DTV-BASED MELODY CUTTING FOR DTW-BASED MELODY SEARCH AND INDEXING IN QBH SYSTEMS

Bartłomiej Stasiak
Institute of Information Technology, Lodz University of Technology,
bartlomiej.stasiak@p.lodz.pl

ABSTRACT

Melody analysis is an important processing step in several areas of Music Information Retrieval (MIR). Processing the pitch values extracted from raw input audio signal may be computationally complex as it requires substantial effort to reduce the uncertainty resulting i.a. from tempo variability and transpositions. A typical example is the melody matching problem in Query-by-Humming (QbH) systems, where Dynamic Time Warping (DTW) and note-based approaches are typically applied.

In this work we present a new, simple and efficient method of investigating the melody content which may be used for approximate, preliminary matching of melodies irrespective of their tempo and length. The proposed solution is based on Discrete Total Variation (DTV) of the melody pitch vector, which may be computed in linear time. We demonstrate its practical application for finding the appropriate melody cutting points in the R^*-tree-based DTW indexing framework. The experimental validation is based on a dataset of 4431 queries and over 4000 template melodies, constructed specially for testing Query-by-Humming systems.

1. INTRODUCTION

Content-based search and retrieval is becoming a popular and attractive way to locate relevant resources in the evergrowing multimedia collections and databases. For Music Information Retrieval (MIR) several important application areas have been defined over the last decades, with *Audio Fingerprinting*, and *Query by Humming* (QbH) being perhaps the most prominent examples. The latter one is specific as it is exclusively based on the user-generated sound signal and it depends mostly on a single parameter of this signal – the pitch of the consecutive notes, forming the melody sung by the user.

In a typical QbH system the *query* in the form of raw audio data is subjected to a pitch-tracking procedure, which yields a sequence of pitch values in consecutive time frames, often referred to as a *pitch vector*. The music resources in the database are represented by *templates*, having the similar form, so that the search is essentially based on simply comparing the pitch vectors in order to find the template melody best matching the query melody.

An additional step of note segmentation may be used to obtain symbolic representation, explicitly defining the pitch and length of separate notes. In this case, several efficient methods based on e.g. transportation distance or string matching algorithms may be used. This approach enables fast searching, although it is difficult to obtain high precision due to unavoidable ambiguities of the note segmentation step. Comparing the pitch vectors directly usually yields higher-quality results but on the cost of the increased computational complexity, as the local tempo variations require to use tools for automatic alignment between the compared melodies.

Dynamic Time Warping (DTW) is a standard method applied for comparing data sequences, generated by processes that may exhibit substantial, yet semantically irrelevant local decelerations and accelerations. The examples include e.g. handwritten signature recognition or gait recognition for biometric applications, sign language analysis, spoken word recognition and many other problems involving temporal patterns analysis. It is also a method of choice for direct comparison of pitch vectors in the Query by Humming task.

2. PREVIOUS WORK

Early works on the QbH systems date back to the 1990's, with some background concepts and techniques being introduced much earlier [21, 24]. Initially, the symbolic, note-based approach was used [6, 20, 30], often in the simplified form comprising only melody direction change (U/D/S - Up/Down/the Same) [6, 21]. In the following decade the direct pitch sequence matching with Hidden Markov Models (HMM) [26] and Dynamic Time Warping [11, 19] was proposed and extensively used, in parallel to note-based approaches employing transportation distances, such as Earth Mover's Distance (EMD) [28, 29]. In many practical QbH systems, such as those presented in the annual MIREX competition [1, 27], a multi-stage approach is applied involving the application of the EMD to filter out most of the non-matching candidate templates, leaving only the most promising ones for the accurate, but more computationally expensive DTW-based search [31, 32, 34].

Another possibility of search time optimization is to accelerate the computation of DTW itself. For this purpose several methods have been proposed, including iterative deepening [2], Windowed Time Warping [18], FastDTW [25] and SparseDTW [3].

Yet another approach, which is of special interest to us, is to apply efficient DTW indexing techniques, based on lower bounding functions [13]. These methods reduce computational complexity by limiting the number of times the DTW algorithm must be run, but – unlike the aforementioned EMD-based multi-stage systems – they are not domain specific. Introduced by Keogh [13, 16] as a general-purpose method for time-series matching, they have also been successfully applied for the Query by Humming task [17, 35].

2.1 Indexing the dynamic time warping

DTW indexing is based on a more general approach to indexing time series, known as GEMINI framework (GEneric Multimedia INdexIng) [5, 14]. In this approach the sequences are indexed with R-trees, R*-trees or other Spatial Access Methods (SAM) [7] after being subjected to dimensionality reduction transformation. The typical SAMs require that the data are represented in a not more than 12-16–dimensional index space I [14, 15]. Searching in the index space is guaranteed to yield a superset of all the relevant templates (i.e. it will produce no false rejections), provided that a proper distance measure ρ_I is defined in I. Let N denote the length of the original time series and let $M \ll N$ be the number of dimensions of the index space. It may be shown [5] that when the distance ρ_X between elements T_N, Q_N of the input space X is properly bounded from below by the distance between their low-dimensional representations T_M, Q_M in the index space, i.e. when:

$$\rho_I(T_M, Q_M) \leq \rho_X(T_N, Q_N) \ , \tag{1}$$

then it is possible to construct an indexing mechanism which guarantees no false dismissals. The efficient, SAM-optimized query in the index space may only return some false positives which are then eliminated by direct matching of the time series in the original, input space X. Depending on the tightness of the lower bound (Eq. 1), the number of times the matching must be done in X may be reduced even by orders of magnitude. The detailed description of the appropriate algorithms for k-nearest neighbor search and range queries may be found in [5, 14].

The generality of the GEMINI framework enables its application with many dimensionality reducing transforms, based on e.g. discrete Fourier transform (DFT) [5], Haar Wavelet Transform [12] or piecewise aggregate approximation (PAA) [14, 33]. However, comparing sequences under dynamic time warping differs quite significantly from the case of Euclidean spaces, i.a. because DTW is not – strictly speaking – a *metric* (it does not satisfy the triangle inequality). It has been however shown that it is possible to define a valid lower-bounding distance measure [13] when proper global constraints, such as Sakoe and Chiba band [24] or Itakura parallelogram [9] are used. The dimensionality reduction may be obtained by the simple PAA algorithm, as demonstrated by Keogh in [13]. A more general approach, based on properties of container-invariant time series envelopes, was introduced by Zhu and Shasha, who extended the lower-bounding criterion to the whole class of linear transforms [35].

The aforementioned techniques of DTW indexing may be successfully applied to accelerate the melody matching in Query by Humming systems, as demonstrated in [35] on an example of a small music database of 50 Beatles songs. However, in real-life, large-scale systems some practical problems are likely to occur, especially when heterogeneous audio material is used as input for querying the database.

One of these problems is that the actual length and tempo of the query, with respect to the potentially matching template, are not known in advance. As we will demonstrate in the following section, this uncertainty makes the use of the global constraints of the DTW difficult, which in turn put in doubt the practical applicability of the DTW indexing schemes in the Query by Humming task.

As a remedy, we propose a novel solution, based on computation of Discrete Total Variation (DTV) of the pitch vector, which enables to assess the optimal cutting point of the query with respect to the template (Sect. 4). In this way, the DTW indexing becomes feasible even for diversified input data, containing the queries of varied length and tempo. Moreover, the analysis of the DTV may appear beneficial also for fixed-length queries. This conclusion will be supported by the experimental results presented in Section 5.

3. PROBLEM SETTING

The implicit assumption underlying the efficient DTW indexing methods introduced in [13] is that the beginning and the end of both compared sequences coincide. Unfortunately, in the query-by-humming task, this condition is rarely met, especially with respect to the ending point. The beginning is less problematic because most users typically sing from the beginning of a phrase [8]. The *length* of the query, on the other hand, is often unknown in advance, both in terms of absolute time units and with relation to the template. It is possible to control the absolute length of the query in the acquisition module, e.g. by stopping the recording session after x seconds. Even then, however, the assessment of the exact number of notes or phrases sung is impossible, mainly due to tempo differences between users. The query may therefore end anywhere within the course of the template, as illustrated in Fig. 1.

Fortunately, the DTW may deal with this case quite easily, as the endpoint of the warping path may effectively be searched for along the last row of the DTW matrix (or along the last column, if we also expect queries longer than templates). The real problem is, however, that it is now impossible to effectively use the global constraints, such as Sakoe and Chiba band (Fig. 1b), which are the *sine qua non* condition in the DTW indexing techniques [13, 35].

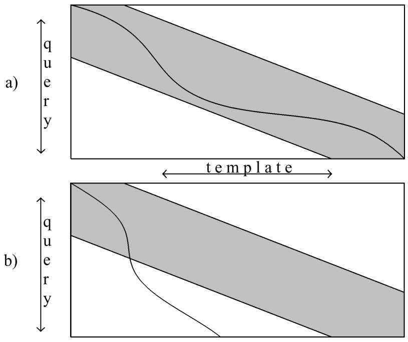

Figure 1. Optimal warping path in the DTW matrix (with Sakoe and Chiba band shown) for fixed-length query, where the query is much shorter than the template: a) The "fast singer" case – both the beginning and the ending points coincide; b) The "normal singer" case – the query ends in the middle of the template melody.

Relaxing the constraints (e.g. increasing the radius of the band) may help to incorporate more queries deviating from the diagonal, but on the cost of making the indexing less efficient. Hence, although in theory the lower bounding techniques guarantee no false dismissals, we arrive again at the trade-off between time-efficiency and retrieval rate. In fact, setting the proper constraints is always a matter of a compromise, but here the problem is much more severe, as even moderate tempo mismatch may lead to significant accumulation of deviations at the end of a query.

A real-life example – a template and the matching queries from Jang's dataset [10] – is presented in Fig. 2 where the ending point of each query, with respect to the template, has been determined on the basis of the optimal warping path in the DTW matrix. Fig. 2 reveals, that although within the fixed time of 8s most users managed to sing between two and three two-bar motifs (out of all four), there were also some "lazy singers" that did not manage to complete two motifs and some "fast singers" who completed the whole or almost the whole melody. With indexing, these queries may be easily lost, unless some extremely loose constraints were applied.

Let us note that the problem occurs even for optimal template length (e.g. from Fig. 2 we may conclude that this particular template is actually too long). In fact, in many datasets the templates tend to be much longer than the queries. For example, in Jang's [10] collection the template length varies from ca. 12 seconds up to over 5 minutes. Hence, determining the reasonable cutting point for the templates, prior to indexing, becomes a necessary preliminary step of processing.

It is important to note that this step cannot be done reliably without some form of melodic content analysis. Naturally we might try – for fixed query length of x seconds – to cut the templates to the same x seconds, assuming that the tempo of the template roughly corresponds to the mean tempo of the queries. However, we have no guarantee that this assumption is correct, which may obviously lead to suboptimal indexing results.

In the following section we present an automatic method for determining the cutting point of the template melody. The same method is also applied to each query to cut it at the point corresponding to the cutting point of the template.

Although it may seem not obvious, we should note that we also touch the problem of query transposition here. A user may sing the melody in any key, so it must be transposed to a reference key before matching, which is typically done by mean subtraction. However, the mean pitch of a melody obviously depends on the location of its ending point, which gives an additional motivation for trying to agree on a common cutting point among all the potentially matching melodies. The proposed method is based on a simple content-based filter, which enables the preliminary match of the lengths of the compared melodies and – in consequence – the practical use of the efficient indexing algorithms.

4. THE DISCRETE TOTAL VARIATION

An intuitive solution to our problem might be formulated as follows: given a perfect note segmentation of the audio files we could cut every melody after a fixed number of notes (the same for all melodies – templates and queries). This would guarantee the endpoint match for efficient indexing and the consecutive DTW would successfully deal with potential rhythm and tempo discrepancies. Unfortunately, while it is straightforward for MIDI-based templates, it is not so for the sung queries. The singer's imprecision on one hand and the specificity of a given pitch tracking algorithm on the other hand may lead to note segmentation errors that will make this approach unusable.

In our approach, instead of a crisp note segmentation we prefer to construct a soft measure of pitch value variability in time. In continuous case, for $p(\tau)$ representing the pitch value at time τ, we would define the following functional:

$$TV(t) = \int_0^t \left| \frac{d}{d\tau} p(\tau) \right| d\tau \; . \tag{2}$$

We may note that this definition, corresponding to the L_1 norm of the pitch signal derivative, may be seen as a one-dimensional version of Total Variation (TV) as introduced by Rudin [22, 23] in the image analysis and noise removal context. The one-dimensional Discrete Total Variation DTV may be defined as:

$$DTV(n) = \sum_{k=1}^{n} |p(k) - p(k-1)| \; , \tag{3}$$

where $p(k)$ denotes the k-th time frame of the pitch vector.

The fundamental property of DTV is that it accumulates pitch changes in the course of the melody, irrespective of the actual direction of the changes (Fig. 3). We may therefore set a threshold T_{DTV} for the accumulated pitch changes and cut all the compared melodies when they

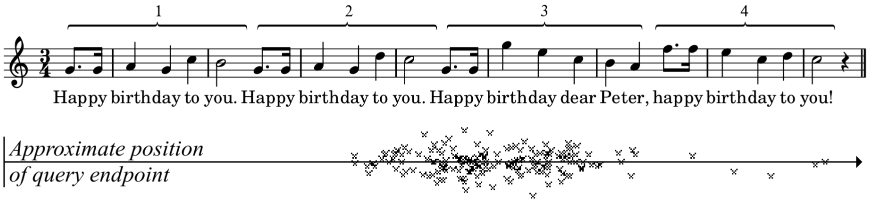
1
2
3
4
Happy birthday to you. Happy birthday to you. Happy birthday dear Peter, happy birthday to you!
Approximate position
of query endpoint

Figure 2. Example template from Jang's [10] collection (top) and the ending points of all the 170 matching queries (bottom).

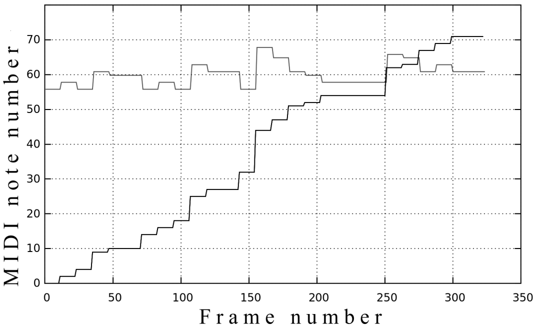

Figure 3. Pitch vector representation of the melody from Fig. 2 (top, light-grey) and the corresponding DTV sequence (bottom, dark-grey).

reach T_{DTV}, as follows:

$$p_c = [p(0), p(1), ..., p(n_c)] \ , \tag{4}$$

where p_c denotes the pitch vector reduced to the first n_c+1 values and where:

$$n_c = \min\left\{n \in N; DTV(n) \geq T_{DTV}\right\} \ . \tag{5}$$

The proposed DTV-based cutting scheme has some further properties that are relevant to the considered problem:

1. The DTV sequence is monotonically nondecreasing and it stays constant only within segments of fixed pitch. The latter fact means that the note lengths are basically ignored – only the number of notes and the span of the consecutive musical intervals (pitch difference) influence the increase of the value of DTV.

2. Ignoring the direction of the pitch changes means that the DTV is not unique. For example, ascending chromatic scale will yield the same DTV sequence as the descending one, assuming the same tempo and the same number of notes (diatonic scale would produce different results due to different ordering of whole steps and half steps).

3. DTV-based cutting leads to obtaining the melody representation robust to *glissandi*, occurring frequently in sung queries, where the pitch changes are "spread" over several consecutive time frames.

4. DTV-based cutting leads to obtaining the melody representation which is not robust to jitter and vibrato, which may be present within single notes, i.e. in segments of – otherwise constant – pitch.

Property 1 implies a fundamental fact that two versions of a melody, consisting of identical pitch sequences but with different rhythm and tempo will yield the same DTV sequence for corresponding notes. In particular, their representations obtained with Eq. 4 may have different number of frames, but they will basically represent the same *melodic content*.

Property 2 means that the DTV may be interpreted as a hash function which may occasionally return equal values for dissimilar input data. In fact, what we need is the opposite implication: the results for similar input must be also similar and – fortunately – this condition is generally fulfilled.

Property 3 is connected to an important advantage of the proposed method to ignore the slope of the pitch changes. When singing a musical interval, the second note is often reached after several frames of intermediate pitch values, as opposed to MIDI-based signals where the changes are instantaneous. This difference is well visible in Fig. 4 (top plots). It can be observed that despite the fuzzy note transitions in frames 30–33 and 51–59 of the query (plot **a**)), the obtained DTV sequences (Fig. 4, the bottom plots) are indeed similar. Therefore setting a given threshold value T_{DTV} in Eq. 5 would allow to obtain the similar melody sections both for the query and for the template, irrespective of the significant tempo discrepancy between the two. For example, for $T_{DTV} = 5$ both sequences would be cut at the onset of the 4th note.

Property 4 indicates a potential weakness of the proposed solution as jitter and vibrato are ubiquitous in pitch vectors of sung melodies. However, a popular and simple median filter, which is often used to pre-process the pitch vectors prior to further melody analysis, may be effectively applied here to suppress the minor pitch fluctuations. The example in Fig. 4 had been already filtered with median filter of 9^{th} order which had appropriately smoothed almost the whole signal, except of the small artefact in frames 86–88. As a comparison, Fig. 5 **a)** presents the original query. The dramatic distortion of the obtained DTV sequence may be assessed even better on the right plot (**b)**), where we see over twofold increase of the accumulated pitch changes for the unfiltered query with respect to the filtered one.

5. EXPERIMENTAL VALIDATION

In order to evaluate the usefulness of the proposed method in melody indexing, we performed tests on the well-

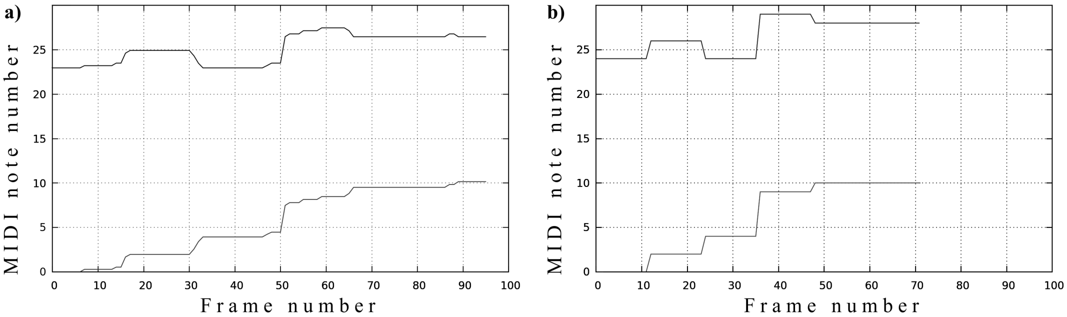

Figure 4. The first motif of the melody from Fig. 2: **a)** query; **b)** template. Top plots present the original pitch vectors (after transposition by 3 octaves down, for visualization purposes); the bottom plots show the DTV sequences.

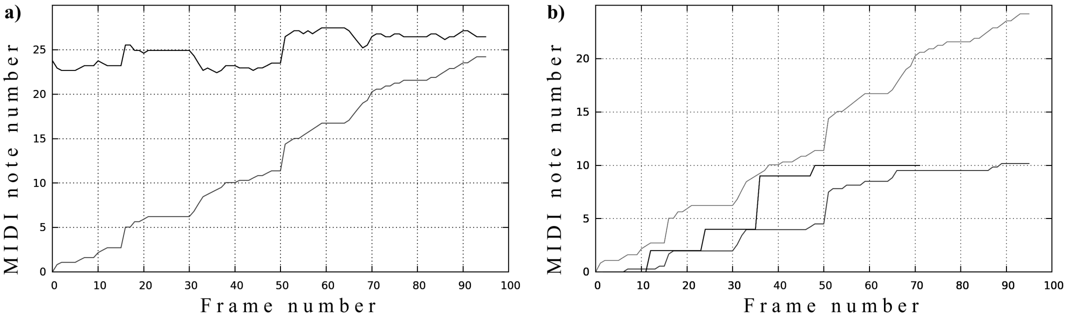

Figure 5. **a)** the same melody as in Fig. 4, but without the median filter; **b)** comparison of the obtained DTV sequences: query without median filter (light grey, top); template (black, middle); query after median filter (dark grey, bottom).

known, publicly available dataset, consisting of a collection of 48 popular songs (in the form of ground-truth MIDI files) to be matched against 4431 queries sung by about 200 users [10]. In order to increase the difficulty of the problem, the set of templates was artificially expanded, so that it contained – apart from the 48 ground-truth files – 4225 additional noise midi files from Essen collection [4].

Piecewise aggregate approximation was applied as a dimensionality reduction technique. Each template melody was represented as a point in 16-dimensional index space where R^*-tree was used as the spatial index. For each query melody the minimal bounding rectangle (MBR) was computed and used for k-NN search, according to [15]. Two quantities were measured during the tests: the CPU time of computation and the number of correctly recognized queries. The baseline results obtained by a non-indexing system, computing the DTW match between every query and every template, were: 4211 out of 4431 queries recognized (95.03%) in 48h 55m 28s.

For the indexing tests several melody cutting strategies were applied and the corresponding results have been presented in Fig. 6. All the queries in the test database are of the same length of 8s. Our first attempt was therefore to apply the straightforward approach based on cutting the templates to the same 8s (250 frames with a step of 32 ms) prior to indexing. This in fact yields the optimal template length also in terms of the melody content because, as we have found, there is no bias towards faster or slower queries in the test database, i.e. the mean tempo of each query is basically the same as the tempo of the corresponding template.

However, it appears that even in this optimal setting, our DTV-based cutting scheme may increase the recognition rate with respect to the fixed template cutting point (for the same number of nearest neighbors). For example, the recognition rate obtained for the fixed-length templates with 1500 nearest neighbors could be obtained for the templates cut on the basis of their DTV with 1100 nearest neighbors, which means ca. 25% gain in the computation time (Fig. 6).

The key point in the effective application of the DTV is setting the proper threshold T_{DTV}. Too low value leads (Fig. 6, $T_{DTV} = 20$) to extracting and indexing very short melody fragments, which means that they contain few notes and hence many templates may even become indistinguishable from each other. Moreover, for short queries the lower bounding measure often happens to be zero which prevents establishing the right order of the results.

On the other hand, too high T_{DTV} threshold means that many queries will not reach it before their "hard" cut point (8s in our case). This problem will not occur in the case of the templates, because they are usually much longer. As a result, after the cutting procedure the templates will be generally longer than the queries and they will also contain much more melodic material, which will make the indexing ineffective.

As a partial remedy, we propose to use a simple condition, limiting the absolute length of the templates to the

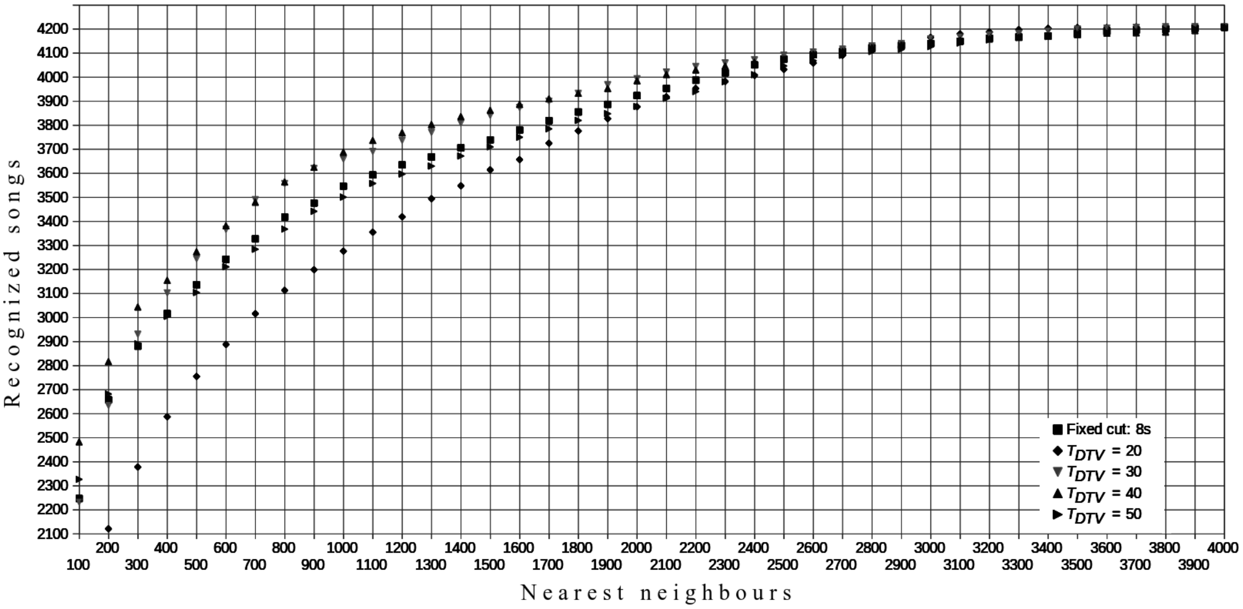

Figure 6. Top-ten results for various cutting point setting

absolute length of the queries:

$$\hat{n}_c = \min\{n_c, |q|\} \tag{6}$$

where n_c is given by Eq. 5 and $|q|$ is the fixed length of the queries. In this way the template lengths are limited similarly as the queries. We should note that the only problem which may occur here is when some monotonous, not-much-varied melodies are sung by a "fast singer", because these queries will contain more melodic material than – then prematurely cut – template. This problem generally cannot be avoided without *a priori* knowledge on the correct classification of the query, but it appears not to have much impact on the recognition rate. The results in Fig. 6 have been obtained with template length limiting (Eq. 6) and as we may see they are stable in a quite broad range of the threshold values ($T_{DTV} = 30$, $T_{DTV} = 40$). Going beyond this optimal range ($T_{DTV} = 50$) deteriorates the recognition but still the obtained results are only slightly worse that the fixed cutting point approach.

6. CONCLUSION

In this work we have introduced a new method of determining the optimal cutting point for melody comparisons, based on Discrete Total Variation of the melody pitch vector. Our solution is fast to compute and it yields useful information about the melody, which enables to effectively apply DTW indexing strategies, introduced in [13]. We demonstrated the usefulness of the proposed solution for the indexing task on a known database, designed for testing Query-by-Humming systems. It should be noted, however, that the method has potentially much broader application area. In particular, much more significant gain may be expected in less constrained, on-line QbH systems, especially when more relaxed limits of the query length are used, and/or when greater singing tempo discrepancies may be expected. The ability to find the appropriate melody length in a fast way, without detailed note-based analysis and without computationally expensive DTW is an advantage which may simplify and accelerate many content-based music information retrieval tasks.

7. REFERENCES

[1] http://www.music-ir.org/mirex.

[2] N. Adams, D. Marquez, and G. Wakefield. Iterative Deepening for Melody Alignment and Retrieval. In *6th Int. Conf. on Music Information Retrieval, ISMIR 2005*, pages 199–206, 2005.

[3] G. Al-Naymat, S. Chawla, and J. Taheri. SparseDTW: A Novel Approach to Speed Up Dynamic Time Warping. In *Proc. of the 8th Australasian Data Mining Conf.*, pages 117–127, 2009.

[4] http://www.esac-data.org, 2009.

[5] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos. Fast subsequence matching in time-series databases. In *SIGMOD1994*, pages 419–429. ACM, 1994.

[6] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith. Query By Humming – Musical Information Retrieval in an Audio Database. In *Proc. of the 3rd ACM Int. Conf. on Multimedia*, pages 231–236, 1995.

[7] A. Guttman. R-Trees: A Dynamic Index Structure for Spatial Searching. In *SIGMOD 1984*, pages 47–57. ACM Press, 1984.

[8] S. Huang, L. Wang, S. Hu, H. Jiang, and B. Xu. Query by humming via multiscale transportation distance in

random query occurrence context. In *IEEE Int. Conf. on Multimedia and Expo*, pages 1225–1228, 2008.

[9] F. Itakura. Minimum Prediction Residual Principle Applied to Speech Recognition. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 23(1):67–72, 1975.

[10] http://mirlab.org/dataSet/public/MIR-QBSH-corpus.rar, 2009.

[11] J.-S. R. Jang and H.-R. Lee. Hierarchical filtering method for content-based music retrieval via acoustic input. In *Proc. of the 9th ACM Int. Conf. on Multimedia*, pages 401–410, 2001.

[12] F. K.-P. Chan, A. W.-C. Fu, and C. Yu. Haar Wavelets for Efficient Similarity Search of Time-Series: With and Without Time Warping. *IEEE Trans. on Knowl. and Data Eng.*, 15(3):686–705, 2003.

[13] E. Keogh. Exact indexing of dynamic time warping. In *Proc. of the 28th Int. Conf. on Very Large Data Bases*, VLDB '02, pages 406–417, 2002.

[14] E. Keogh, K. Chakrabarti, M. Pazzani, and S. Mehrotra. Dimensionality reduction for fast similarity search in large time series databases. *Knowledge and information Systems*, 3(3):263–286, 2001.

[15] E. Keogh and C. A. Ratanamahatana. Exact Indexing of Dynamic Time Warping. *Knowl. Inf. Syst.*, 7(3):358–386, 2005.

[16] E. J. Keogh. Efficiently Finding Arbitrarily Scaled Patterns in Massive Time Series Databases. In *PKDD*, volume 2838 of *Lecture Notes in Computer Science*, pages 253–265. Springer, 2003.

[17] E. Lau, A. Ding, and J. Calvin. MusicDB: A Query by Humming System. Final Project Report, Massachusetts Institute of Technology, USA, 2005.

[18] R. Macrae and S. Dixon. Accurate Real-time Windowed Time Warping. In J. S. Downie and R. C. Veltkamp, editors, *Proc. of the 11th Int. Society for Music Information Retrieval Conf., ISMIR 2010*, pages 423–428, 2010.

[19] D. Mazzoni and R. B. Dannenberg. Melody Matching Directly From Audio. In *2nd Annual Int. Symposium on Music Information Retrieval, ISMIR 2001*, pages 73–82, 2001.

[20] R. J. McNab, L. A. Smith, I. H. Witten, C. L. Henderson, and S. J. Cunningham. Towards the digital music library: tune retrieval from acoustic input. In *Proc. of the 1st ACM Int. Conf. on Digital Libraries*, DL '96, pages 11–18, 1996.

[21] D. Parsons. *The Directory of Tunes and Musical Themes*. S. Brown, 1975.

[22] L. A. Rudin. *Images, Numerical Analysis of Singularities and Shock Filters*. PhD thesis, 1987.

[23] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear Total Variation Based Noise Removal Algorithms. *Phys. D*, 60(1-4):259–268, November 1992.

[24] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978.

[25] S. Salvador and P. Chan. FastDTW: Toward accurate dynamic time warping in linear time and space. In *3rd Workshop on Mining Temporal and Sequential Data*, 2004.

[26] J. Shifrin, B. Pardo, and W. Birmingham. HMM-Based Musical Query Retrieval. In *Joint Conf. on Digital Libraries. 2002. Portland, Oregon*, pages 295–330, 2002.

[27] B. Stasiak. Follow That Tune - Adaptive Approach to DTW-based Query-by-Humming System. *Archives of Acoustics*, 39(4):467 – 476, 2014.

[28] R. Typke, P. Giannopoulos, R. C. Veltkamp, F. Wiering, and R. van Oostrum. Using transportation distances for measuring melodic similarity. In *ISMIR 2003*, pages 107–114, 2003.

[29] R. Typke, F. Wiering, and R. C. Veltkamp. Transportation distances and human perception of melodic similarity. *Musicae Scientiae*, pages 153–181, 2007.

[30] A. L. Uitdenbogerd and J. Zobel. Manipulation of Music for Melody Matching. In *Proc. of the 6th ACM Int. Conf. on Multimedia*, pages 235–240. ACM, 1998.

[31] L. Wang, S. Huang, S. Hu, J. Laing, and B. Xu. An effective and efficient method for query by humming system based on multi-similarity measurement fusion. In *Int. Conf. on Audio, Language and Image Processing*, pages 471–475, 2008.

[32] L. Wang, Sh. Huang, Sh. Hu, J. Liang, and B. Xu. Improving searching speed and accuracy of query by humming system based on three methods: feature fusion, candidates set reduction and multiple similarity measurement rescoring. In *INTERSPEECH*, pages 2024–2027. ISCA, 2008.

[33] B.-K. Yi and C. Faloutsos. Fast Time Sequence Indexing for Arbitrary Lp Norms. In *Proc. of the 26th Int. Conf. on Very Large Data Bases*, VLDB '00, pages 385–394, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[34] B. Zhu and H. Liu. MIREX 2015 QBSH task: Tencent Bestimage's solution. pages 1–2, 2015.

[35] Y. Zhu and D. Shasha. Warping indexes with envelope transforms for query by humming. In *Proc. of the 2003 ACM SIGMOD Int. Conf. on Management of Data*, pages 181–192, 2003.