Vocal onset detection and pitch segmentation in medieval choral music guided by original notational sources

Samuel D. Bellows*†, Sarabeth S. Mullins*‡, and Brian F.G. Katz*
* Sorbonne Université, CNRS UMR 7190, France
† University of Utah Asia Campus, Department of Electrical and Computer Engineering, Republic of Korea
‡ Treble Technologies, Iceland
E-mail: samuel.bellows11@gmail.com, sarabeth.mullins@posteo.co

***Abstract*—Accurate vocal onset detection is essential to many music information retrieval and music performance analysis tasks. While most algorithms and approaches perform favorably on contemporary popular instrumental music, precise vocal onset detection remains an elusive challenge, especially when expanded to include genres outside the typical applications. In particular, onset detection of medieval choral music faces further obstacles due to melismatic passages, variable expressing timing, fast-time and slow-time pitch variations, and an abstract notational system. To overcome these challenges, this work proposes a jumping dynamic time warping classifier to segment recorded audio frames to their respective plainchant neume. The approach uses an iterative maximum a posteriori pitch estimate to dynamically adjust neume pitch class from the initial scored values. The proposed algorithm shows significant improvements in vocal onset detection compared to standard and state-of-the-art audio-only and audio-alignment approaches.**

I. Introduction

Note onset detection is a foundational task in music information retrieval (MIR) and musical performance analysis (MPA) research. Within MIR, precise onset detection is required for automatic transcription, beat tracking, and melodic segmentation, whereas in MPA, it enables the analysis of ensemble timing deviations, expressive timing modeling, and pitch and articulation studies. Nonetheless, despite the importance of onset detection for the successful execution of MIR and MPA tasks, robust algorithms for use in vocal performance analysis, particularly for non-Eurogeneric music, remain limited.

A variety of algorithms have been proposed and implemented to determine precise note onsets in recorded music, typically falling into *audio-only* or *audio-alignment* based approaches. Audio-only, or blind onset detection algorithms, mark an audio frame as an onset when an indicator function based on one or more acoustic features surpasses a given threshold [1]. Audio-alignment approaches use metadata contained in MIDI scores, lyric sheets, rhythmic structures, or ground-truth performances to detect note onsets within an audio file using, for example, dynamic time warping (DTW) or hidden Markov models (HMM) [2]–[4]. For the purpose of MPA, a key advantage of audio-alignment techniques is that each note in a score will always be associated with a unique onset time, simplifying cross-performance analysis.

Audio-only and audio-alignment approaches both require human guidance in setting optimal algorithm parameters or preparing metadata files for use in the detection stage. Advanced statistical modeling and machine-learning techniques can reduce the amount of human overhead required by inferring this information directly from data [5]–[7]. These approaches show great potential for increasing the efficiency of the onset detection task but maintain a bias based on the training data.

While onset detection algorithms have been successfully applied to a variety of instruments and musical genres, vocal music presents long-recognized challenges, including soft attacks and the absence of fixed tuning or temperament [5], [8]. These issues are further compounded in the case of *a cappella* medieval vocal music, which stems from a notational and performance tradition that lacks many of the structural features leveraged by modern onset-detection algorithms. In particular, the original notational system based on *neumes* reflects a fundamentally different relationship to pitch, rhythm, and text than that implied by modern staff notation.

Efforts to generate symbolic representations, such as MIDI scores, from these sources often fail to capture the interpretive flexibility of performance and risk imposing artificial rhythmic structures on repertoire not composed with such regularity in mind. As with other musics outside the Western metric-tonal canon such as South Indian Carnatic music [9], *a cappella* Turkish Makam [8], or melismatic Flamenco [10], the distinct characteristics of medieval choral music suggest the need for adapted alignment and onset detection strategies.

This study addresses these challenges in the context of the analysis of a large dataset of medieval choral performances. The proposed method adapts audio-alignment techniques to this application by classifying audio frames into neume classes using a jumping dynamic time warp (JDTW) algorithm. Iterative maximum a posteriori (MAP) estimation of each neume's fundamental frequency refines the onset detection by dynamically accounting for both pitch and temporal variation across performance conditions. The method's robustness is demonstrated through comparison with baseline algorithms applied to professionally recorded medieval vocal ensembles.

979-8-3315-7206-8/25/$31.00

II. Methods

A. Musician Recordings and Data Collection

1) Musical Features: The choral pieces considered in this work come from the 10th to 14th centuries. While this repertoire influenced many aspects of later European musical traditions, its formal features pose unique challenges to contemporary onset detection approaches.

In the earliest pieces, rhythm is determined by the choir's interpretation of Latin textual stress rather than by any consistent pulse. In later examples, rhythmic modes (standardized groupings of long and short durations) structure the music, but without a fixed proportional relationship between those durations or any metric division such as bar lines [11]. The musicians in this dataset performed from photocopies of original medieval manuscripts, interacting directly with the neume notation in its historical form. This notational system indicates melodic contour, textual association, and general rhythmic shape, but does not specify absolute pitch (e.g., "A4 = 440 Hz"), assume equal temperament, or provide exact syllable-level timing alignment. As a result, even the most regular passages in this repertoire resist alignment with modern beat-based timing frameworks.

Some passages of the dataset are also highly melismatic. In one excerpt, a single syllable spans over 100 notated pitches, performed over 60 s to 80 s. This extreme prolongation further challenges onset detection methods that rely on tight coupling between text and note onsets. Finally, as with choral performances of a more modern repertoire, the choirs adjust tuning both horizontally (melodically) and vertically (harmonically) [12], complicating any alignment strategy that depends on pitch stability.

2) Data Set: The dataset this algorithm was developed for is a selection of choral recordings featuring *a capella* monophonic and polyphonic vocal performances [13]. The recordings were made under anechoic conditions with head-mounted microphones, resulting in monophonic recordings with good bleed-through isolation from other members of the ensemble. In all, the dataset includes approximately 400 minutes of 4-channel recordings, containing approximately 110,000 vocal onsets between all vocal parts.

The ground-truth onsets used in the algorithm development were manually annotated by the authors in Sonic Visualiser [14] and imported into Matlab 2023a for reprocessing with the MIRtoolbox (Version 1.8.2) [15]. The function `mironsets` was directed to detect onsets using the *'Emerge'* process [16] within a 200 ms window centered on each manual annotation using the MIRtoolbox wrapping routine `MM-BOP` [17]. This approach allowed for an objective verification of the initial annotations.

B. Onset Detection Algorithm

1) Onset Detection as a Classification Problem: For MPA tasks, alignment-based onset detection approaches have several advantages, including a one-to-one matching between the number of notes and the number of onsets. However, score alignment algorithms typically make use of a MIDI score, which contains precise pitch and timing information [2], [4]. In some works, the MIDI score is directly synthesized to perform an audio-to-audio alignment [4] and may incorporate accompanying lyrics to improve the match [8], [18]. However, the neumes used to notate medieval contrapuntal music significantly vary from a MIDI score because they 1) do not convey precise pitch information and 2) do not convey precise rhythmic information. Rather, neumes provide a relative ordering in both time and pitch of the sung notes, although there may be significant flexibility in the final musical realization. Such a concept is fairly similar to that applied in [10] for the automatic transcription of Flamenco music. However, this work leverages the known neumes to guide pitch segmentation for improved performance.

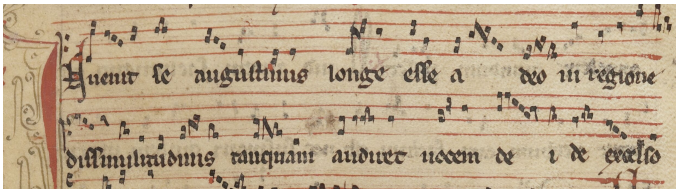

Fig. 1. Score of excerpt 1, used by the musicians: *Responsory*. F-Pn NAL.1235 f. 255r.

Treating onset detection as a classification problem facilitates the adaptation of commonly employed concepts in score alignment algorithms. Consider a musical score consisting of N neumes. From this score, define $N+1$ classes denoted C_n, one for each neume plus one class representing any pauses (referred to as "rests" in this work) between notes. Next, consider an audio recording of the score subdivided into M frames, each frame being regularly spaced by a time step Δt. The objective of the alignment and onset detection algorithm is to assign each frame of audio data at time t_m to one of the $N+1$ classes C_n. Because the audio frames associated with the nth neume class have a temporal order, the frame with the earliest time t_m for a fixed class n is the *onset*, while the frame with the latest time is the *offset*. Lastly, the sequential order of each of the N neume classes dictated by the score also imposes an additional requirement: once a frame t_m is assigned to the nth class, all frames with index $i > m$ must be assigned to a neume class with $j \geq n$. A dynamic time-warping (DTW) algorithm will enforce this final constraint.

2) Jumping Dynamic Time Warping: Dynamic time warping (DTW) is commonly used for score alignment algorithms [2], [4]. Given two sequences of possibly differing lengths, it finds the pairing of sequence indices which minimizes a distance metric accumulated over the aligned sequence, referred to as the DTW distance. The most basic formulation of the DTW applies constraints on the allowable alignment paths including monotonicity (the aligned sequence indices must be non-decreasing), continuity (the aligned sequence indices must not skip any element), and boundary conditions (the aligned sequence indices cannot deviate too far from a reference path).

In addition to the formulation and constraints of the DTW,

a key aspect of the algorithm is determining which feature should be used to calculate the DTW distance. Previous works employing DTW have used spectral features, such as the energy within fixed bandwidths defined by a harmonic series, to align recorded music with a score [2], [4]. However, the spectral characteristics of medieval music can diverge from the rigid harmonic structure that may be assumed for contemporary music. When performing MIR tasks on musical styles with more exacting pitch variations, previous works have used fundamental frequency as the defining acoustic feature [9], [10]. Similarly, this work uses the Euclidean distance between fundamental frequencies as the DTW distance, providing narrower spectral resolutions to properly resolve differences between notes.

The DTW aligns the neumes with an audio recording to determine vocal onsets by creating a mapping between two sequences. The first sequence, denoted $\mathbf{f}$ with elements f_m, contains the calculated fundamental frequencies for each of the M time frames. Because human perception and musical scores are more closely aligned with logarithmic scales, the sequence uses fundamental frequency converted to units of semitones rather than the SI units of Hertz. The second sequence, denoted $\mathbf{F}$, is formed by interleaving the estimated fundamental frequency F_n of each of the N scored neumes with a null value $\varnothing$ representing the rest class, i.e., $F = [\varnothing, F_1, \varnothing, F_2, \varnothing, \cdots, \varnothing, F_N, \varnothing]$. As a result, $\mathbf{F}$ is of length $2N + 1$. The interleaving of a null value between neumes is required because neumes, unlike a MIDI score, do not always precisely indicate where pauses between musical notes might occur. Relaxing the continuity constraints of the DTW to create a jumping dynamic time warp (JDTW) allows the algorithm to skip or 'jump' over the null value if not needed as discussed in [4], [19] and illustrated in Fig. 2.

3) Maximum A Posteriori Class Pitch Adjustment: Previous DTW-based alignment algorithms have relied on the assumption that the played pitch should nearly exactly match those written in the score. In contrast, the medieval music considered in this work requires a more flexible approach to the relationship between the pitch of the scored neume and that produced by the singers for several reasons. First, a choir has complete freedom to choose the starting pitch of the song, so long as the rest of the melody follows the prescribed church mode. Second, melismatic expressions, portamenti, ornamentation, and glissandi can directly lower, raise, or otherwise alter the pitch of a given neume, referred to here as fast-time variations. Third, approximately constant but relative shifts and gradual pitch alterations that occur during *a cappella* performance, such as pitch drift, can lead to slow-time variations.

In order to account for these possible variations, the present work proposes to iteratively refine the assigned fundamental frequency F_n of each of the N neumes in the score using a maximum a posteriori (MAP) estimate. Thus, rather than assuming the pitch of a sung neume will always be F_n, one assumes that the sung pitch follows a normal distribution $N(F_n, \sigma^2)$, where σ^2 represents the variability of this prior

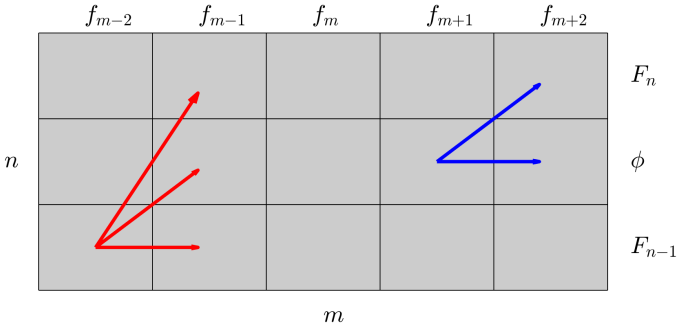

Fig. 2. Possible path choices for the JDTW if the current class index n corresponds to a (red arrow) neume class or the (blue arrow) rest class.

assumption. For example, for contemporary popular music with strict tuning, selecting $\sigma^2 \to 0$ indicates that the sung f_m should be very close the scored F_n. On the other hand, for medieval vocal music with more varied pitch realizations, σ^2 should be larger to reflect the higher uncertainty in the sung note relative to the scored value.

To apply a MAP estimate of the class pitch to the alignment algorithm, the fundamental frequencies of each neume class F_n are initially set to the relative equal temperament values given in the score, representing the prior assumption. Next, the JDTW algorithm performs a first alignment between the measured fundamental frequencies f_m with the scored frequencies F_n, classifying each frame into either one of the N neume classes or into the rest class. For example, the vertical black lines in Fig. 3(a) delineate the frames f_m belonging to each neume over a selection of excerpt 1 after the first alignment.

However, the measured values of f_m, shown as colored dots in Fig. 3(a), will rarely exactly match the initial value F_n from the score, indicated as the solid red horizontal lines in Fig. 3(a). Let

$$\hat{f}_n = \frac{1}{K_n} \sum_{m \in C_n} f_m \tag{1}$$

be the sample mean of a given neume class C_n with K_n members of the set. Furthermore, let

$$\sigma_n^2 = \frac{1}{K_n - 1} \sum_{m \in C_n} (f_m - \hat{f}_n)^2 \tag{2}$$

be the sample variance of the fundamental frequency of the class C_n. The maximum a posteriori estimate of the class's fundamental frequency follows as [20]

$$F_n^{(MAP)} = \frac{\sigma^2 K_n}{\sigma^2 K_n + \sigma_n^2} \hat{f}_n + \frac{\sigma_n^2}{\sigma^2 K_n + \sigma_n^2} F_n \tag{3}$$

The MAP estimate can be interpreted as a weighted average of the class sample mean $\hat{f}_n$ and the assumed F_n prior value derived from the score.

Several advantages of using a MAP estimate for iteratively adjusting the class fundamental frequencies F_n become clear from this equation. In particular, the MAP estimate ensures that significant adjustments to F_n are made only when 1) the variance σ_n^2 of the assigned members of a class is small,

indicating a consistently different value and 2) the number of assigned members of a class K_n is large, indicating a large number of frames belonging to a class. This latter condition is important because the JDTW algorithm will occasionally assign an unfeasibly small duration (e.g., one or two frames) to a class, such as visible at the 1.8 s mark in Fig. 3(a). A final benefit of this approach is that setting $\sigma^2 \to 0$ removes the MAP adjustments, returning the algorithm to a JDTW alignment. Consequently, the proposed technique extends previous DTW algorithms to handle more complex pitch variations.

After calculating the new neume pitch $F_n^{(MAP)}$ for each of the N neumes, a second JDTW aligns the sequences f_m and $F_n^{(MAP)}$, typically leading to some new class adjustments and reassignments (see Fig. 3(b)). The process of recomputing the neume's musically realized pitch through a MAP estimate could be iterated until convergence, but a modified approach was found to yield faster and more accurate results. After the second JDTW between f_m and $F_n^{(MAP)}$, $F_n^{(MAP)}$ is recalculated using the new class assignments but with the old priors F_n. Then, the mean deviation of each class from the score, calculated as

$$\Delta F = \frac{1}{N}\sum_{n=1}^{N}(F_n^{(MAP)} - F_n) \tag{4}$$

is used to simultaneously adjust all the priors as

$$F_n^{(i+1)} = F_n^{(i)} + \Delta F, \tag{5}$$

where i is used here to indicate the iteration number. This step improves the convergence rate because it directly accounts for global (slow-time) pitch variations occurring in the score by uniformly adjusting all neume pitches. In contrast, the MAP estimates modify each neume pitch individually (fast-time variations), which may take more iterations to converge depending on the variance of a given class.

To summarize, each iteration of the algorithm computes two JDTW alignments, making both fast- and slow-time adjustments to the neume's realized fundamental frequencies. The process continues until the class assignments do not change after a completed full iteration. The entire procedure is outlined as Algorithm 1 and illustrated in Fig. 3(a)-(c).

III. Results

A subset of the dataset, 80 monophonic recordings of excerpt 1, served as the reference data for benchmarking the proposed algorithm. This reference contained 8320 manually annotated onsets following the two-step process outlined in Sec. II-A2. In addition to the JDTW+MAP approach, which used a ∼12 ms time step and a YIN-based fundamental frequency estimation, the benchmark test includes five audio-only algorithms and two audio-alignment algorithms. Parameter values for each algorithm were derived by manual tuning on one complete recording of the excerpt, prioritizing the reduction of false positives.

Algorithm 1 MAP JDTW

procedure CLASSIFY(**f**)
  $F_n \leftarrow$ neumes ▷ Class frequencies from neumes
  while $C_n^{(i)} \neq C_n^{(i-1)}$ **do**
    $C_n \leftarrow \text{JDTW}(f_m, F_n)$ ▷ Classify f_m
    $F_N^{(MAP)} \leftarrow \frac{\sigma^2 K_n}{\sigma^2 K_n + \sigma_n^2}\hat{f}_n + \frac{\sigma_n^2}{\sigma^2 K_n + \sigma_n^2}F_n$
    $C_n \leftarrow \text{JDTW}(f_m, F_n^{(MAP)})$ ▷ Re-classify f_m
    $F_N^{(MAP)} \leftarrow \frac{\sigma^2 K_n}{\sigma^2 K_n + \sigma_n^2}\hat{f}_n + \frac{\sigma_n^2}{\sigma^2 K_n + \sigma_n^2}F_n$
    $\Delta F \leftarrow \frac{1}{N}\sum_{n=1}^{N}(F_n^{(MAP)} - F_n)$
    $F_n \leftarrow F_n + \Delta F$ ▷ Re-assign F_n
  end while
  return C_n
end procedure

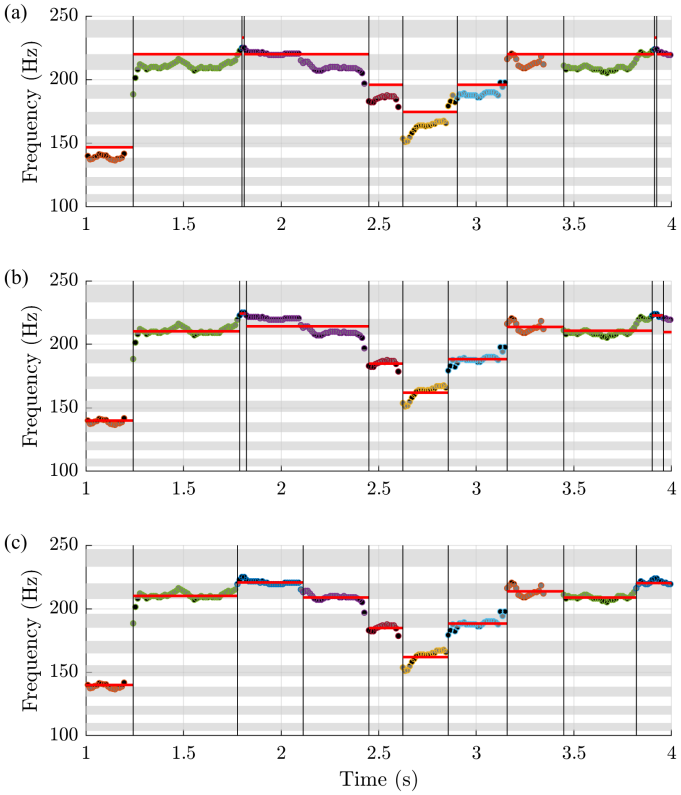

Fig. 3. A selection of a performance of excerpt 1 during the segmentation process. The white and gray shading indicates the boundaries of each semitone. (a) Initial JDTW. (b) Second JDTW after MAP class refinements. (c) Final class assignments after five complete iterations. Of note is the improvement in segmenting the third and tenth neumes, occurring around 2 s and 4 s, respectively.

The benchmark selection prioritized algorithms that were readily accessible via open-source toolboxes or libraries. The audio-only algorithms used in the benchmark included 1) the Spectral Flux (SF) approach [21], [22], implemented in the `librosa` [23] python library with backtracked peak detection, 2) the SuperFlux (SuF) [24] algorithm and 3) a pretrained convolution neural network (CNN) [25], both implemented in the `madmom` [26] library with its native peak-picking

algorithm, 4) the monophonic note segmentation algorithm CREPE Notes (Cr) [27], and finally 5) a base version of the Emerge (EM) method without metadata guidance from the `MIRtoolbox` in Matlab [15], [16]. As the EM method returns a list of ranked onset possibilities, the benchmark incorporates the 104 strongest onsets.

The audio-alignment algorithms included in the benchmark were 1) a standard DTW alignment [21] to a MIDI approximation of the excerpt, and 2) an implementation of the JDTW algorithm without any MAP adjustments.

The benchmarking algorithms performances are summarized in Table I using a commonly employed $50\,\mathrm{ms}$ tolerance window around the ground truth onsets [5].

Algorithm	Precision	Recall	F-score	# Onsets
SF	0.42 ± 0.02	0.19 ± 0.01	0.26 ± 0.01	45.52 ± 2.99
SuF	0.80 ± 0.03	0.07 ± 0.01	0.13 ± 0.01	8.44 ± 0.90
CNN	0.59 ± 0.03	0.30 ± 0.02	0.40 ± 0.03	47.67 ± 2.59
Cr	0.36 ± 0.02	0.60 ± 0.02	0.45 ± 0.02	164.66 ± 5.38
EM	0.48 ± 0.01	0.53 ± 0.01	0.50 ± 0.01	104 ± 0.00
DTW	0.46 ± 0.01	0.48 ± 0.02	0.47 ± 0.02	104 ± 0.00
JDTW	0.57 ± 0.02	0.59 ± 0.02	0.58 ± 0.02	104 ± 0.00
JDTW+MAP	0.69 ± 0.02	0.72 ± 0.01	0.70 ± 0.01	104 ± 0.00

TABLE I
COMPARISON OF ONSET DETECTION METHODS USING A 50 MS TOLERANCE WINDOW. VALUES ARE REPORTED AS MEAN ± 95% CONFIDENCE INTERVAL ACROSS 80 RECORDINGS, EACH CONTAINING 104 ONSETS.

IV. DISCUSSION

The design and implementation of any onset detection technique should align with its intended use. The proposed JDTW+MAP algorithm was developed within the context of a broader research pipeline for an MPA task analyzing expressive timing in sung medieval choral music under varying performance conditions. Consequently, a central design consideration was to minimize the time required for post-processing manual onset validation, rather than merely optimizing a chosen metric as a formality.

For instance, while the audio-only algorithm SuF achieved higher precision than the proposed JDTW+MAP, this advantage is significantly undermined by its identification of, on average, less than 10% of the total onsets in a score. Similarly, although another audio-only method, Cr, yielded comparable recall to JDTW (without the MAP adjustments), it over-predicted onsets by more than 50% on average. As a result, both of these algorithms would require significant manual effort in either detecting missing onsets or removing spurious ones. Consequently, while standard metrics like precision, recall, and F-score do offer some insight into overall algorithm performance, they often fail to fully capture the practical overhead of additional manual processing required for real-world MPA tasks using MIR tools to expedite a laborious initial step.

One particularly noteworthy result is the performance of the simple yet robust DTW algorithm, whose F-score surpassed four of the five audio-only methods, including both of the machine learning approaches, CNN and Cr. Furthermore, allowing for jumps or gaps in the alignment process through JDTW–an approach applied with various adaptations for over half a century–performed better, with respect to the F-score, than *all* audio alignment methods, including state-of-the-art machine learning approaches. The proposed method, which incorporates MAP class pitch estimates to the JDTW, led to the highest F-score of both the audio-only and audio-alignment methods. The additional flexibility introduced by the MAP pitch adjustments to the JDTW thus built upon an already strong algorithmic approach to achieve both higher precision and recall. These results underscore the versatility of the DTW-based algorithms due to their data-independent formulations, which allow them to generalize to musical repertoires like medieval choral music that are underrepresented in public databases.

Other improvements to the alignment algorithm could be made, for example, by refining the MAP estimation procedure to operate without the need for a predefined estimated starting pitch, or developing a model which incorporates the known lyrics [8], [18], increasing the autonomy and generalization of the algorithm. However, some medieval vocal music consists of a single sung word, making lyrics less feature-rich compared to other musical styles.

V. CONCLUSIONS

This work proposed an alignment algorithm for determining vocal onsets in medieval choral music. The algorithm employs a jumping dynamic time warp between measured fundamental frequencies and those suggested by a score to classify audio frames to an associated neume class. Iterative adjustments to the classes' pitch based on maximum a posteriori estimates enable robust alignment even with the slow- and fast-pitch changes occurring in *a cappella* vocal music. Comparisons with seven other onset detection algorithms demonstrated improved performance with respect to recall and F-score. Future work should include adapting the approach to polyphonic recordings, instrumental music, or other music genres and cultures outside of popular Western music.

VI. ACKNOWLEDGMENT

The ANR Project PHEND, Grant No. ANR-20-CE38-0014 supported this work.

VII. AUTHOR CONTRIBUTION STATEMENT

S.D.B. conceived and implemented the proposed algorithm. S.S.M. identified the core research problem and performed the data collection and benchmarking. B.F.G.K. oversaw the project. All authors assisted with the writing. S.D.B and S.S.M contributed equally to the work and share first authorship.

REFERENCES

[1] J. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler, "A tutorial on onset detection in music signals," *IEEE Trans. Speech Audio Process.*, vol. 13, no. 5, pp. 1035–1047, 2005. DOI: 10.1109/TSA.2005.851998.

[2] N. Orio and D. Schwarz, "Alignment of monophonic and polyphonic music to a score," in *Proc. of the Int. Computer Music Conference (ICMC) 2001*, 2001, pp. 1–4.

[3] N. Orio and F. Déchelle, "Score following using spectral analysis and hidden markov models," in *Proc. of the Int. Computer Music Conference (ICMC) 2001*, 2001, pp. 1–4.

[4] R. J. Turetsky and D. P. W. Ellis, "Ground-truth transcriptions of real music from force-aligned MIDI syntheses," in *Proc. of the Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2003, pp. 1–7.

[5] C. C. Toh, B. Zhang, and Y. Wang, "Multiple-feature fusion based onset detection for solo singing voice," in *Proc. of the Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2008, pp. 515–520.

[6] J. Schlüter and S. Böck, "Improved musical onset detection with convolutional neural networks," in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 6979–6983. DOI: 10.1109/ICASSP.2014.6854953.

[7] R. Nishikimi, E. Nakamura, K. Itoyama, and K. Yoshii, "Musical note estimation for F0 trajectories of singing voices based on a Baysian semi-beat-synchronous HMM," in *Proc. of the Int. Society for Music Information Retrieval Conf. (ISMR)*, 2016, pp. 461–467.

[8] G. Dzhambazov, A. Srinivasamurthy, S. Sentürk, and X. Serra, "On the use of note onsets for improved lyrics-to-audio alignment in Turkish Makam music," in *Proc. of the Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2016.

[9] G. Koduri, J. Serrá, and X. Serra, "Characterization of intonation in Carnatic music by parameterizing pitch histograms," in *Proc. of the Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2012.

[10] E. Gómez and J. Bonada, "Towards computer-assisted Flamenco transcription: An experimental comparison of automatic transcription algorithms as applied to a cappella singing," *Computer Music Journal*, vol. 37, no. 2, pp. 73–90, 2013. DOI: 10.1162/COMJ_a_00180.

[11] J. Yudkin, *Music in Medieval Europe*. Prentice Hall, 1989, ISBN: 978-0-13-608225-5.

[12] D. M. Howard, "Equal or non-equal temperament in a cappella SATB singing," *Logopedics Phoniatrics Vocology*, vol. 32, no. 2, pp. 87–94, Jan. 2007. DOI: 10.1080/14015430600865607.

[13] S. S. Mullins, "Des voix du passé : the historical acoustics of Notre-Dame de Paris and choral polyphony," Theses, Sorbonne Université, Sep. 2024. [Online]. Available: https://theses.hal.science/tel-04879114.

[14] C. Cannam, C. Landone, and M. Sandler, "Sonic Visualiser: An open source application for viewing, analysing, and annotating music audio files," in *Proc. of the ACM Multimedia 2010 Int. Conf.*, Firenze, Italy, Oct. 2010, pp. 1467–1468.

[15] O. Lartillot, P. Toiviainen, and T. Eerola, "A MATLAB toolbox for music information retrieval," in *Data Analysis, Machine Learning and Applications*, Springer, 2008, pp. 261–268. DOI: 10.1007/978-3-540-78246-9_31.

[16] O. Lartillot, D. Cereghetti, K. Eliard, and W. J. Trost, "Estimating tempo and metrical features by tracking the whole metrical hierarchy," in *Proc. of the 3rd Int. Conf. on Music and Emotion (ICME3)*, Jyväskylä, Finland, Jun. 2013.

[17] P. Cairns, *Marker metadata-based onset picking (MMBOP)*, Zenodo, Jun. 2024. DOI: 10.5281/zenodo.12581698.

[18] R. Gong, P. Cuvillier, N. Obin, and A. Cont, "Real-time audio-to-score alignment of singing voice based on melody and lyric information," in *Proc. of Interspeech 2015*, 2015, pp. 1–5.

[19] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, no. 3, pp. 443–453, 1970. DOI: https://doi.org/10.1016/0022-2836(70)90057-4.

[20] T. K. Moon and W. C. Stirling, *Mathematical Methods and Algorithms for Signal Processing*. Prentice Hall, 2000.

[21] S. Dixon, "Onset detection revisited," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx)*, Montreal, Canada, 2006.

[22] S. Böck, F. Krebs, and M. Schedl, "Evaluating the online capabilities of onset detection methods," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx)*, Montreal, Canada, Sep. 2006.

[23] B. McFee, C. Raffel, D. Liang, *et al.*, "Librosa: Audio and music signal analysis in python," in *Python in Science Conference*, Austin, Texas, 2015, pp. 18–24. DOI: 10.25080/Majora-7b98e3ed-003.

[24] S. Böck and G. Widmer, "Maximum filter vibrato suppression for onset detection," in *Proc. of the Int. Conf. on Digital Audio Effects (DAFx)*, Maynooth, Ireland, Sep. 2013.

[25] J. Schlüter and S. Böck, "Musical onset detection with convolutional neural networks," in *Proc. of the Int. Workshop on Machine Learning and Music (MML)*, 2013, pp. 79–82.

[26] S. Böck, F. Korzeniowski, J. Schlüter, F. Krebs, and G. Widmer, "Madmom: A new python audio and music signal processing library," in *Proc. of the 24th ACM Int. Conf. on Multimedia*, Amsterdam, The Netherlands, Oct. 2016, pp. 1174–1178. DOI: 10.1145/2964284.2973795.

[27] X. Riley and S. Dixon, "CREPE Notes: A new method for segmenting pitch contours into discrete notes," in *Proceedings of the 20th Sound and Music Computing Conference*, Stockholm, Sweden, 2023, pp. 1–5. [Online]. Available: https://arxiv.org/pdf/2311.08884.