

USING PITCH STABILITY AMONG A GROUP OF ALIGNED QUERY MELODIES TO RETRIEVE UNIDENTIFIED VARIANT MELODIES

Jörg Garbers, Peter van Kranenburg, Anja Volk, Frans Wiering, Remco C. Veltkamp, Louis P. Grijp*

Department of Information and Computing Sciences

Utrecht University and *Meertens Institute, Amsterdam

garbers@cs.uu.nl

ABSTRACT

Melody identification is an important task in folk song variation research. In this paper we develop methods and tools that support researchers in finding melodies in a database that belong to the same variant group as a set of given melodies. The basic approach is to derive from the pitches of the known variants per onset a weighted pitch distribution, which quantifies *pitch stability*. We allow for partial matching and AND and OR queries.

Technically we do so by defining a distance measure between weighted pitch distribution sequences. It is based on two applications of the *Earth Mover's Distance*, which is a distribution distance. We set up a distance framework and discuss musically meaningful parameterizations for two tasks: a) Study the inner-group distances between the group as a whole and single members of the group. b) Use the group's weighted pitch distribution sequence to query for variant melodies.

The first experimental results seem very promising: a) The inner-group distances correlate to expert assigned subgroups. b) For variant retrieval our method works better than last year's MIREX winner.

1 INTRODUCTION

In folk song variation research, collection items are associated with each other and grouped by different methods. These include text analysis, meta-information analysis and score-analysis. In the WITCHCRAFT project (*What Is Topical in Cultural Heritage: Content-based Retrieval Among Folksong Tunes*) we try to aid this process with computational methods based on the musical content.

Our initial approach was to use a *melody query in – ranked melodies out* (short: M2M, Melody to Melody) search engine, based on a transportation distance (EMD) in the real valued onset-pitch domain. The actual M2M system, described in [7], proved to be the most effective retrieval system in the 2006 MIREX competition. We hoped that the result list would inspire folk song researchers by presenting items of the collection in new, similarity driven orderings. However, the users were rather disturbed by the amount of false positives (items

on top of the ranking list that should not be there), so we tried to find more informed ways to query the database.

An important task in folksong research is the grouping of related melodies, generally because they might derive from a common melodic ancestor. To do so, one employs features that are *stable* across sets of melodies. However, already in 1951, Bronson [1] pointed out, that finding and describing these stable aspects can be musicologically quite challenging. He proposes to use punch cards and a sorting machine to quantify stability and to describe the relationships between variants.

In a similar approach [2] we have developed computer aided methods and tools to study pitch and harmonic stability. In the current paper we follow up on this approach by making use of the by-products of such kind of studies, namely sets of (manually) aligned melodies: we formally develop the idea to query a database for unclassified variants with a group of known melodic variants (short: G2M, Group to Melody).

In this paper we assume that a set of melodic variants or variant phrases is aligned with respect to the temporal axis and that musically reasonable transformations of the material have been applied. The latter includes translations to a common key, selection of suitable bars and reduction to a particular metrical level, as done in a semi-automatic way in [2]. For a given onset time we thus get the pitches and rests that occur in any of the melodies. By querying with sequences of pitch sets, we abstract from the individual variations that resulted in the different melodies but still maintain stable pitch requirements.

In section 2 we first set up a mathematical and computational framework and define the mapping of musical notes to mathematical values. In section 3 we discuss musically reasonable parameterizations of the framework and in section 4 we conduct and evaluate computational experiments with selected parameterizations and compare the results with previous approaches.

2 THE COMPUTATIONAL FRAMEWORK

This research builds on the *Earth Mover's Distance* (EMD), whose musical application was studied in [7]. It is also an important building block in Typke's M2M search engine, which we used in our first approach to folk song variation retrieval. However, instead of combining the on-

set and pitch dimensions into one Euclidean ground distance, as done in Typke’s contour matching, we compute the EMD on the onset and pitch dimensions separately.

In section 2.1 we introduce the reader into the parameters of the EMD. In the following sections we define musically and task specific values.

2.1 The Earth Mover’s Distance

The EMD is a geometrically motivated transportation distance, which defines a distance between node distributions (weighted node sets) where a *ground distance* is given. To apply it we must identify two sets of nodes and node weights, which for later usage we call *query* set $Q = \{(q_j, w_{q_j})\}$ and *candidate* set $P = \{(p_i, w_{p_i})\}$: Between the query nodes q_j and the candidate nodes p_i we draw edges for which we have to define a reasonable ground distance d_{ij} . (See section 2.2.)

The EMD then computes an edge-flow $F = [f_{ij}]$ with f_{ij} being the weight flow between between p_i and q_j that minimizes the overall cost

$$WORK(P, Q) = \sum_{i=1}^m \sum_{j=1}^n d_{ij} f_{ij}, \quad (1)$$

with the constraint that either all weight flows from P to Q or vice versa. To get a distance, this work is divided by the total weight flow:

$$EMD(P, Q) = \frac{WORK(P, Q)}{\sum_{i=1}^m \sum_{j=1}^n f_{ij}}. \quad (2)$$

When the two distributions have unequal weight, the EMD might nonetheless be zero. Therefore, the EMD has the property to allow for *partial matches*. For mathematical and computational details see [6].

2.2 The EMD for Pitch Distribution Sequences

For musical distance modelling we make use of the EMD twice and later reason about the parameters. But first we define the term *pitch distance* in a way that includes also distances between notes and rests.

Let P be the combined set of all possible pitches given in some representation and rests. To simplify the formulation in the following we call any element of P a pitch, even if it is a rest. We assume that there is a ground distance d_p between all the elements in P , e.g. the MIDI-pitch difference and a penalty distance d_r between rests and non-rests. Let furthermore be d_t a distance in musical time, e.g. in quarter notes or seconds.

A *pitch distribution* is a set $WP = \{(\tilde{w}_i, p_i)\}$ of weighted pitches (same onset time). We define the distance d_{wp} between two pitch distributions as the EMD between the weighted pitches with the ground distance d_p .

A *weighted pitch distribution sequence* is a sequence $WPDS = \{(\hat{w}_j, WP_j, t_j)\}$ of externally weighted (\hat{w}_j) pitch distributions (WP_j) given per onset (t_j). We define the distance d_{wpds} between two $WPDS$ as their

EMD. We therefore combine the pitch distribution distance d_{wp} and the time distance d_t into a ground distance $d_{twp} = d_{wp} \oplus d_t$. Here \oplus can be freely defined: we may scale the single distances and compute the Euclidean ($\sqrt{(\alpha d_{wp})^2 + (\beta d_t)^2}$) or manhattan distance ($\alpha d_{wp} + \beta d_t$). (See section 4.3 for α and β choices.)

2.3 From Weighted Melodies to Pitch Distribution Sequences

In this section we describe how we can turn a set of melodies $M = \{(M_k)\}$ into a weighted pitch distribution sequence. We additionally assume that the melodies are given weights $W = \{(W_k)\}$, which affect the absolute and relative weights of the pitches in the pitch distributions. We will use W to formulate different types of queries (AND vs. OR, see section 3.1) and to manually refine a query by putting different emphasis on more stable core variants than on peripheral variants of a query group. (We do not make use of different W_k in this paper.)

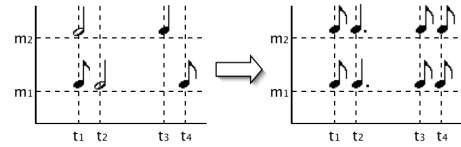


Figure 1. Creating common onsets for a set of melodies: Long notes in melodies m_1 and m_2 are split wherever an onset occurs.

To get to collective pitch distributions, we compute the set of all time positions at which a note-on or note-off event occurs. (See figure 1.) For the time span Δt_j between adjacent time positions we compute a pitch distribution containing the pitches of the melodies at that position.¹ We construct the pitch distribution weights \tilde{w}_i by summing the weights W_k that each melody M_k contributes to a particular pitch i . The external weight \hat{w}_j equals the time span Δt_j . (See section 3.1.)

A special case is the construction of a weighted pitch distribution sequence from a single melody. Here each note turns into a pitch distribution that contains just the note’s pitch with weight $\tilde{w} = 1$ and the external weight of the pitch distribution \hat{w}_j equals the duration of that note.

2.4 Inner Group Distances

As an application of a concrete distance measure between a group of melodies and a single melody, we can apply this measure to single members of the group and the group as a whole. If the distances of the melodies to the group are small, then we have a quite narrow query. If on the other

¹ Note that by doing this for each position where an event occurs in any of the melodies, we get shifts in the weight configuration depending on the set of all events. The M2M search engine does not do this but allows to split long notes during preprocessing, which also leads to slightly different results. We omit to investigate these issues in section 4.5 because we generally project all query notes to the eighth notes metrical level.

hand the distances are quite big, then we either want to look for rather distant variations, or the group is ill defined in terms of the distance measure. (See section 4.4 for an example.)

To check if a group member belongs to the core of a group or to its outskirts, we can either directly compute the distance between the member and the whole group or remove the member from the group before computing the distance. If we want to get the melody that is most characteristic for the group, we can take the melody with the minimum distance to the whole group. We can also repeatedly remove the member that lies at most at the outskirts until we are left with only melodies that have the same distance to the group.

In any case it would be useful to present to the user of a search engine the resulting group-internal ranking list, enabling him to understand the working of the distance measure and allowing him to modify the query melody group, e.g. by removing all group melodies above a distance threshold.

3 INSTANCIATIONS OF THE COMPUTATIONAL FRAMEWORK

In this section we discuss free parameters within the computational framework in ways that make musical sense and help to solve particular tasks.

3.1 Weights

In our application we have the special situation that we match pitch distributions that stem from the query melodies with single pitches of a variant candidate melody from the database. Without loss of generality we turn a pitch into a one element pitch distribution with weight $\tilde{w} = 1$, as described in section 2.3. Given this normalization, how should the query melodies contribute weight to the pitch distribution?

If we set the pitch weights $\tilde{w}_i \geq 1$, we make sure in the first application of the EMD that the candidate pitch is matched with the closest pitch. This is like an *OR*-query. If we let the \tilde{w}_i sum up to 1, we match all pitches at the same time, thus an *AND*-query. In addition we might weight the different query melodies by their *importance* for the melody group. (See W_k in section 2.3.)

The WPDS weights \hat{w}_j are more restricted: Since we typically want to match the same amount of musical time (or small multiples thereof), \hat{w}_j should be proportional to the inter-onset-intervals of the WPDS events. This way a half note in the query can match two quarter notes in the candidate melody.

3.2 Pitch Distances

At the core of the algorithm are pitch distance calculations.

For the *MIDI pitch to MIDI pitch distance* we want to discuss the following setups: 1) d_{p_1} is the absolute difference of the pitch values. 2) d_{p_2} is 0 if the pitches are

the same and 1 otherwise. 3) d_{p_3} is the octave invariant MIDI pitch distance (which has 6 semitones as its maximum value). Expressing d_p as the distance within the musical scale would be another option, which we do not discuss here.

We expect that option d_{p_1} coupled with an AND-query returns lower distances for candidates that have the same contour as the average melody. d_{p_2} counts the pitches that mismatch and thus realizes a boolean notion of *absolute pitch stability*. d_{p_3} ignores octave variations that result from a too small voice range of the singer whose performance was recorded in the transcription. However it does not catch smaller steps that the singer eventually made to return to his or her favorite pitch region.

3.3 Scaling Pitch versus Time

In how we combine d_{wp} and d_t into d_{twp} , we define the time window used for pitch matching. If we put more emphasis on d_{wp} , then we match candidate melodies that have similar pitch material with respect to a rather large temporal segment. We thus consider e.g. pitch sequence (A B C D) similar to (A C B D).

If we put more emphasis on d_t , then we match only temporally very close pitches. Variations like turning half notes into two quarter notes or changing from 4/4 to 6/8 measure will not be matched so easily.

In the original M2M search engine setup, we used pitches given in MIDI-pitch and time given in seconds or quarter notes. There multiplying the time distance with a factor of $\beta = 4$ and taking the Euclidean distance $\sqrt{(\alpha d_{wp})^2 + (\beta d_t)^2}$ did a good job for contour similarity searches. This value might be slightly too large when we compute *OR*-queries on pitches, because we take the minimum distance, which is generally smaller than the pitch to pitch distance in the M2M setup.

3.4 A Simple Example

Figure 2 and table 1 illustrate the effects of α and β for the *OR*-queries in a very simple case. Time is given in quarter notes and pitches in semitones. Each quarter note's weight is 1 and all note duration must flow from the upper melody to the chords, with both chords to be completely matched with duration 2.



Figure 2. Matching the pitch distribution from the accent notes of 3 melodies (lower staff) with a single melody (upper staff). Purely by chance the query melodies form major chords. All \tilde{w}_i are 1 and the \hat{w}_j are proportional to the duration of the notes (*OR*-query). Different α and β can result in different matches for upper notes F and C.

The G flows in any case to the C major chord and the A to the F major chord. The rest depends on the onset scaling factor: If it is large then F prefers a ‘close mismatch’ both in time and pitch and matches with the remaining duration of the C major chord. Note C matches perfectly. If the pitch scaling factor is high, then F matches with the F from the F major chord and the C satisfies the remaining duration of the C major chord. (See table 1.)

Distance	g	f	c	a
$d_{wp}(ceg, x)$	hq:0	1	hq:0	2
$d_{wp}(cfa, x)$	2	hq:0	0	hq:0
$d_t(1, x)$	hq:0	1	2	3
$d_t(2, x)$	1	hq:0	1	2
$d_t(3, x)$	2	1	hq:0	1
$d_t(4, x)$	3	2	1	hq:0
$d_{1m1}(1, x)$	hq:0	h:2	2	5
$d_{1m1}(2, x)$	1	q:1	1	4
$d_{1m1}(3, x)$	4	1	hq:0	h:1
$d_{1m1}(4, x)$	5	2	1	q:0
$d_{4m1}(1, x)$	hq:0	5	h:2	11
$d_{4m1}(2, x)$	1	4	q:1	10
$d_{4m1}(3, x)$	10	hq:1	0	h:1
$d_{4m1}(4, x)$	11	2	1	q:0

Table 1. Distances between the melody notes and the chords. $d_{\alpha m \beta}(n, x)$ denotes Manhattan distance, with pitch distance factor α and time distance factor β between chord at onset position n and the melody tone given in the column header. Minimum WORK is achieved when choosing the marked entries to create a complete match. A h or q in front of a distance value marks inexpensive edges that the EMD chooses to move weight from the melody to the chords. Marker h is valid for the case that we put the half-note chord weight completely onto the strong beats. Marker q is valid for the case that we split each chord into two chords with quarter-note duration.

3.5 Optional Transformations

As usual when computing distances between musical items, we have to consider additional requirements and whether we can implement them with the parameters of the framework or with additional transformations. *Octave invariance* could be implemented by adjusting d_p or by projecting all notes to the same octave before applying any of the pitch dependent distances. *Transposition invariance* can be achieved in a brute force manner if the number of considered transpositions is reasonable limited, e.g. when sticking to the set of MIDI pitches. Then we simply transpose the candidate melody as often as necessary and compute the minimum over the distances of the transposed candidate and the query group. In a similar manner we can handle onset translations when we want to find the best partial match of a query pattern within a candidate melody. (See [4] for a more explained description of how to implement this efficiently.)

Another option is to abstract from or extend the candidate melody and/or the query melodies. By splitting large notes into smaller notes to a common grid, as done in the q -case of table 1, we emphasize less the exact onset of notes but more its duration, resulting in an equivalence class of queries or melodies that all project to the same grid-values. We can also replace metrically or otherwise less important notes with rests or previously occurring important notes. Thereby we make sure that the less important mismatches do not count as much as the important ones. (See [2] for examples on this issue.)

4 EXPERIMENTS

In this section we make use of the computational framework and the parameterization discussions from the previous section to set up and conduct experiments. We first describe a musicological task and a computational goal. Then we analyse an example data set and derive a reasonable parameterization of the computational framework for this case. We calculate the inner group distances (see section 2.4) and query the database.

4.1 The Task and Goal

We want to develop a tool for folk song researchers for the purpose of variation research. Folk song researchers want to know which melodies of a collection stem from each other. To find genetically related items the researchers may first group musically related melodies and then check by means of other information sources, like historical records or geographic information, whether inheritance is plausible.

We translate the first part of this research process into a computational task: for a given group of musically related melodies we want to retrieve and present melodies from a database that are good additional candidate members for that group. Since musicologists (sub-)consciously consider a large, not fully described set of musical features when comparing and classifying melodies, we state a more modest goal: we want to find melodies that follow the notion of stable notes as described in [2] and by this criterion are good group member candidates.

4.2 Example

Figure 3 shows the beginnings of six melodies from the database, which were properly aligned to study the differences between these rather similar melodies. When we follow the lyrics further, we see that variants 1, 3 and 5 (OGL41101m, OGL36012m, OGL33006m) start each odd verse with the same text, like the first verse in bars 1–2 and continue in the even verses differently.

Musically we can identify a core subgroup consisting of items 1, 3, 5 and maybe 4. In this subgroup each verse starts quite similar in its first bar and shows more differences in the second bar. Also we note that metrically strong notes are typically more stable across variants (see [2] for more evidence on this statement). The

remaining two melodies share fewer pitches with the core group but are metrically, harmonically and textually very similar. But from just the *pitch stability* point of view we might want to exclude these melodies from the group.

4.3 Chosen Parameterizations

In a pre-experiment we compute the inner-group distances as described in section 2.4. We express musical time in quarter notes and pitches in MIDI pitch numbers. We define the distance between a MIDI pitch and a rest as 1 and define the distance between MIDI pitches as the absolute difference d_{p1} .

To compare results, we project all melodies to the 8th-note metrical level and thus abstract from mainly textually motivated rhythmic variations. (See [5], file Moeder-Aligned.pdf.) For the weighting between pitch and time dimensions we choose α and β (see section 2.2) from $\{0, 1, 4\}$. For $\beta = 1$ the results are most intuitive (see section 3.4). 4 allows us to compare our search results with those of our previous M2M search engine.

In section 4.4 we compare the different inner-group distance results for different α and β values. For each setting we compute 4 distances: a) With or without the candidate included in the query group. b) Using AND- or OR scheme pitch weights. (For results see [5], file ResultsMIDIPIchDistance.)

4.4 Inner-Group Distances

In this section we discuss the results for different distance measures and α and β values. We use the following terminology: When a melody *scores* high, its *distance* to the group is rather low and we call it a *good match*. When a pitch *does not match* it means that no other member of the group has the same pitch at that onset position.

Setting $\alpha = 0$ and $\beta \geq 0$ ignores pitch. Because of the splitting of notes according to the 8th-note grid duration, there are no rhythmical differences, so the results are not meaningful (all zero).

Figure 3. Beginnings of six manually aligned melodies from the collection *Onder de Groene Linde*. [3] The lyrics correspond to the staves. Please see the remaining bars at [5], file MoederLargeOGLFirst.pdf.

For $\alpha > 0$ and $\beta = 0$ we actually compare pitch histograms for the entire melodies. OGL36012m and OGL41101m score high in OR-queries, while OGL35003m is rather distant, probably because of unmatched (D, E, F#)-phrases in the beginning of odd verses. In AND-queries, OGL33006 performs best, which means, it is most average in pitch.

For $\alpha = 1$ and $\beta = 1$ OGL36012m scores best in OR-queries and can be called the most representative melody. OGL41101m is closest to the average pitch contour. OGL28602m is most distant in both cases, which corresponds nicely with an expert’s decision to put OGL28602m into a different subgroup than the other melodies.

For $\alpha = 1$ and $\beta = 4$ we penalize pitch distances much more. As a result in AND- and OR-queries, OGL37102m is more distant than OGL28602m, which might result from unmatched pitches in the end of bar 1 and beginning of bar 3, which are 7 semitones apart from the other melodies. We think that this is overweighted or should be corrected with a pitch distance measure that takes the harmonic context into account.

For $\alpha = 4$ and $\beta = 1$ pitch differences are weighted high, but we allow to match with neighbor pitches. OGL33006m and OGL36012m are close to the group in OR-queries, additionally OGL41101m again scores best in AND-queries. Worst are OGL35003m and OGL28602m, where the former scores over the latter in OR-queries and vice versa in AND-queries, again probably because of the unmatched (D, E, F#)-phrases.

4.5 Finding Other Candidates

In this section we compare the performance of the group-distance approach with our previous setup, which allowed a M2M query. In a recently established ‘ground truth’ by our expert, there is a so called *melody norm*² *Moeder* which includes our group members and additional items from the database. By constructing ranking lists, we evaluate qualitatively, how useful our method is in showing these items in top positions (small rank numbers).

In a short experiment with the database we restrict ourselves to OR-queries with the following setup: $\alpha = 1, \beta = 1, d_p = d_{p1}$. We do not perform transpositions and time stretching, because the general guideline of the transcriptions of this database was to transcribe it in tonality G (any mode).

Our test collection contains 144³ Dutch folk songs from [3], and 68 unrelated additional popular ring tone melodies. The database contains both complete MIDI files and each of the verses of the folk songs separately, because we look also for melodies that match just in one verse with the query. These kind of partial matches are supported by the EMD. In total there are 212 melodies

² Within the Meertens Institute the concept of melody norm is used to group historically or ‘genetically’ related melodies. If there is not enough historical evidence, such as in this case, then melodies may be attached to a melody norm on the basis of convincing melodic similarity.

³ Three more than used in other ISMIR 2007 publications.

broken into 804 items. Independent from our system our expert assigned 13 melodies to the melody norm *Moeder*, including the 6 query melodies. She created a subgroup ‘1’ with 11 of the melodies and put the remaining two melodies (OGL28602 and OGL33112) into subgroup ‘2’. Our query results (see [5], files *MoederRanksShort* and *MoederResults*) are very promising:

The first 8 items found in the ranking list, including the 6 query melodies, are actual members of the melody norm. The next members are found on ranks 10, 14 and 15. OGL33112, which is part of subgroup 2, is found at rank 22 (of 212). We did not find OGL25309 at the top of the ranking list. It is also part of the melody norm but noted in quarter notes instead of eighth notes.

We compare this findings with the result of a single M2M query with OGL36012m as the best representative of the query group. Here besides OGL25309 we miss 3 items (including OGL33112) in the top ranks. So our G2M appears to work much better than M2M. (See [5], files *OGL36012mRanksShort* and *OGL36012mResults* and figure 4.)

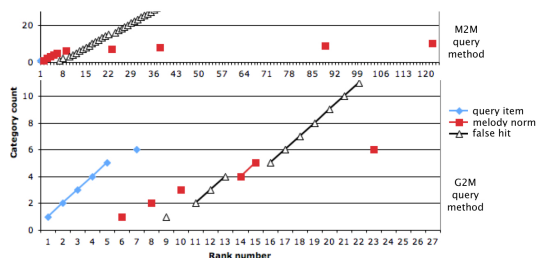


Figure 4. Number of found melodies by category (query item, melody norm item or false hit) counted up to a given rank, shown for M2M and G2M separately (on different scales). Only the first hit of an item is considered in the ranking list. Only melody norm items that are not part of the *query* group are included in the *melody norm* category. M2M: 6 melody norm melodies are not found among the first 20 ranks (melodies at ranks 195 and 200 are not shown). G2M: 2 melody norm melodies are not found among the first 20 ranks (melody at rank 87 is not shown). The complete data can be found at [5], files *MoederResultCategoryCount* and *OGL36012mResultCategoryCount*.

5 CONCLUSION AND FUTURE WORK

We have implemented and tested the method G2M for finding in a database (unclassified) variants, given a group of melodies that are known variants of each other. In our evaluation we found with this method a more complete set of true positives and smaller number of disturbing false positives in the top of a ranking list than with our previous search method M2M, with which Typke won the MIREX 2006 Symbolic Similarity Contest. The results are particularly very promising, because we got there only by reasoning about the free parameters and without manual or automatic fine-tuning.

For the future, a more thorough quantitative evaluation of different cases and different parameterization options remains to be done. We will also compare the group-query results with the ranking that we get with a linear combination of the distances between any member of the group and the candidates. We expect that the former works better with harmonically stable aspects while the latter works better when contour is important in the variant group. To further improve on harmonic generalization from pitches, we consider to apply automatic harmonic analysis. [2]

From the user interface point of view we will study, whether a sort of relevance feedback approach is possible to support the incremental interactive shaping of variant groups. There may be an initial M2M query, for which a ranked list is returned. In this list, the user indicates good hits, which we (semi-)automatically align and use as a G2M query, with new group candidates. Finally, we will elaborate a query formulation and refinement process and allow users to express pitch and timing wildcards.

6 ACKNOWLEDGEMENTS

This work was kindly supported by the Netherlands Organization for Scientific Research within the WITCHCRAFT project NWO 640-003-501, which is part of the program *Continuous Access to Cultural Heritage*. We thank Ellen van der Grijn for the melody norm ground truth.

7 REFERENCES

- [1] Bronson, B.H., "Melodic Stability in Oral Transmission", In: *Journal of the International Folk Music Council*, Vol. 3, 1951, pp. 50-55, 1951.
- [2] Garbers, J., Volk, A., van Kranenburg, P., Wiering, F., Grijp, L., Veltkamp, R. C., "On pitch and chord stability in folk song variation retrieval", In: *Proceedings of the First International Conference of the Society for Mathematics and Computation in Music* (to be published, [5], file *GarbersMCM2007.pdf*), 2007.
- [3] "Onder de groene linde. Verhalende liederen uit de mondelinge overlevering", Uitgeverij Uniepers, Amsterdam, 1987-1991.
- [4] Lubiw, A., Tanur, L. "Pattern Matching in Polyphonic Music as a Weighted Geometric Translation Problem", In: *Proceedings of the 5th International Conference on Music Information Retrieval*, 2004. Also <http://ismir2004.ismir.net/proceedings/p054-page-289-paper154.pdf>.
- [5] Garbers, J., "Additional Resources for ISMIR 2007", <http://www.cs.uu.nl/people/garbers/ISMIR2007/>.
- [6] Rubner, Y., Tomasi, C., Guibas, L. J. "The Earth Mover's Distance as a Metric for Image Retrieval", In: *International Journal of Computer Vision*, vol. 40, no. 2, pp. 99-121, 2000.
- [7] Typke, R. *Music Retrieval Based on Melodic Similarity*. PhD thesis, Utrecht University, 2007.