	 Sarti, Giuseppe: Gli amanti consolati 240.003.908-1.36.1	4	0.6667	0.9111

A more complex example can be found in Table 3, which shows the ADR for a sample result of our EMD-based algorithm.

4.3. Comparison with normalized discounted cumulative gain

The average dynamic recall (ADR) shares many advantages with the cumulative gain measures introduced by Järvelin and Kekäläinen (2002a), who state that their measures are, among other things, obvious to interpret, are based on recall bases instead of only on retrieved lists, systematically combine document rank and degree of relevance, and, in their normalized forms, support the analysis of performance differences.

- ADR is obvious to interpret: at any number of retrieved items, it gives the average recall among the documents that the user should have seen so far. It can be calculated not only for the first n documents, if n is the number of items in the ground truth, but also for other numbers of documents.
- ADR is based on an absolute ground truth, not on retrieved lists alone, and therefore does not vary uncontrollably if the considered retrieved lists change.

• ADR systematically combines actual document rank and desired document rank.
• ADR supports the analysis of performance differences of different IR methods since it is normalized.

An important difference between ADR and cumulated gain-based measures is that ADR does not rely on relevance scores and therefore does not take them into consideration. This avoids the problem of correctly choosing a discount function for a discounted cumulative gain measure. By choosing the discount function for the normalized discounted cumulative gain (nDCG) (Järvelin and Kekäläinen, 2002a) accordingly, one can sometimes invert the result of performance analyses. Different discount functions put, for instance, different emphasis on the beginnings of result lists. Because of this, it is possible to construct pairs of result lists that differ at the beginning in a way such that with, for example, $\log_2$ as discount function, the first list gets a better nDCG score than the second one. With $\log_3$ as the discount function and the same pair of lists, the nDCG score of the second list can be better than that of the first list.

Besides the discount function, the relative differences between relevance scores also have a high impact on nDCG results. Changing the relevance scores can also lead to opposite comparison results. So, to make nDCG results meaningful, one has to know exactly how the value of a relevant item decreases with a growing position in the result list — this determines the discount function —, and also exactly how relevant every document is in relation to the other documents. The ADR, on the other hand, only requires a partially ordered list as a ground truth for delivering meaningful results.

A weakness of the ADR is that situations can arise where different documents are both counted as relevant or both as irrelevant, with no distinction between them, although it is known which one of the two should be ranked higher.

As an illustration of this problem, consider a ground truth of < (1), (2), (3), (4) > and the result lists < 4, 3, 5, 6 > and < 3, 4, 5, 6 >. It would be nice if the second result list would get a better score because it is known that item 3 should be ranked higher than item 4. But the ADR does not distinguish between them since at the second position, both item 3 and item 4 are not yet in the dynamic set of relevant documents, and at the third position, it is too late to treat them differently because both item 3 and item 4 are already in the set of encountered documents. In a similar way, one can construct examples where pairs of relevant items from different groups in the ground truth are encountered so late in a result list that both are counted as relevant, no matter in which order they appear, although it is known which one of the two should be ranked higher.

Problems like this can be caused in two ways during the calculation of the ADR: by items which are first counted as irrelevant and later as relevant (like item 3 in the example above), or by items which are encountered at a higher position than the end of the group to which they belong in the ground truth. Therefore, one could break

ties like this by calculating an ADR score based on a list containing those problematic items and an inverted ground truth. In this constructed list, all other items are replaced with one item from the most highly ranked group.

In the example above, items 3 and 4 fulfill the condition for inclusion in the constructed list, while items 5 and 6 do not, so we would construct the lists < 4, 3, 1, 1 > and < 3, 4, 1, 1 >. If we now calculate the ADR on these constructed lists using the inverted ground truth (here: < (4), (3), (2), (1) >), the problem with items being treated the same although it is known that they should be ranked differently cannot occur anymore (because of the way the list was constructed). The ADR calculated from these constructed lists and the inverted ground truth could be used to break ties. However, to have a measure that is obvious to interpret, we simply used the ADR as described in Section 4.2 for our comparison of melodic similarity algorithms.

5. Conclusions and future work

We have shown how transportation distances such as the Earth Mover's distance can be used for measuring melodic similarity. Our proposed measure is continuous, makes partial matching possible, supports tempo and pitch fluctuations, and works for any combination of polyphonic and monophonic notated music. The weighting scheme and the ground distance can be chosen independently to give the measure desirable properties.

Weighting scheme and ground distance are two areas where we still see large potential gains. The weighting scheme should ideally be based on the importance of notes for human perception. After all, the weight of a note directly determines its influence on the overall distance measure. One could use an algorithm by Lui, Horner, and Ayers (2005) to attach better weights to notes. Their algorithm was originally intended for reducing the number of voices in polyphonic MIDI files without changing the perceived music too much. It ranks phrases by importance; it should be possible to translate this importance more or less directly into weights.

The ground distance should probably not be simply the Euclidean distance, but reflect perceived distances between pitches. Earlier work by Krumhansl (1990) should help us improve the ground distance. One could, after determining the key in the area around two notes for which the ground distance needs to be measured, determine the dissimilarity between the pitches as perceived by humans in the given tonal context. This dissimilarity could be used instead of the difference of pitches as the vertical component of the ground distance. For the time dimension, the ground distance could still work like the Euclidean distance. In other words, one could use $\sqrt{K_{key}\,(p_1, p_2)^2 + (o_1 - o_2)^2}$ instead of $\sqrt{(p_1 - p_2)^2 + (o_1 - o_2)^2}$, where p_i are pitches, o_i onset times, and K_{key} is the Krumhansl-inspired dissimilarity measure between two given pitches in a given tonal context. This could work in a way very similar to what Shmulevich *et al.* describe (2001), but with certain differences that are made

necessary by the fact that the ground distance is applied to notes from different pieces, not from the same piece.

MIREX 2005 has shown that our ground truth for incipits from the RISM A/II collection in combination with our proposed "Average Dynamic Recall" measure can serve as a basis for a benchmark for evaluating music information retrieval systems. The ground truth, along with the sets of queries, candidates, and experimental results, can be found at http://give-lab.cs.uu.nl/orpheus. We encourage music retrieval researchers to apply their favourite methods to the RISM A/II collection and compare their results to our ground truth.

Address for correspondence:
Rainer Typke
The Austrian Research Institute for Artificial Intelligence (OFAI)
of the Austrian Society for Cybernetic Kudies (OSGK)
Freyung 6/6
A-1010 Vienna (Austria)

Frans Wiering
Remco C. Veltkamp
P.O. Box 80.089
3508 TB Utrecht, the Netherlands

• REFERENCES

Byrd, D. & Crawford, T. (2002). Problems of music information retrieval in the real world. *Information Processing and Management*, *38*, 249-72.

Cooper, W. S. (1968). Expected search length: A single measure of retrieval effectiveness based on weak ordering action of retrieval systems. *Journal of the American Society for Information Science and Technology*, *19 (1)*, 13-41.

Hawking, D. (2001). Overview of the TREC-9 Web Track. In E. M. Voorhees & D.K. Harman (eds), *Proceedings of the 9th Text Retrieval Conference TREC-9*, Gaithersburg, MD.

Howard, J. (1997). Plaine and easie code: A code for music bibliography. In E. Selfridge-Field (ed), *Beyond MIDI: The Handbook of Musical Codes.* Cambridge, MA: MIT Press.

Järvelin, K. & Kekäläinen, J. (2002a). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems*, *20 (4)*, 422-46.

Kekäläinen, J. & Järvelin, K. (2002b). Using graded relevance assessments in IR evaluation. *Journal of the American Society for Information Science and Technology*, *53 (13)*, 1120-9.

Krumhansl, C. L. (1990). *Cognitive Foundations of Musical Pitch*, Oxford: Oxford University Press.

Lemström, K., Mäkinen, V., Pienimäki, A., Turkia, M., & Ukkonen, E. (2003). The C-BRAHMS project. In *Proceedings of the Fourth International Conference on Music Information Retrieval* (pp. 237-8).

Losee, R. M. (1998). *Text retrieval and filtering: analytic models of performance.* Boston: Kluwer Academic.

Lui, S. H., Horner, A., & Ayers, L. (2005). MIDI to SP-MIDI transcoding using phrase stealing. In A. Lewin-Richter & X. Serra (eds), *Proceedings of the 2005 International Computer Music Conference*, San Francisco: International Computer Music Association.

Min, P. (n.d.). Evofunc: evolutionary optimisation function of the library "Reusable Evolutionary Algorithms in Shape Matching" (REALISM). Retrieved from http://www.cs.uu.nl/centers/give/multimedia/matching/shame.html.

Müllensiefen, D. & Frieler, K. (2004). Measuring melodic similarity: Human vs. algorithmic judgments. In R. Parncutt, A. Kessler & F. Zimmer (Eds.), *Proceedings of the Conference on Interdisciplinary Musicology CIM04 on CD-ROM*. Graz, Austria.

Prechelt, L. & Typke, R. (2001). An interface for melody input. *ACM Transactions on Computer-Human Interaction, 8 (2)*, 133–49.

RISM (2002). *Répertoire International des Sources Musicales (RISM). Série A/II, manuscrits musicaux après 1600.* München, Germany: K. G. Saur Verlag.

Rocchio, J. J. (1966). *Document retrieval systems — Optimization and evaluation.* PhD dissertation. Harvard.

Rubner, Y., Tomasi, C., & Guibas, L. J. (1998). A metric for distributions with applications to image databases. In *Proceedings of the IEEE International Conference on Computer Vision* (pp. 59-66), Washington, DC: IEEE Computer Society.

Salton, G. & McGill, M. J. (1983). *Introduction to Modern Information Retrieval.* New York: McGraw-Hill.

Selfridge-Field, E. (1998). Conceptual and representational issues in melodic comparison. *Computing in Musicology, 11*, 3-64.

Shmulevich, I., Yli-Harja, O., Coyle, E., Povel, D., & Lemström, K. (2001). Perceptual issues in

music pattern recognition: Complexity of rhythm and key finding. *Computers and the Humanities*, *35*, 23-35.

Typke, R., den Hoed, M., de Nooijer, J., Wiering, F., & Veltkamp., R. C. (2005). A ground truth for half a million musical incipits. *Journal of Digital Information Management*, *3 (1)*, 34–9. Ground truth data available at http://give-lab.cs.uu.nl/orpheus/.

Typke, R., Giannopoulos, P., Veltkamp, R. C., Wiering, F., & van Oostrum, R. (2003). Using transportation distances for measuring melodic similarity. In *Proceedings of the Fourth International Conference on Music Information Retrieval* (pp. 107-14).

Typke, R., Veltkamp, R. C., & Wiering, F. (2004). Searching notated polyphonic music using transportation distances. In L. Dongge (ed), *Proceedings of the ACM Multimedia Conference* (pp. 128-35), Alpha, NJ: Sheridan Printing.

Voorhees, E. M. & Harman, D. K. (2000). Overview of the eighth Text REtrieval Conference (TREC-8). In E.M. Voorhees & D.K. Harman (eds), *Proceedings of the Eighth Text REtrieval Conference (TREC-8)*, pp. 1-24.

- **Distancias de transporte y percepción humana de similitud melódica**

Este artículo describe cómo distancias de transporte tales como "Earth Mover's Distance" (EMD) pueden ser empleadas para medir similitud melódica en la música escrita. Representamos la notación musical como conjuntos de notas con diverso peso en un espacio bidimensional sobre la base del tiempo y la altura. La EMD puede ser empleada para comparar conjuntos de puntos determinando cuánto llevaría transformar un conjunto de puntos en el otro desplazando el peso entre ambos conjuntos.
Para evaluar la concordancia de éstos y otros métodos con la percepción humana de similitud melódica, establecemos un marco de validación para la colección A/II del RISM, basada en las opiniones de diversos expertos.
La colección A/II del RISM contiene en torno a medio millón de *incipits* musicales. Para veintidós cuestiones, filtramos la colección, de forma que dejamos en torno a cincuenta posibles respuestas para cada pregunta, cada una de las cuales fue presentada a treinta expertos — seleccionados de un grupo de treinta y siete posibles — para llevar a cabo una clasificación final. Presentamos nuestros métodos de filtrado, el modelo de la experiencia, el marco de validación resultante y una nueva medida — denominada "media dinámica de memoria" — que puede ser empleada para comparar diferentes medidas de similitud con el marco de validación.

- **Distanze di trasporto e percezione umana della similarità melodica**

Il presente articolo descrive il modo in cui le distanze di trasporto come la Earth Mover's Distance (EMD) si possano utilizzare per misurare la similarità melodica nella musica in notazione. Rappresentiamo la notazione musicale come insiemi di punti dotati di peso in uno spazio bidimensionale formato da coordinate di tempo e altezza. La EMD si può applicare per confrontare insiemi di punti, determinando la quantità di lavoro necessaria per convertire un insieme di punti nell'altro spostando il peso fra i due insiemi.
Per valutare il grado di concordanza di questo e altri metodi con la percezione umana della similarità melodica, abbiamo fissato un *ground truth* per la raccolta RISM A/II, basato sulle opinioni di soggetti umani esperti di musica.
La raccolta RISM A/II contiene circa mezzo milione di incipit musicali. Per 22 domande, abbiamo filtrato la raccolta in modo che rimanessero circa 50 candidati per ciascuna domanda, ed abbiamo quindi presentato ciascuno di essi a circa 30 umani esperti (su un gruppo di 37) per una valutazione finale. Presentiamo i nostri metodi di filtro, il progetto dell'esperimento, il *ground truth* risultante, ed una nuova misura (chiamata "Average Dynamic Recall") che si può utilizzare per confrontare diverse misure di similarità con il *ground truth*.

• Distances de transport et perception humaine de similarités mélodiques

Dans cet article, nous expliquons la façon dont les distances de transport comme la *Earth Mover's Distance* peuvent être utilisées pour mesurer la similarité mélodique dans le cas de musique notée. Nous représentons la notation musicale comme ensemble de points pondérés dans un espace à deux dimensions : le moment de l'attaque et la hauteur. La *Earth Mover's Distance* peut alors être utilisée pour comparer des ensemble de points en mesurant le travail nécessaire pour convertir un des ensembles en l'autre en déplaçant la pondération entre les ensembles de points. Afin d'évaluer l'accord de cette méthode et d'autres méthodes avec la perception humaine de la similarité mélodique, nous avons établi une base de vérité, tirée de la collection RISM A/II fondée sur l'avis d'experts humains.
La collection RISM A/II comprend plus ou moins un demi-million d'incipit musicaux. Nous avons filtré la collection pour inclure 22 questions, ce qui nous a laissé avec environ 50 candidats par question ; nous avons alors soumis chacun à environ 30 experts humains (pris dans un groupe de 37 experts), pour établir un classement définitif. Nous présentons ici nos méthodes de filtrage, la structure de l'expérience, la vérité de base qui en résulte et une nouvelle mesure, l'*Average Dynamic Recall*, qui peut être utilisée pour comparer différentes mesures de similarité avec la vérité de base.

• Transportdistanzen und menschliche Wahrnehmung melodischer Ähnlichkeit

In diesem Artikel wird beschrieben, wie Transportdistanzen wie beispielsweise die „Earth Mover's Distance" bei der Berechnung melodischer Ähnlichkeit in der Musiknotation verwendet werden können. Wir stellen notierte Musik mit gewichteten Punktmengen im zweidimensionalen Raum dar, der durch Zeit und Tonhöhe aufgespannt wird. Die „Earth Mover's Distance" vergleicht zwei gewichtete Punktmengen durch die Berechnung des Aufwands, der nötig wäre, um eine der beiden Punktmengen durch ein optimales Verschieben von Masse in die andere Punktmenge zu überführen. Dabei steigen die zu minimierenden Kosten mit dem zu transportierenden Gewicht und mit der Entfernung, über die Masse transportiert werden muss. Um festzustellen, wie gut diese Ähnlichkeitsberechnung mit der menschlichen Wahrnehmung melodischer Ähnlichkeit übereinstimmt, haben wir für die RISM A/II-Sammlung aufgrund der Meinungen menschlicher Experten für einige Suchanfragen optimale Ergebnisse berechnet. Die RISM A/II-Sammlung enthält etwa eine halbe Million Musikincipits. Für 22 Suchanfragen haben wir etwa 50 potentielle Beispiele aus der Sammlung herausgefiltert, die wir dann etwa 30 Musikexperten vorgelegt haben, um eine Rangreihenfolge zu bestimmen. In diesem Artikel beschreiben wir unsere Filtermethoden, das experimentelle Design, die resultierenden Ranglisten und ein neues Qualitätsmaß für Suchergebnisse („Average Dynamic Recall"), das dazu verwendet werden kann, verschiedene Ähnlichkeitsmaße mit den Ergebnissen menschlicher Wahrnehmung zu vergleichen.