DRIFTIN' DOWN THE SCALE: DYNAMIC TIME WARPING IN THE PRESENCE OF PITCH DRIFT AND TRANSPOSITIONS

Simon Waloschek, Aristotelis Hadjakos
Center of Music and Film Informatics, Detmold University of Music, Germany
{s.waloschek, a.hadjakos}@cemfi.de

ABSTRACT

Recordings of a cappella music often exhibit significant pitch drift. This drift may accumulate over time to a total transposition of several semitones, which renders the canonical 2-dimensional *Dynamic Time Warping* (DTW) useless. We propose *Transposition-Aware Dynamic Time Warping* (TA-DTW), an approach that introduces a 3rd dimension to DTW. Steps in this dimension represent changes in transposition. Paired with suitable input features, TA-DTW computes an optimal alignment path between a symbolic score and a corresponding audio recording in the presence of pitch drift or arbitrary transpositions.

1. INTRODUCTION

Existing audio-to-score alignment systems based on DTW are not yet able to handle performances appropriately if they exhibit significant pitch drift. However, pitch drift is rather common in a cappella singing and choir performances due to accumulation of intonation inaccuracies over time.

Absolute pitch, which is the ability to recognize and produce a given pitch without an external reference, is a rare ability of about 0.01% of the population [23]. Therefore, most singers have to rely on a combination of referencing with previous and simultaneous pitches together with muscle memory to intonate appropriately. In solo singing, the accuracy of a good singer has been reported to range from about 13 cents (standard deviation) [24] to about 22 cents [27] for very short melodies and intervals. Expert listeners judge a deviation of 20-25 cents to be still in tune [27]. The accuracy of note production can however be influenced adversely by various factors, e.g. by an unbalanced ratio of the sound pressure level of the reference sound in relation to the feedback sound of the singer's own voice [25]. Also the presence of vibrato, the absence of common partials between the voices and the absence of high partials make it more difficult to intonate correctly [24].

Due to the lack of an absolute reference, these minor deviations can lead to relatively large pitch deviations in the long run. It is widely reported that choirs have significant pitch drift, see [1]. Seaton et al. [20] surveyed amateur and professional choir singers and conductors regarding their experiences with pitch drift. Nearly half of the participants report that pitch drift occurs regularly while only 14% report that drift happens rarely or not at all. Nearly 80% of the participants say that the direction of the drift is usually downward while almost all other participants say that drift occurs in either direction similarly often.

However, pitch drift is not just an addition of small inaccuracies: Howard argues that pitch drift is almost inevitable when singing unaccompanied music that modulates from one key to another [12]. This arises mathematically from the observation that singers use non-tempered intonation based on the ratios of small integer numbers. Howard's measurements provide evidence that singers in fact use non-tempered intonation and that they consequentially shift their intonation as hypothesized. He even argues "that conductors who have a desire to correct overall intonation drift for its own sake in an a cappella performance [...] may be misguided" [12] if the piece contains considerable modulation.

This paper contributes a novel method called *Transposition-Aware Dynamic Time Warping* (TA-DTW) aiming at making an alignment between a symbolic score and a corresponding audio recording. TA-DTW is able to handle pitch drift (in contrast to a constant transposition), which makes it particularly useful to synchronize choir and singing recordings. Furthermore, it may be used as a drop-in replacement for existing solutions that can handle "only" fixed transpositions, which are commonly encountered in transcriptions of a piece for another instrument and historically informed performance practice.

The structure of this paper as follows: Section 2 describes the feature design, derived from Harmonic Pitch Class Profiles. Section 3 introduces TA-DTW as a 3-dimensional DTW based on the aforementioned features, followed by an evaluation, conclusions and future work in Sections 4 and 5. Related work is discussed in comparison to our approach within the individual sections.

2. ROBUST PITCH CLASS FEATURES IN THE PRESENCE OF PITCH DRIFT

Audio-to-Score Alignment algorithms that are based on DTW generally use pitch features such as chroma features [13, 15] as an intermediate data representation format

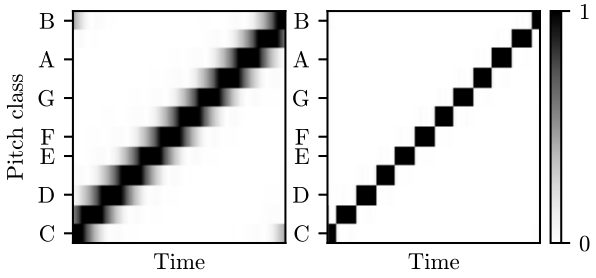

Figure 1. Chromagrams showing logarithmic sine sweeps from C4 to B4. *Left:* Canonical CQT chromas. *Right:* HPCPs with frame-wise tuning frequency estimation as used in this paper.

between the symbolic score and corresponding audio data. Chroma features are 12-dimensional vectors that describe, to a certain extent, tonality of a specific and usually very short sequence of music. They are obtained by measuring the relative intensity of each of the 12 pitch classes (C, C♯, D, ..., B) of the equal-tempered scale within an analysis frame. While this undoubtedly reduces the informational content in relation to tonal characteristics, this very reduction makes chroma features robust to changes in instrumentation and timbre. They still capture melodic and harmonic characteristics of music and thus provide a useful abstraction for various tasks within the music information retrieval research area.

For symbolic scores chroma features can be computed directly by mapping the pitch of the individual notes to their corresponding element in the chroma vector [13]. In case of audio data they are mostly computed using the *Fast Fourier Transform* (FFT) or the specialized *Constant Q Transform* (CQT) [3]. The latter is especially useful for western music since it allows for 1-to-1 mapping of filter bins to MIDI pitches. CQT can be expressed as a filter bank with fixed center frequencies for all filters, defined by a given reference pitch. In the presence of pitch drift, however, this reference pitch and thus the filter center frequencies have to be dynamically adapted to get high selectivity between adjacent semitones. Figure 1 (left) shows the effect of fixed center frequencies for a continuous sine sweep: The resulting chromagram appears to be "fuzzy" with leakage between semitones. For music that is more complex inaccurate center frequencies can lead to practically unusable chroma features.

2.1 Tuning Frequency Estimation

One way of dealing with these inaccuracies is the usage of multiple filter banks with slightly diverging reference pitch [18] and picking the best fit for each frame. A more general solution, however, is the use of tuning frequency estimation. Gnann et al. [8] proposed a real-time estimation algorithm, specifically addressing pitch drift in choir music. While their method of reducing the quadratic tuning deviation serves the purpose of having an active "pitch drift warning system" for rehearsals quite well, it does not allow for a time resolution down to a single analysis frame. The same problem arises in an approach by Dressler [7] based on circular statistics: These methods calculate the tuning frequency iteratively resulting in an initial delay. Such behavior is unfavorable for DTW algorithms that rely on greatest feature accuracy possible in each frame. Both tuning estimation approaches were evaluated by Degani et al. [6] together with a third option that utilizes *Harmonic Pitch Class Profiles* (HPCP) [9, 29] and allows for the calculation of the deviation from reference pitch within a single analysis frame [11]. As all three tuning estimation methods demonstrated similar performance, we will focus on HPCPs and their superior time resolution.

HPCPs are closely related to chroma features but differ in one important aspect: They are tuning independent by definition, so that the reference frequency is not explicitly defined. The result of HPCP computation is an octave-independent histogram with 12, 24, 36, or even more bins, depending on the needed frequency resolution as shown in Figure 2. For a constant quality spectrum C with N bins in total and 36 per octave, the value of the k-th bin of a 36-bin HPCP H is calculated by

$$H_k = \sum_{n=0}^{N/36} |C_{k+36n}| \quad \forall k \in [1:36]. \tag{1}$$

In order to estimate the tuning deviation, each HPCP frame is processed with a peak detection algorithm. Multiple peaks might be found in such a frame and the global deviation from an assumed reference pitch can be averaged over the individual deviations of the peaks.

In this paper we decided to use quadratic interpolation as described by Smith in [22] with 36-bin HPCPs. However, we do not look for the peaks explicitly but rather accumulate the magnitudes of each semitone's 3 corresponding bins:

$$m_k = \sum_{n=0}^{11} H_{3n+k} \quad \forall k \in \{1, 2, 3\}. \tag{2}$$

We assume that m_2 is the highest of these values, otherwise we would have shifted the values of m cyclically by a value $s \in \{-1, 0, 1\}$. To be consistent with [22] and increase readability, we define $\alpha = m_1$, $\beta = m_2$, and $\gamma = m_3$. Next, we fit a parabola to these magnitudes, i.e. through

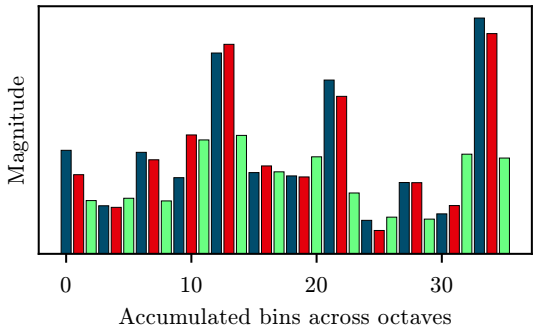

Figure 2. Harmonic Pitch Class Profile with 36 bins per octave for a single analysis frame.

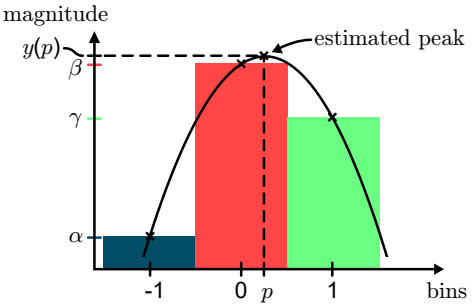

Figure 3. Parabola that is fitted to the bins' magnitudes α, β, and γ.

$(-1, \alpha)$, $(0, \beta)$, and $(1, \gamma)$ as shown in Figure 3. (The bins have been arbitrarily renumbered about the estimated peak that is represented by the parabola's vertex.)

Looking at the general formula for a parabola

$$y(x) = a(x - p)^2 + b \tag{3}$$

we can directly tell the location of the vertex: The center point p gives us the error offset of our actual pitch in bins, while the amplitude $y(p) = b$ equals the peak amplitude. All three magnitudes can be calculated as follows:

$$\begin{aligned} \alpha &= ap^2 + 2ap + a + b, \\ \beta &= ap^2 + b, \\ \gamma &= ap^2 - 2ap + a + b \end{aligned} \tag{4}$$

The peak location in (fractional) bins is given by

$$p = \frac{1}{2}\frac{\alpha - \gamma}{\alpha - 2\beta + \gamma} \in \left[-\frac{1}{2}, \frac{1}{2}\right] \tag{5}$$

and the estimated peak magnitude gets calculated by:

$$y(p) = \beta - \frac{1}{4}(\alpha - \gamma)p. \tag{6}$$

2.2 Feature Computation

Knowing the global tuning deviation p, we can calculate the estimated peak magnitude for every pitch class (from its 3 bins) of our HPCP H via Equation 6 with s being the potential cyclic shifting done after Equation 2 and

$$\begin{aligned} \alpha &= H_{3k-2-s} \\ \beta &= H_{3k-1-s} \quad \forall k \in [1:12]. \\ \gamma &= H_{3k-s} \end{aligned} \tag{7}$$

This step effectively reduces the 36 bins per octave to 12 bins per octave, which makes the resulting HPCPs comparable to standard chroma features. To decrease differences in dynamics between the features, each HPCP vector is normalized to have length 1. We obtain a chromagram as exemplary shown in Figure 1 (right) by repeating this for the entire audio recording.

If the estimated tuning is off by approximately -0.5 or $+0.5$ bins and the actual tuning of the recording fluctuates, the resulting features can be off by 1 semitone in either direction. This can be considered a local "unintended transposition" and will be handled in the next section.

3. TRANSPOSITION-AWARE DYNAMIC TIME WARPING

Support for changing transpositions over time is very limited in current alignment systems. In this context the term *transposition* covers intended alterations in pitch, e.g. "baroque pitch" or simply singing in a different key, as wells as unintended pitch drift that exceeds the scope of a semitone. Most alignment systems, such as *Antescofo* [4] pass the problem of unknown transpositions on to the user and force them to adapt their symbolic score accordingly by themselves. Niedermayer [19] solves this step by computing all possible transpositions and picking the best fit. This is similar to a work by Müller [16] that determines the best transpositions for each individual pitch feature, though the results are not used for audio-to-score alignment. All these systems assume that the global tuning does not change over time and has to be estimated only once at the beginning, except Arzt [2]: He uses fingerprinting to determine a musical piece, the current position within that piece, and its transposition before the actual alignment. Hence, the system is theoretically able to "recover" from unforeseen pitch changes after some time but is (for now) restricted to piano music and does not allow for continuous alignment in such cases.

We propose an extended version of DTW called *Transposition-Aware Dynamic Time Warping* (TA-DTW) that allows for continuous changes in transposition during alignment. It shares conceptual ideas with the multidimensional DTW presented in [28] but focuses on special properties of chroma features and the nature of musical transpositions. The remainder of this section presupposes basic knowledge of the original DTW algorithm. A comprehensive overview can be found in [15].

3.1 Distance Calculation & Transpositions

In order to compute the alignment we need the distances between the vectors from the score and the audio HPCP vectors as computed in Section 2. Various metrics have been used to calculate these distances such as the Euclidean distance (2-norm distance) [13, 15] and Manhattan distance (1-norm distance) [2, 15]. For computational complexity reasons, however, we opted for the *Cosine Distance* [1] which is defined for two nonzero vectors $\mathbf{a}$ and $\mathbf{b}$ as

$$c(\mathbf{a}, \mathbf{b}) = 1 - \cos(\theta) = 1 - \frac{\mathbf{a} \cdot \mathbf{b}}{\|\mathbf{a}\|_2 \cdot \|\mathbf{b}\|_2} \tag{8}$$

and represents the angular distance ranging from 0 (equal orientation) to 2 (diametrically opposed). Taking into account that our HPCP features already have the length 1, the denominator can be reduced to 1 and leaves us with

$$c(\mathbf{a}, \mathbf{b}) = 1 - \mathbf{a} \cdot \mathbf{b} = 1 - \sum_{i=1}^{n} a_i \cdot b_i. \tag{9}$$

This operation can be extended to two matrices with the same number of rows and unit length columns. It results in

[1] As the cosine distance does not obey the triangle inequality it is strictly speaking not a proper distance metric, see [21]. Nevertheless, it can be used as such in this particular context.

a matrix that contains distances between all combinations of column vectors of these matrices ($\mathbb{1}$ denotes the $N \times M$ matrix of ones):

$$c(\mathbf{A}, \mathbf{B}) = \mathbb{1} - \mathbf{A}^{\mathrm{T}}\mathbf{B} \tag{10}$$

To make use of this in our context we will represent our sequence of HPCP vectors $\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_N$ from the score as a matrix [2]:

$$\mathbf{A} = \begin{bmatrix} a_{1,1} & a_{1,2} & \ldots & a_{1,N} \\ a_{2,1} & a_{2,2} & \ldots & a_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{12,1} & a_{12,2} & \ldots & a_{12,N} \end{bmatrix} \tag{11}$$

The same applies to the sequence of HPCP vectors $\mathbf{B} = [\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_M]$ from the audio recording.

Cyclically shifting the elements of a pitch class vector by a value t equals transposing by t semitones as pointed out by Goto [10]. We make use of this property and compute all 11 possible transpositions $t \in [1 : 11]$ for the score HPCP matrix $\mathbf{A}$. Shifting the rows of $\mathbf{A}$ can be done by multiplying the cyclic permutation matrix $\mathbf{P} \in \mathbb{R}^{12\times 12}$ with $\mathbf{A}$:

$$\mathbf{A}_t = (\mathbf{P}^t)\mathbf{A}\,, \mathbf{P} = \begin{bmatrix} 0 & 0 & \ldots & 0 & 1 \\ 1 & 0 & \ldots & 0 & 0 \\ 0 & 1 & \ldots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \ldots & 1 & 0 \end{bmatrix} \tag{12}$$

Finally, we calculate all distances for all transpositions by

$$c(\mathbf{A}, \mathbf{B}, t) = \mathbb{1} - (\mathbf{A}_t)^{\mathrm{T}}\mathbf{B}. \tag{13}$$

3.2 Accumulated Multidimensional Cost Matrix

Throughout this section we will express the results of Equation 13 as the cost matrix $\mathbf{C} \in \mathbb{R}^{N\times M\times 12}$.

With $\mathbf{C}$ we can compute the accumulated cost matrix $\mathbf{D} \in \mathbb{R}^{N\times M\times 12}$ by means of dynamic programming. Additionally to steps in the $n \times m$ plane, as performed in the canonical DTW, steps in the transposition dimension t need to be considered. Moving a semitone cyclically downwards and upwards along the $t \in [0 : 11]$ axis will be defined as

$$\begin{aligned} t_- &= (t-1) \bmod 12 \\ t_+ &= (t+1) \bmod 12 \end{aligned} \tag{14}$$

which allows arbitrary transposition changes that may even exceed one octave. We will restrict the possible transposition changes between two adjacent analysis windows to one semitone in order to keep the underlying math concise in this paper. While this seems reasonable in practice, too, it is not an inherent restriction of the algorithm. Figure 4 visualizes the resulting valid steps inside the accumulated cost matrix $\mathbf{D}$ for a possible alignment path.

[2] Music analysis frameworks such as *librosa* or *madmom* already use such a representation anyway.

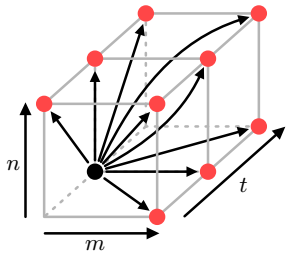

Figure 4. Valid steps inside the accumulated multidimensional cost matrix $\mathbf{D}$.

The accumulated multidimensional cost matrix will be calculated as follows:

$$\mathbf{D}_{n,m,t} = \begin{cases} \sum_{k=1}^{m} \mathbf{C}_{1,k,t} & \text{if } n = 1 \\ \sum_{k=1}^{n} \mathbf{C}_{k,1,t} & \text{if } m = 1 \\ \min(steps) + & \\ w(\Delta t)\mathbf{C}_{n,m,t} & \text{otherwise} \end{cases} \tag{15}$$

The (recursive) $steps$ for computing $\mathbf{D}$ are defined as

$$steps = \left\{ \begin{matrix} \mathbf{D}_{n,} & m-1, & t \\ \mathbf{D}_{n-1,} & m, & t \\ \mathbf{D}_{n-1,} & m-1, & t \\ \mathbf{D}_{n,} & m-1, & t_- \\ \mathbf{D}_{n-1,} & m, & t_- \\ \mathbf{D}_{n-1,} & m-1, & t_- \\ \mathbf{D}_{n,} & m-1, & t_+ \\ \mathbf{D}_{n-1,} & m, & t_+ \\ \mathbf{D}_{n-1,} & m-1, & t_+ \end{matrix} \right\}. \tag{16}$$

Steps along the t-axis alone are not allowed since it is impossible to calculate $\mathbf{D}$ under these conditions. The factor $w(\Delta t)$ is a weighting factor for penalizing relative movements along the t axis. An increased weight has shown to stabilize the algorithm by reducing accidental transposition changes for special cases, e.g. monophonic passages, where regular pitch changes might look equivalent to transpositions.

3.3 Backtracking

In order to compute the warping path $\mathbf{p}$ we use a backtracking algorithm. The starting point $\mathbf{p}_L$ for the recursive computation is the point along the (N, M, t)-axis in $\mathbf{D}$ with the least costs:

$$\begin{aligned} T &= \arg\min_t \left(\mathbf{D}_{N,M,t}\right) \\ \mathbf{p}_L &= (N, M, T)\,. \end{aligned} \tag{17}$$

Feature	Algorithm	Drift	Percentage of events with absolute misalignment error						
			≤0.15s	≤0.20s	≤0.25s	≤0.30s	≤0.40s	≤0.50s	≤1.00s
Chroma	DTW		83.46%	87.75%	89.72%	90.87%	92.04%	92.67%	93.99%
HPCP	DTW		86.28%	91.09%	93.23%	94.39%	95.75%	96.36%	97.64%
HPCP	TA-DTW		**86.52%**	**91.69%**	**93.96%**	**95.18%**	**96.40%**	**96.92%**	**97.89%**
Chroma	DTW	✓	20.51%	23.00%	24.61%	25.98%	28.35%	30.23%	36.53%
HPCP	TA-DTW	✓	**79.89%**	**88.35%**	**92.09%**	**93.97%**	**95.56%**	**96.28%**	**97.31%**

Table 1. Results of the audio-to-score alignment evaluation. Best results are emphasized.

Now we move recursively through the accumulated cost matrix:

$$\mathbf{p}_{\ell-1} = \begin{cases} \arg\min \begin{pmatrix} \mathbf{D}_{1,m-1,t} \\ \mathbf{D}_{1,m-1,t_-} \\ \mathbf{D}_{1,m-1,t_+} \end{pmatrix} & \text{if } n = 1 \text{ and } m \geq 2 \\ \arg\min \begin{pmatrix} \mathbf{D}_{n-1,1,t} \\ \mathbf{D}_{n-1,1,t_-} \\ \mathbf{D}_{n-1,1,t_+} \end{pmatrix} & \text{if } m = 1 \text{ and } n \geq 2 \\ \arg\min(steps) & \text{if } n, m \geq 2 \end{cases} \tag{18}$$

The resulting 3-dimensional warping path can be orthogonally projected onto different planes as shown in Figure 5:

$n \times m$ corresponds to the final alignment between the input feature matrices $\mathbf{A}$ and $\mathbf{B}$.

$m \times t$ gives information about the location of transposition changes in the audio data in relation to the score. The "local unintended transpositions" as outlined in Section 2.2 are clearly visible.

$n \times t$ shows accordingly these transposition change positions in the score.

The transposition changes in the planes $n \times t$ and $m \times t$ can be refined by adding the center frequency offsets p as calculated in Section 2 for each audio HPCP vector. This allows for computing continuous pitch drift data.

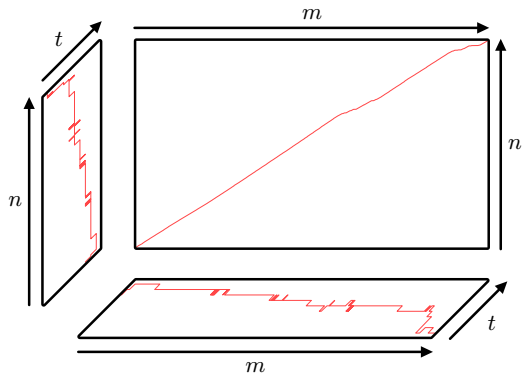

Figure 5. Orthogonal projections of the 3-dimensional warping path onto the $n \times t$, $m \times t$, and $n \times m$ plane.

4. EVALUATION

We evaluated the alignment accuracy of our TA-DTW with HPCPs in comparison with the canonical DTW and plain CQT-based chroma features. Due to the lack of datasets with choral music and corresponding beat-level annotations, we generated the evaluation data from the complete MusicNet dataset [26]. It consists of almost 1.3 million note events (that were manually verified by expert annotators) for approximately 34 hours of chamber music performances with various instrumentations. All material is available in 44.1 kHz sampling rate. Although a cappella music is not part of the dataset, we considered it meaningful for evaluation due to its substantial scope. Since the pieces of the dataset present no significant pitch drift, we extracted all recordings to raw PCM files and introduced continuously changing random artificial pitch drift.

This was done by loading each of the 330 pieces into a *Digital Audio Workstation* (DAW) and generating 100 equidistantly distributed random pitch change markers along the time axis for each piece. The amount of introduced pitch drift was kept within the range of ± 4 semitones and followed brownian motion to introduce correlation with previous markers. Between the markers, the pitch deviation was linearly interpolated. A randomization as such is a reasonably realistic simulation according to the pitch drift model for a cappella music of Mauch et al. in [14].

Based on the evaluation methodology of Cont et al. in [5] for audio-to-score alignment, the absolute alignment errors in seconds for note onsets were calculated. 1024 samples per window and no overlap for the audio data were used. This equals a feature rate of ~ 43 vectors per second or a window length of approximately 23ms. The results are shown in Table 1. We found $w(\Delta t) = 6.5$ to be a suitable penalty factor for changes along the t-axis in $\mathbf{D}$, see Equation 15.

In the absence of any pitch drift, we found that using HPCPs showed superior performance in contrast to plain chroma features, regardless of whether the calculation of the alignment was done using DTW or our proposed TA-DTW. This can be explained by slight deviations in tuning frequency of the recordings that are compensated in the computation of HPCPs. Using them as calculated in this paper introduces occasional errors in the form of "local transpositions" as explained in Section 2.2. Such errors can be minimized by switching from DTW to TA-DTW, which further improves the alignment.

For music with drifting pitch our proposed method shows comparable results while the "classic" approach failed for the majority of note onsets. We assumed that the remaining $\sim 30\%$ are based on the limited maximum pitch drift in our test data, resulting in this amount of data effectively being not or only marginally pitched. This hypothesis was validated by exemplarily computing the alignment with plain DTW using audio files that exhibit constant pitch drift >1 semitone. These cases showed results $<1\%$ for all shown time intervals.

The drawbacks of this approach are the increased memory and computation requirements. TA-DTW requires a cost matrix of dimension $N \times M \times 12$, which is 12 times larger compared to the 2-dimensional cost matrix for DTW. Computing the warping path with TA-DTW involves computing $N \times M \times 12 \times 9$ path scores. This is 36 times greater in contrast to $M \times N \times 3$ in DTW. We have empirically verified these numbers. For music recordings with a length that exceeds several minutes, the algorithm demands well-equipped hardware in terms of memory.

5. CONCLUSIONS & FUTURE WORK

In this paper we presented a DTW-based method to compute audio-to-score alignment for audio data that suffers from drift in global pitch throughout the recording. To this end we explained the computation of suitable pitch features that allow for "sharper" distinction between adjacent notes on the pitch scale. We used these features in conjunction with a DTW algorithm that we extended to support static and dynamically changing transpositions. A final evaluation proved the robustness and effectiveness of our approach.

Apart from normalizing our pitch features to have length 1, we did not process them any further. This ensures that they can be used as basis for additional feature enhancements [17]. Similarly, this applies to our DTW extension: Since the *Transposition-Aware DTW* is conceptually very close to the original DTW algorithm, many variations and improvements such as varying step size conditions, local weights, or global constraints [15] can be adapted easily.

Potential on-line variants of TA-DTW could greatly reduce the computational complexity by only calculating cost values for $t \pm 1$ for the current audio window.

6. REFERENCES

[1] Per-Gunnar Alldahl. *Choral intonation*. Gehrmans Musikförlag, 2008.

[2] Andreas Arzt. *Flexible and Robust Music Tracking*. PhD thesis, Johannes Kepler University, Linz, Austria, 2016.

[3] Judith C Brown. Calculation of a constant Q spectral transform. *The Journal of the Acoustical Society of America*, 89(1):425–434, 1991.

[4] Arshia Cont. Antescofo: Anticipatory synchronization and control of interactive parameters in computer music. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 33–40, 2008.

[5] Arshia Cont, Diemo Schwarz, Norbert Schnell, and Christopher Raphael. Evaluation of real-time audio-to-score alignment. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, 2007.

[6] Alessio Degani, Marco Dalai, Riccardo Leonardi, and Pierangelo Migliorati. Comparison of tuning frequency estimation methods. *Multimedia Tools and Applications*, 74(15):5917–5934, 2015.

[7] Karin Dressler and Sebastian Streich. Tuning frequency estimation using circular statistics. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, 2007.

[8] Volker Gnann, Markus Kitza, Julian Becker, and Martin Spiertz. Least-squares local tuning frequency estimation for choir music. In *Audio Engineering Society Convention 131*. Audio Engineering Society, 2011.

[9] Emilia Gómez. *Tonal description of music audio signals*. PhD thesis, Universitat Pompeu Fabra, Barcelona, Spain, 2006.

[10] Masataka Goto. A chorus section detection method for musical audio signals and its application to a music listening station. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1783–1794, 2006.

[11] Christopher Harte and Mark Sandler. Automatic chord recognition using quantised chroma and harmonic change segmentation. *Centre for Digital Music, Queen Mary University of London*, 2009.

[12] David M Howard. Intonation drift in a capella soprano, alto, tenor, bass quartet singing with key modulation. *Journal of Voice*, 21(3):300–315, 2007.

[13] Ning Hu, Roger B Dannenberg, and George Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 185–188, 2003.

[14] Matthias Mauch, Klaus Frieler, and Simon Dixon. Intonation in unaccompanied singing: Accuracy, drift, and a model of reference pitch memory. *The Journal of the Acoustical Society of America*, 136(1):401–411, 2014.

[15] Meinard Müller. *Information retrieval for music and motion*, volume 2. Springer, 2007.

[16] Meinard Müller and Michael Clausen. Transposition-invariant self-similarity matrices. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, 2007.

[17] Meinard Müller and Sebastian Ewert. Chroma toolbox: Matlab implementations for extracting variants of chroma-based audio features. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, 2011.

[18] Meinard Müller, Peter Grosche, and Frans Wiering. Automated analysis of performance variations in folk song recordings. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, 2010.

[19] Bernhard Niedermayer. *Accurate audio-to-score alignment: data acquisition in the context of computational musicology*. PhD thesis, Johannes Kepler University, Linz, Austria, 2012.

[20] Richard Seaton, Dennis Pim, and David Sharp. Pitch drift in a cappella choral singing – work in progress. In *Proceedings of the Institute for Acoustics Annual Spring Conference*, volume 35, 2013.

[21] John R Smith. *Integrated spatial and feature image systems: Retrieval, analysis and compression*. PhD thesis, Columbia University, New York, USA, 1997.

[22] Julius O. Smith. *Spectral audio signal processing*. W3K publishing, 2011.

[23] Annie H Takeuchi and Stewart H Hulse. Absolute pitch. *Psychological bulletin*, 113(2):345–361, 1993.

[24] Sten Ternström and Johan Sundberg. Acoustical factors related to pitch precision in choir singing. *Speech Transmission Laboratory Quarterly Progress and Status Report (STL-QPSR*, 2(3):1982, 1982.

[25] Sten Ternström and Johan Sundberg. Intonation precision of choir singers. *The Journal of the Acoustical Society of America*, 84(1):59–69, 1988.

[26] John Thickstun, Zaid Harchaoui, and Sham Kakade. Learning features of music from scratch. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

[27] Allan Vurma and Jaan Ross. Production and perception of musical intervals. *Music Perception: An Interdisciplinary Journal*, 23(4):331–344, 2006.

[28] Martin Wöllmer, Marc Al-Hames, Florian Eyben, Björn Schuller, and Gerhard Rigoll. A multidimensional dynamic time warping algorithm for efficient multimodal fusion of asynchronous data streams. *Neurocomputing*, 73(1-3):366–380, 2009.

[29] Yongwei Zhu, Mohan S Kankanhalli, and Sheng Gao. Music key detection for musical audio. In *Proceedings of the 11th International Multimedia Modelling Conference (MMM)*, 2005.