PAN ARCHIVES OF ACOUSTICS
Vol. **39**, No. 4, pp. 467–476 (2014)

DOI: 10.2478/aoa-2014-0050

Research Papers

Follow That Tune – Adaptive Approach to DTW-based Query-by-Humming System

Bartłomiej STASIAK

Institute of Information Technology, Lodz University of Technology
Wólczańska 215, 90-924 Łódź, Poland; e-mail: basta@ics.p.lodz.pl

(*received February 4, 2013; accepted February 7, 2014*)

Dynamic Time Warping is a standard algorithm used for matching time series irrespective of local tempo variations. Its application in the context of Query-by-Humming interface to multimedia databases requires providing the transposition independence, which involves some additional, sometimes computationally expensive processing and may not guarantee the success, e.g., in the presence of a pitch trend or accidental key changes.

The method of *tune following*, proposed in this paper, enables solving the pitch alignment problem in an adaptive way inspired by the human ability of ignoring typical errors occurring in sung melodies. The experimental validation performed on the database containing 4431 queries and over 5000 templates confirmed the enhancement introduced by the proposed algorithm in terms of the global recognition rate.

Keywords: Query-by-Humming, Dynamic Time Warping, MIREX, tune-following.

1. Introduction

The impressive diversity of methods and goals formulated in the area of Music Information Retrieval (MIR) reflects the intrinsic complexity of our perception of music and of music itself. Out of the many research issues considered in the field, the problem of query specification for content-based music retrieval has been attracting significant attention for years. Among many proposed solutions, such as Query by Tapping, pitch contour specification with Parsons code or various forms of simplified musical notation, the Query by Singing/Humming (QbSH) interface is perhaps one of the most natural approaches to searching for a piece of music in multimedia databases.

The main issue in QbSH problem is basically *melody matching* where the melody is understood as a sequence of notes with given pitches and durations. The database entities (*templates*) are often given already in this form, although it should be noted that identification of representative fragments (Głaczyński, Łukasik, 2013) and melody extraction (Salamon, Gómez, 2012; Lau *et al.*, 2005) from original music files is a far from trivial task itself.

Converting the user's input – a sung or hummed melody – into a sequence of pitch values, referred to as a *pitch vector*, is a typical preliminary step of processing. Many pitch detection algorithms (PDA) are available for this purpose (Dziubiński, Kostek, 2005; Gerhard, 2003; Boersma, 1993), so a reliable representation may be usually obtained even in a relatively noisy environment. The potential problems involved here include the frequency resolution and precision of the PDA (usually of minor significance in the QbSH task), octave errors (may occasionally become an issue), and the imprecision of the sung query itself, which is one of the main sources of confusion in practice.

The precise onset time and duration of a note are more difficult to be unambiguously determined. This is a point at which the approaches used for solving the QbSH problem may be roughly divided into two main groups.

Note-based approaches. These methods aim at obtaining a reliable note segmentation with respect to the pitch and temporal parameters. Their greatest advantage is a compact representation allowing for efficient melody searching with string matching algorithms (Ghias *et al.*, 1995). The methods proposed here include edit distance computation based on note insertion/deletion/replacement cost (McNab *et al.*, 1996), transportation distances such as the Earth Mover Distance (EMD) (Typke *et al.*, 2007; Huang *et al.*, 2008) and n-grams matching (Uitdenbogerd, Zobel, 1999; Wolkowicz *et al.*, 2008). The note-based methods rely on the quality of the note segmentation stage which generally makes them potentially imprecise and dependent on the underlying onset detection algorithms. This is in fact a separate MIR research area where several additional factors, e.g., related to timbre, type of accents or musical expression, must be taken into account (McNab *et al.*, 1996; Eyben *et al.*, 2010; Bisesi, Parncutt, 2013).

Direct matching. In these approaches the note segmentation problem is deliberately ignored and the pitch vectors are directly compared on a per-frame basis. High matching precision may be usually obtained in this way but at the cost of increased computational complexity (Jang, Lee, 2001; Zhu, Shasha, 2003). Not only is the melody representation much longer than the sequence-of-notes form, but also variations of tempo in the sung query make the standard Euclidean distance between vectors inaccurate and a more sophisticated matching algorithm must be applied.

The method of choice for aligning the query with a template via a non-linear scaling of the time domain is known as Dynamic Time Warping (DTW). Proposed initially for isolated words recognition (Itakura, 1975; Sakoe, Chiba, 1978) it has been widely adopted in many other fields of artificial intelligence and signal processing.

One of the fundamental issues in a practical application of the DTW algorithm for melody matching is to obtain a key-invariant representation. The melody is defined by a sequence of relative pitches, so their absolute values are basically irrelevant. The user can sing a melody in any key, so all the notes may be shifted with respect to the template by the same interval, which may result in a large value of the DTW distance, even for a perfectly sung query. In this paper a novel solution is proposed, in which the query is "tuned in" to the template via gradual decrease of the pitch difference between the two.

In the next section the principles of the DTW algorithm will be briefly presented along with a summary of previous works which influenced the development of the method in the context of QbSH and melody matching problems. Next, the proposed modification of the algorithm and the results of experiments demonstrating the obtained enhancement in recognition rate will be presented.

2. Basic concepts

2.1. Previous work

The problem of minimizing the distance between two time series which may vary in time or speed occurs naturally in numerous application areas. Early works of Itakura (1975) and of Sakoe and Chiba (1978) introduced the DTW as an effective solution in the speech processing task. The fundamental concepts laid out there have been later used with slight modifications in many fields of artificial intelligence and data mining, including audio and video stream monitoring, biomedical signal inspection, financial data analysis, human motion and gesture recognition (Sakurai *et al.*, 2007; Keogh, 2002). The variants of the method include full sequence matching (Sakoe, Chiba, 1978) and subsequence matching (Sakurai *et al.*, 2007; Lijffijt *et al.*, 2010). Efficient indexing techniques allowing to significantly reduce the searching time in large databases were introduced by Keogh (2002) and applied in the Musical Information Retrieval context by Zhu and Shasha (2003) and Lau *et al.* (2005). Several solutions regarding the speed vs accuracy tradeoff have been proposed, including iterative deepening (Adams *et al.*, 2005), Windowed Time Warping (Macrae, Dixon, 2010), and FastDTW (Salvador, Chan, 2004). The application of several variants of the DTW algorithm for the QbSH problem has been addressed in numerous works, including Jang and Lee (2001), Lijffijt *et al.* (2010), Yu *et al.* (2008), Jeon and Ma (2011) and Wang *et al.* (2008).

2.2. The fundamentals of Dynamic Time Warping

Let q_j denote the pitch value in the j-th frame of the query pitch vector $\mathbf{q}$, where $j = 1, 2, ..., J$. Similarly, t_i represents the i-th frame of the template $\mathbf{t}$, where $i = 1, 2, ..., I$. The Euclidean distance between the two:

$$d_{\text{Euclid}}(\mathbf{q}, \mathbf{t}) = \sqrt{\sum_i |q_i - t_i|^2}, \qquad (1)$$

may be computed only if the size of the two vectors is the same, which is typically not the case. Moreover, reinterpolating the sequences linearly to the same length may not be sufficient in the presence of local tempo variations (Fig. 1).

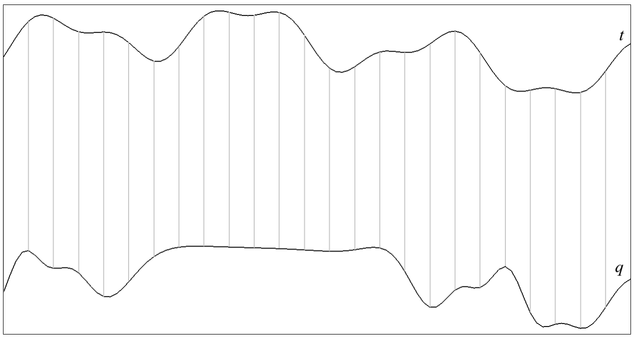

Fig. 1. Sequence matching with the Euclidean distance.

The solution is to scale the time domain of the sequences with a proper *warping function* so that the corresponding frames are properly matched

(Fig. 2). The warping function may be represented on the i-j plane by a path, i.e., a sequence of points $c(1), c(2), ..., c(K)$, where $c(k) = (i(k), j(k))$ (Fig. 3). Every path is assigned a cost:

$$E = \sum_{k=1}^{K} d(c(k)), \tag{2}$$

where the cost of matching an individual point $c(k)$ may be defined as:

$$d(c(k)) = d(i, j) = |q_{j(k)} - t_{i(k)}|. \tag{3}$$

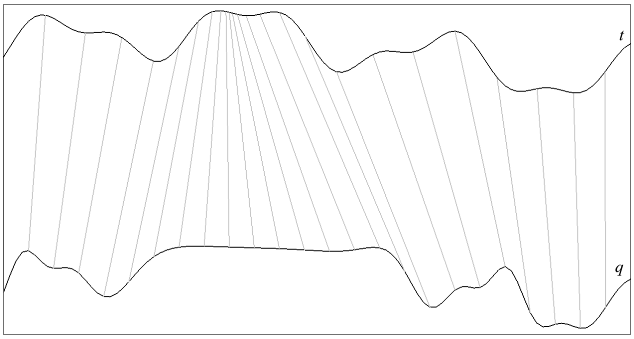

Fig. 2. Sequence matching after non-linear rescaling.

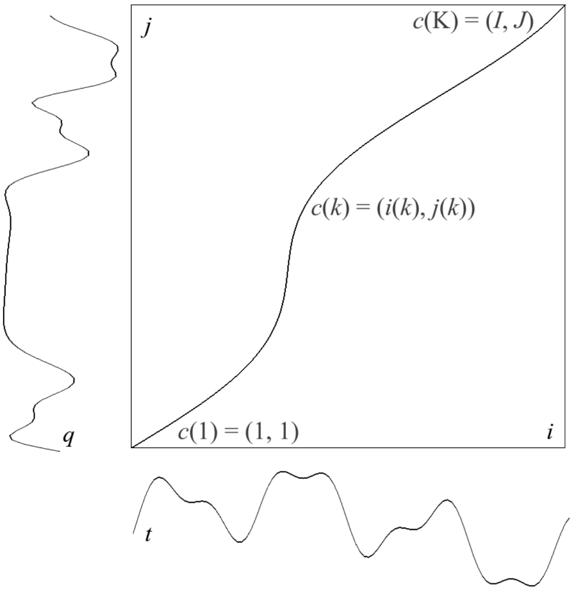

Fig. 3. Warping function.

The DTW algorithm, finding the optimal path in the sense of minimization of Eq. (2), is based on the dynamic programming (DP) principle. The i-j plane is represented as a two-dimensional array g. Every element $g[i, j]$ is assigned a minimal cost of reaching the point (i, j) from the beginning point $c(1) = (1, 1)$:

$$\underset{\substack{i=1,2,...,I \\ j=1,2,...,J}}{\forall} \quad g[i,j] = d(i,j) + \min \begin{cases} g[i, j-1] \\ g[i-1, j-1] \\ g[i-1, j] \end{cases} \tag{4}$$

with the boundary conditions:

$$\begin{aligned} g[0,0] &= 0, \\ g[0,j] &= \infty, \quad \text{for} \quad j = 1, 2, ..., J, \\ g[i,0] &= \infty, \quad \text{for} \quad i = 1, 2, ..., I. \end{aligned} \tag{5}$$

After computing all the values of the array g, the total cost of the optimal path is found in $g[I, J]$. This value is typically multiplied by $(I + J)^{-1}$ to allow comparisons between queries of different lengths.

The DP-equation (4) is a simple variant most often found in literature (Sakurai *et al.*, 2007; Keogh, 2002). Several more sophisticated variants incorporating local slope constraints and weighting coefficients were initially proposed by Sakoe and Chiba (1978). Global constraints in the form of the *Sakoe and Chiba band* (Sakoe, Chiba, 1978) or *Itakura parallelogram* (Itakura, 1975) are also often applied (Fig. 4). The general role of the constraints is to limit the area of the i-j plane under consideration in order to speed up computations and to reduce the risk of "pathological warping" of the sequences. Global constraints also play a fundamental role in efficient indexing techniques introduced by Keogh (2002).

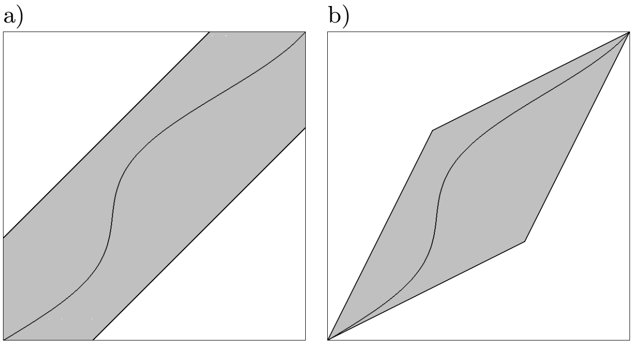

Fig. 4. DTW global constraints: a) Sakoe and Chiba band, b) Itakura parallelogram.

The boundary conditions may be modified to allow for a situation when only a fragment of one sequence is to be matched against the second one. This is generally a subsequence matching problem in which the compared sequences may not start at the same position and/or end at the same position (Sakurai *et al.*, 2007).

2.3. Melody matching

In a typical approach, the query sung by a user is matched against a database consisting of a collection of MIDI files. The templates from the database are converted, similarly to the query, to the form of pitch vectors, expressed in MIDI note numbers rather than as frequency values in Hz. The conversion is straightforward in the case of the MIDI files and always yields unambiguous results. On the other hand, query pitch vectors often need some clean-up to decrease the influence of noise, octave errors, etc., and they generally represent the intended melody only approximately. Many users sing out of tune and they cannot sing with sufficient precision, especially in case of bigger intervals (Yang *et al.*, 2010).

The general problem which is addressed in this work is how to match a melody sung in a different key than in the template. There exist several approaches to deal with this issue. Many researchers use a simple method of subtracting the mean pitch from the whole sequence (Jeon, Ma, 2011). The problem occurs when the melody represented in a query is only a part of the template, or vice versa, in which case subtracting the mean is of no use.

In a different approach the melody may be represented in the form of relative changes of consecutive pitches (differential/delta representation) (Jang *et al.*, 2011). This eliminates the problem but representing raw pitch vectors in this form often yields poor results. In this case the MIDI-based templates consist mostly of zeroes with non-zero values only at the points of note transitions. On the other hand, the note transitions in a query may be spread over several frames, which makes the true comparison impossible.

An effective alternative may be to repeat the matching procedure several times with different transpositions of the query pitch vector. The query may be transposed by, e.g., all possible numbers of semitones within the octave (Yu *et al.*, 2008) or from −5 to +5 semitones in half-of-the-semitone steps (Jang *et al.*, 2011). Various numbers of repetitions may be considered but in any way this is clearly a brute-force approach which increases the computational complexity significantly. Another problem which is still not solved is that the transposition may appear within the query when the user fails to sing an interval (usually a greater one) precisely and continues in a different key.

A solution proposed in this work is to try to follow the melody of the template by gradually decreasing the difference between the query and the template. This is intended to resemble the way in which humans follow the known melody irrespective of pitch inaccuracies and key changes.

3. The proposed algorithm

The input query pitch vector $\mathbf{q}_{\mathrm{raw}}$ is obtained from audio data sampled at 8kHz, with the non-overlapped frame size of 256 samples. It is first preprocessed in order to obtain a smooth melody line without large jumps and unvoiced fragments. The preprocessing includes the following steps:

1. The leading and trailing unvoiced fragments, denoted by the pitch detection algorithm as “0”, are removed.
2. The median of the remaining data is computed and all the pitch values distant from the median by more than a given threshold T_1 are marked as unvoiced, i.e., set to zero. This may help in the case of poor quality of the input data resulting from noise or from errors introduced by the pitch detection algorithm. The quality of the database used in the experiments made this correction necessary in 1% of the queries for $T_1 = 24$ semitones.
3. For the same reason the maximum jump between two consecutive frames can not exceed the threshold T_2. Setting $T_2 = 14$ semitones resulted in 3.8% corrected files.
4. Every unvoiced frame is set to the pitch value of the last voiced frame. In this way one continuous melody is obtained, without any breaks resulting from breathing or articulation. It should be noted that this operation also leads to rejecting some potentially useful information about the rhythm and beat.
5. Median filter of the order of 9 frames is applied to smooth the pitch contour. Preliminary experiments showed that it enhances the recognition results significantly.

The smoothed query pitch vector $\mathbf{q}$ is then compared with all the templates from the database. For every template $\mathbf{t}$ the pitch difference d_{beg} between the beginnings of the query and the template is computed and then subtracted from all the elements of the query pitch vector:

$$\underset{j=1,2,\ldots,J}{\forall} q_j := q_j - d_{\mathrm{beg}}, \tag{6}$$

where J is the length of $\mathbf{q}$ after preprocessing.

This makes both sequences start in the same key. In practice, the value of d_{beg} is computed as:

$$d_{\mathrm{beg}} = \frac{q_2 + q_3}{2} - \frac{t_2 + t_3}{2}. \tag{7}$$

The first pitch value may be unreliable, so it is rejected and the mean of the next two is taken into account.

As the database used for the experiments contained only queries sung from the beginning, this procedure enabled to obtain good matching results with the standard DTW algorithm described in Subsec. 2.2. On the other hand, the queries from the database often ended in arbitrary positions with respect to the template sequences, so using the arithmetic mean computed for *all* the values of $\mathbf{q}$ and $\mathbf{t}$ instead of d_{beg} in (6) yielded poor results.

In a separate set of preliminary experiments the influence of DTW constraints on the recognition results has been tested. It has been found that setting the slope constraint condition $P = 1/2$, as defined by Sakoe and Chiba (1978), yielded the best results.

Having the beginning of the query shifted properly along the frequency axis, one has to deal with transpositions possibly occurring later in the course of the query (Fig. 5). For this purpose the standard DTW procedure is applied first to find the warping function aligning the query and the current template. Going along the path on the i-j plane defined by the warping function, the procedure defined by the block diagram in Fig. 6 is applied. The resulting signal $\widehat{\mathbf{q}}$ is a version of $\mathbf{q}$ modified to follow the pitch values defined by the template $\mathbf{t}$. This process is controlled by the parameter $\alpha \in (0, 1]$. The greater the value of α, the faster will the pitch of the query be aligned with the template. The final value of $\alpha = 0.05$ was used in the experiments.

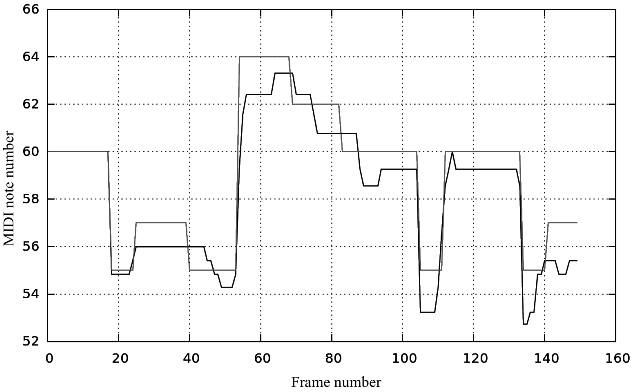

Fig. 5. Example of a transposition (*Old McDonald had a farm*). The first 17 samples of the template (light) and the median-filtered query (dark) are in tune. Most of the remaining part of the query is one-two semitones below the template.

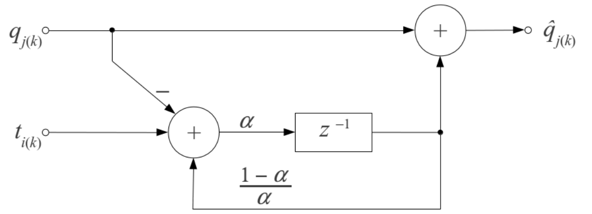

Fig. 6. Block diagram of the tune follower.

The example with the same query and template sequences as in Fig. 5 is shown in Fig. 7. The enhancement introduced by the tune-following procedure is clearly visible. In most places the distance between the sequences decreased and two fragments in the second half of the query got tuned to the template exactly. It should be noted that both Fig. 5, and Fig. 7 present the aligned, i.e., time-warped version of the sequences.

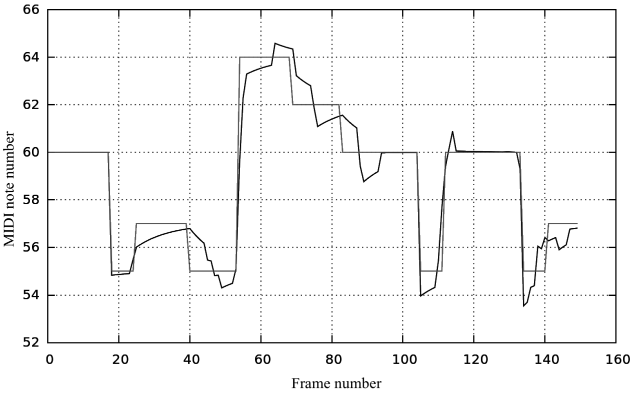

Fig. 7. Result of application of the tune follower ($\alpha = 0.1$).

The final matching cost is then computed for the sequence $\widehat{\mathbf{q}}$ with formula (2). One important thing that should be noted here is that although this cost is lower in comparison with the standard DTW algorithm for the matching template, it can also be lower for the non-matching ones. The fundamental question is whether the proposed tune-following procedure is able to make the matching template win easier in the competition with the others despite the fact that all of them may benefit from its application. In the following section the test results supporting a positive answer to this question will be presented.

It should also be noted that the computational complexity of the presented approach is not significantly increased with respect to the standard DTW algorithm. This results from the fact that adjusting the pitch of the query and computing the updated matching cost is performed as a post-processing step on the already found optimal path. The time complexity of this step is therefore linear, in contrast to the standard DTW which has the complexity of $O(I \cdot J)$. In fact, any method yielding the proper time-warping of the query may be used prior to the tune-following procedure. Hence, many approaches targeted at speeding up the matching process (cf. Subsec. 2.1) may be used instead of the plain DTW algorithm in the context of the system proposed hereby. For example, data dimensionality reduction and application of a lower bounding distance measure (Keogh, 2002; Lau *et al.*, 2005) would allow for efficient database indexing and preselection of a relatively small set of best candidates for further processing. Only those candidates would need the DTW-based matching followed by the tune-following algorithm, thus reducing the computation time, possibly by orders of magnitude.

4. Experimental results

4.1. Database

The publicly available datasets, used in the MIREX 2013 *Query by Singing/Humming* evaluation task (MIREX, 2013), have been chosen to verify the proposed solution. Roger Jang's MIR-QBSH corpus (Jang, 2009) consists of a collection of 48 popular songs (ground-truth MIDI files) to be matched against 4431 queries sung by about 200 subjects. The 48 ground-truth files are mixed with 5274 "noise" files from Essen collection (ESAC-DATA, 2009).

4.2. Testing procedure and results

Each of the 4431 queries was compared with all of the 48 + 5274 template files, one of which was the correct one. For every query **q**, all the templates were ordered by their DTW distance from **q**. According to the rules used in the MIREX evaluation, a search was treated as successful when the correct template was among the top 10 results. The obtained results are presented in Table 1.

Table 1. Total number of recognized queries.

	Top Ten Score	Best Hit Score	δ
DTW	3077 (69.44%)	2109 (47.60%)	0.39532
DTW + Tune Follower	3332 (75.20%)	2455 (55.41%)	0.42975

Additionally, the number of cases when the correct template was the first one on the list of DTW distances was also recorded (the *Best Hit Score* column). The last column displays the mean relative difference between the first and the second file on the list:

$$\delta = \frac{1}{N}\sum_{n=1}^{N}\frac{E_2^{(n)} - E_1^{(n)}}{E_1^{(n)}}, \tag{8}$$

where the sum is computed only over those N queries for which the best hit was the correct one ($N = 2109$ or 2455, respectively). The value $E_p^{(n)}$ denotes the DTW matching cost for the n-th query and the template located at p-th position on the list, i.e., the value $E_1^{(n)}$ represents the score of the template best matching the n-th query (naturally, $E_1^{(n)} < E_2^{(n)}$).

The character of changes introduced by the proposed algorithm may be better assessed on an example of a single query shown in Table 2. The correct template was found to be the closest to the query, both with and without the tune follower (all of the remaining files come from the Essen collection). It may however be observed that the DTW distance of the first template decreased significantly, from 544.56 to 382.80, while the second template remained almost equally distant from the query (659.72 vs. 675.03). Application of the tune follower reordered the list and introduced some changes in the top-ten matching templates (e.g., file Q0095.pv appeared and Q0082.pv was removed).

Table 2. Results for a single query. Year: *2003*, Person: *00011*, File: *00020.pv* (*Happy Birthday*).

No	DTW		DTW + Tune Follower	
	Template	DTW Distance	Template	DTW Distance
1	00020.pv	544.56	00020.pv	382.80
2	V0003F.pv	675.03	V0003F.pv	659.72
3	E0820.pv	731.27	E0820.pv	672.47
4	A0302.pv	752.51	Q0075P.pv	678.65
5	Q0114N.pv	814.22	Q0095.pv	697.24
6	Q0082.pv	825.53	A0302.pv	712.63
7	Q1102J.pv	830.32	Q0114K.pv	712.67
8	Q0080B.pv	840.34	Q0137F.pv	734.59
9	Q0080A.pv	849.02	Q2079J.pv	738.94
10	E0110B.pv	876.25	Q0048C.pv	745.74

4.3. Discussion

The presented results consistently show that the proposed tune-following procedure may have a positive influence on the DTW-based melody search. Although it is true that it generally makes the matching cost smaller for most of the templates, one can expect that this decrease will be more significant in the case of the correct template than for all the non-matching ones (Table 2).

This may result from the effect of accumulation of the corrections for consecutive notes. For example, when the pitch of a note sung by a user is too low with respect to the correct template then it is gradually increased by our procedure until it approaches the right tune, provided that the note is long enough (cf. Fig. 7, frames 24–39). If it is relatively short, it is at least partially corrected (Fig. 7, frames 105–110). In either case, if the note was sung too low, then it is probable that the pitch of the next note will also be too low in which case it will get corrected immediately or – at least – faster. This effect may be observed, e.g., when comparing Fig. 5 and Fig. 7. The pitch discrepancy in frames 105–110 is made significantly smaller due to correction which occurred in the previous frames.

This type of correspondence between the signs of the pitch differences in consecutive notes cannot be generally expected when comparing a query with a non-matching template. Correcting one note may result in increasing the initial difference between the next note and the template. This may even result in increasing the total matching cost, although for long notes and infrequent pitch changes the tune follower will make the query closer to most of the templates.

Further investigation revealed that the exact number of cases when the standard DTW failed to put the correct template on the first place and at the same time the proposed solution managed to do so, was equal to 493. Yet in 147 cases the opposite was true, i.e., the correct template disappeared from the first position when the tune follower was turned on. The analysis of those cases leads to some interesting conclusions which may be used to further improve the results, as demonstrated in the next section.

5. Tuning the tune follower

Melody is a special kind of time series, interpreted according to a rich set of subtle, sometimes very sophisticated semantic rules. As a result, the similarity of melodies can be only approximately assessed on the basis of their general contour. For example, changing the key from major to minor of the – otherwise identical – melody[1], may seem much more important for a human listener than for a DTW-based matching algorithm.

The rhythm and the metric structure of a melody may also be more perceptually important and discriminative than the DTW results would suggest. The lack of correlation of the precise note onset/offset times between the query and a non-matching template may not adequately increase the DTW distance, provided that the general time-warped melody contours are similar. As the proposed tune-following procedure enhances this similarity even further, we may resort to checking whether the note changes in the template coincide with appropriate changes in the query pitch vector (after time alignment). In this way we may increase the influence of the temporal factor on the matching result.

Figure 8 presents a misaligned query with initial DTW distance equal to 746.9 which decreased to 388.29 when the tune follower was turned on. For the true correct template the corresponding values were 485.48 and 454.6, respectively. This is therefore one of those 147 "spoiled" examples in which the proposed method significantly decreased the discrepancy between the query and a non-matching template. It may be observed, however, that the note changes do not correlate in several places (e.g., frames 9–10 or 49–50). This gives rise to the idea of using some kind of a penalty factor to adaptively modify the tune-following procedure in such cases.

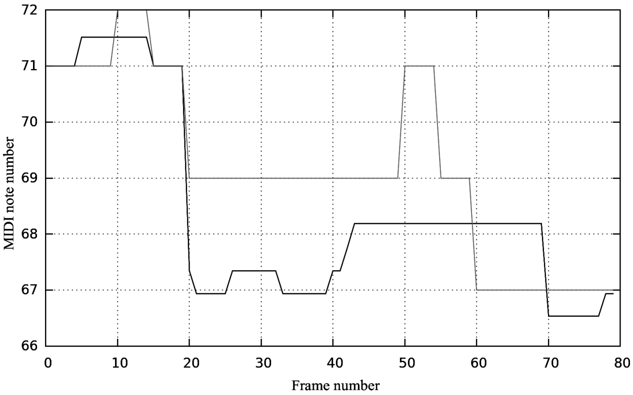

Fig. 8. Query: *Old McDonald* (dark) and a non-matching template: Q3095F.pv (light).

5.1. Adaptive tune following

Two practical issues must be solved – how to measure the correspondence in note changes between the compared melodies and, secondly – how to use this information. As for the first problem, the note changes in the template are analyzed and for every frame i in which the pitch (MIDI note number) changes, the following coefficient is computed:

$$\eta_i = \sum_{j=i}^{i+k-1} q_j - \sum_{j=i-k}^{i-1} q_j, \tag{9}$$

where the parameter k has been set to 3.

Naturally, the coefficient η_i is expected to be positive for the lower-upper note sequence in the template and negative in the other case. The second problem – what to do if these expectations are not met – is solved by adaptively modifying the α parameter of the tune follower. Each time a note transition in the template is not accompanied by an appropriate one in the query, i.e., when:

$$\eta_i(t_i - t_{i-1}) < 0, \tag{10}$$

the α parameter is decreased by a predefined step Δ_{down}, which will effectively make the distance between the following parts of the melodies greater. In the opposite case, when the note transitions correspond to each other, the α parameter is increased by Δ_{up}.

In order to determine the proper values of the Δ_{up} and Δ_{down} parameters, a series of tests has been performed on the database composed of those $493+147 = 640$ queries for which the tune follower had made the difference in the best-hit score. On the basis of the results, presented in Fig. 9, the final values: $\Delta_{\mathrm{up}} = 0.001$ and $\Delta_{\mathrm{down}} = 0.02$ have been taken.

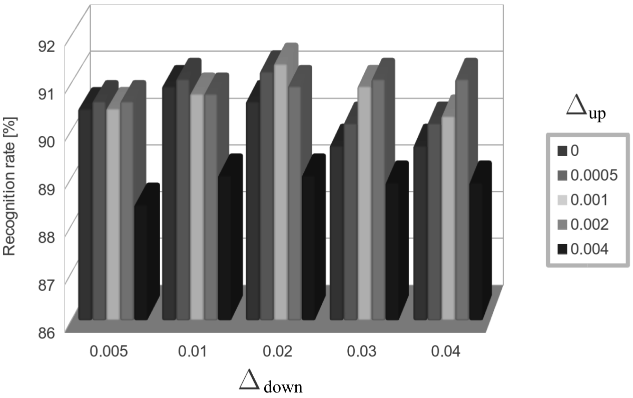

Fig. 9. Influence of the parameters of the adaptive tune follower on the best hit score.

The disproportion in the order of magnitude of those values may be explained quite easily. It seems to result from the fact that the condition (10) holds true infrequently even in non-matching templates, so its relative significance should be greater.

[1] Which typically means that a small fraction of the notes will be shifted by one semitone.

On the other hand, if all note changes match in the compared melodies, the α parameter is increased by Δ_{up} with each note transition. For this reason the initial value of α should be set somewhat lower as compared to the previous experiments (Sec. 4). The tests indicated the best initial $\alpha = 0.03$ and this value has been used to generate the results presented in Fig. 9.

5.2. Subsequence matching

The detailed inspection of the results presented in Sec. 4 revealed also that some templates from the Essen collection (e.g., the one shown in Fig. 8) appeared quite often as the best match for different queries. The apparent reason was their short length, leading to a situation when only the initial part of the query was matched. The database construction (Jang, 2009) allows to assume that the queries are basically shorter than the templates, so they should be matched as a whole.

In the previous experiments (Sec. 4) both asymmetries, i.e., a whole query vs beginning of the template (Fig. 10a) and a whole template vs beginning of the query (Fig. 10b), were allowed. Technically, the lowest cost value has been searched for along both the top and right sides[2] of the DTW cost matrix $g[i, j]$ (Eq. (4)). In the following section the final similarity is computed by analysis of the top row only, which effectively means that the whole query is always matched.

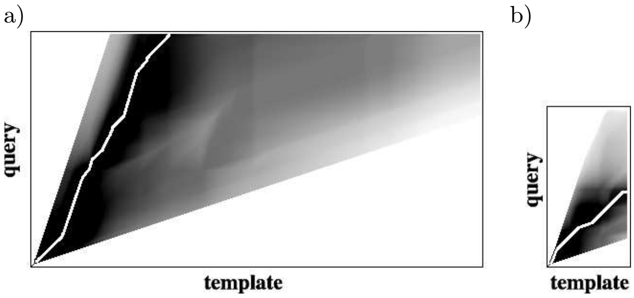

Fig. 10. DTW cost matrices (with optimal paths marked white): a) whole query vs beginning of the template; b) whole template vs beginning of the query.

5.3. Results and discussion

After having established the parameter values and decided on the whole-query approach, we have repeated the tests on the same database (Subsec. 4.1) (Table 3).

Table 3. Total number of recognized queries for the adaptive tune follower.

Top Ten Score	Best Hit Score	δ
3507 (79.15%)	2936 (66.26%)	0.50899

Although rather moderate improvement (4%) may be seen in the top ten recognition score, the best hit score has increased by over 10% (cf. Table 1). This means that the proposed refinement of the tune-following procedure successfully helps to resolve ambiguities within the list of the closest matches. However, only in a relatively small fraction of cases it does help the correct template get to the list, if it had not been there.

The reason for this may lie in the database construction, which contains many very similar melodies or even duplicates[3] in its "noise templates" part (ESAC-DATA, 2009). It may therefore happen that the top ten list for a query is populated by several variants of the same, wrong template, so that the correct one stays aside.

It is worth to note that the proposed adaptive improvement is partially based on the note-based philosophy. Hence, it may be seen as a quite significant modification of the initial DTW approach, going towards hybrid solutions to the QBSH problem.

In general, the proposed tune follower and its adaptive variant enable to efficiently refine the results without computationally complex methods such as repeating the DTW for all possible transpositions (Yu *et al.*, 2008). It should be noted that they can be used independently from efficient indexing techniques (Zhu, Shasha, 2003; Keogh, 2002) or note-based approximate algorithms (Wang *et al.*, 2008) to increase the speed and reliability of a QBSH-based search engine.

6. Conclusion and future works

In this work a modification of the Dynamic Time Warping procedure have been proposed to enhance the results of melody matching in the Query by Humming problem. The modification is inspired by the human ability to match melodies irrespective of the key and pitch inaccuracies. It may be stated that the proposed tune-following procedure plays a similar role for pitch alignment as the DTW does for the case of time alignment and thus it may be seen as a frequency-domain complement to DTW. Similarly, while the DTW decreases the matching cost with respect to the Euclidean distance, the tune-following procedure decreases it even more, with respect to the DTW alone. Although the distance is lower both for the matching and non-matching templates, the presented experimental results clearly demonstrated the superiority of the proposed solution in terms of recognition rate and separation between the matching and non-matching templates.

Additional, significant improvement has been achieved by adaptive modification of the tuning speed

[2] Note that this is a straightforward extension to the case of equal-length sequences where only the value of $g[I, J]$ is considered.

[3] It should be noted that in the MIREX competition a cleaned version of the Essen Database, reduced by over a half of the original size, is used.

on the basis of note events in a template. The proposed enhancement opens a possibility to incorporate the note-based information into the melody matching process without explicit segmentation and conversion into a symbolic representation.

The concept of tune-following will be further investigated in future works. Apart from the issue of parameter settings and possible modifications of the presented procedure itself, it should be noted that it is currently being applied to the already time-aligned sequences, i.e., after the DTW algorithm. It is however possible to integrate the two and modify the pitch adaptively during the dynamic programming optimization of the path cost. This would enable to obtain a different warping function in some cases and, possibly, to match more imprecisely sung queries. However, it seems unclear if this would lead to the overall recognition rate improvement – further research is hence necessary here.

The generalization of the proposed method to subsequence matching problem in which a query does not necessarily start from the beginning of a template would also be of great practical importance. It would eventually enable to construct a flexible hybrid system incorporating several methods, both direct and note-based, that would benefit from the tune-following algorithm to offer enhanced results in a shorter time.

References

1. Adams N., Marquez D., Wakefield G. (2005), *Iterative deepening for melody alignment and retrieval*, [in:] ISMIR 2005, 6th Int. Conf. on Music Information Retrieval, pp. 199–206.
2. Bisesi E., Parncutt R. (2013), *An accent-based approach to automatic rendering of piano performance: Preliminary auditory evaluation*, Archives of Acoustics, **36**, 2, 283–296.
3. Boersma P. (1993), *Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound*, Institute of Phonetic Sciences, University of Amsterdam, Proceedings, **17**, 97–110.
4. Dziubiński M., Kostek B. (2005), *Octave error immune and instantaneous pitch detection algorithm*, Journal of New Music Research, **34**, 3, 273–292.
5. ESAC-DATA (2009), http://www.esac-data.org.
6. Eyben F., Böck S., Schuller B., Graves A. (2010), *Universal onset detection with bidirectional long short-term memory*, [in:] Neural Networks, 11 th International Society for Music Information Retrieval Conference (ISMIR 2010), pp. 589–594.
7. Gerhard D. (2003), *Pitch extraction and fundamental frequency: History and current techniques*, Tech. rep., Dept. of Computer Science, University of Regina.
8. Ghias A., Logan J., Chamberlin D., Smith B.C. (1995), *Query by humming – musical information retrieval in an audio database*, [in:] Proc. of the 3rd ACM Int. Conf. on Multimedia, MULTIMEDIA '95, pp. 231–236.
9. Głaczyński J., Łukasik E. (2011), *Automatic music summarization. A "thumbnail" approach*, Archives of Acoustics, **36**, 2, 297–309.
10. Huang S., Wang L., Hu S., Jiang H., Xu B. (2008), *Query by humming via multiscale transportation distance in random query occurrence context*, [in:] IEEE Int. Conf. on Multimedia and Expo, pp. 1225–1228.
11. Itakura F. (1975), *Minimum prediction residual principle applied to speech recognition*, IEEE Transactions on Acoustics, Speech, and Signal Processing, **23**, 1, 67–72.
12. Jang (2009), http://mirlab.org/dataset/public/mir-qbsh-corpus.rar.
13. Jang D., Song C.J., Shin S., Lee J.S., Park S.J., Jang S.J., Lee S.P., Seo K.H. (2011), *Query by singing/humming system based on the combination of DTW distances for MIREX 2011*, http://www.music-ir.org/mirex/abstracts/2011/JSSLP1.pdf
14. Jang J.S.R., Lee H.R. (2001), *Hierarchical filtering method for content-based music retrieval via acoustic input*, [in:] Proceedings of the ninth ACM International Conference on Multimedia, MULTIMEDIA '01, pp. 401–410.
15. Jeon W., Ma C. (2011), *Efficient search of music pitch contours using wavelet transforms and segmented dynamic time warping*, [in:] IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 2304–2307.
16. Keogh E. (2002), *Exact indexing of dynamic time warping*, [in:] Proceedings of the 28th International Conference on Very Large Data Bases, VLDB '02, pp. 406–417.
17. Lau E., Ding A., Calvin J. (2005), *MusicDB: A query by humming system*, Final project report, Massachusetts Institute of Technology, USA.
18. Lijffijt J., Papapetrou P., Hollmén J., Athitsos V. (2010), *Benchmarking dynamic time warping for music retrieval*, [in:] Proceedings of the 3rd International Conference on Pervasive Technologies Related to Assistive Environments, PETRA '10, pp. 59:1–59:7.
19. Macrae R., Dixon S. (2010), *Accurate real-time windowed time warping*, [in:] Proceedings of the 11th International Society for Music Information Retrieval Conferenc, Downie J.S., Veltkamp R.C. [Eds.], ISMIR 2010, pp. 423–428.
20. McNab R.J., Smith L.A., Witten I.H., Henderson C.L., Cunningham S.J. (1996), *Towards the digital music library: tune retrieval from acoustic input*, [in:] Proceedings of the first ACM International Conference on Digital Libraries, DL '96, pp. 11–18.
21. MIREX (2013), http://www.music-ir.org/mirex/wiki/2013:Main_Page.
22. Sakoe H., Chiba S. (1978), *Dynamic programming algorithm optimization for spoken word recognition*, IEEE Transactions on Acoustics, Speech, and Signal Processing, **26**, 1, 43–49.

23. Sakurai Y., Faloutsos C., Yamamuro M. (2007), *Stream monitoring under the time warping distance*, Research showcase, Carnegie Mellon University, URL http://repository.cmu.edu/compsci/529.

24. Salamon J., Gómez E. (2012), *Melody extraction from polyphonic music signals using pitch contour characteristics*, IEEE Transactions on Audio, Speech, and Language Processing, **20**, 6, 1759–1770.

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

26. Typke R., Wiering F., Veltkamp R.C. (2007), *Transportation distances and human perception of melodic similarity*, Musicae Scientiae, pp. 153–181.

27. Uitdenbogerd A., Zobel J. (1999), *Melodic matching techniques for large music databases*, [in:] Proceedings of the seventh ACM International Conference on Multimedia (Part 1), MULTIMEDIA '99, pp. 57–66.

28. Wang L., Huang S., Hu S., Laing J., Xu B. (2008), *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, pp. 471–475.

29. Wolkowicz J., Kulka Z., Keselj V. (2008), *N-gram based approach to composer recognition*, Archives of Acoustics, **33**, 1, 43–55.

30. Yang J., Liu J., Zhang W. (2010), *A fast query by humming system based on notes*, [in:] INTERSPEECH, pp. 2898–2901.

31. Yu H.M., Tsai W.H., Wang H.M. (2008), *A query-by-singing system for retrieving karaoke music*, IEEE Transactions on Multimedia, **10**, 8, 1626–1637.

32. Zhu Y., Shasha D. (2003), *Warping indexes with envelope transforms for query by humming*, [in:] Proceedings of the 2003 ACM SIGMOD International Conference on Management of Data, SIGMOD '03, pp. 181–192.