AUDIO-TO-SCORE ALIGNMENT SYSTEM TO SYNCHRONIZE MEDIEVAL CHANT RECORDINGS

Pablo Cabañas Molero[1], *Raquel Cortina Parajón*[2], *Jaime García Martínez*[1],
Elías F. Combarro Álvarez[3], *Pedro Vera Candeas*[11]

[1]Universidad de Jaén, Linares, España
[2]Universidad de Oviedo, Gijón, España
[3]Universidad de Oviedo, Oviedo, España

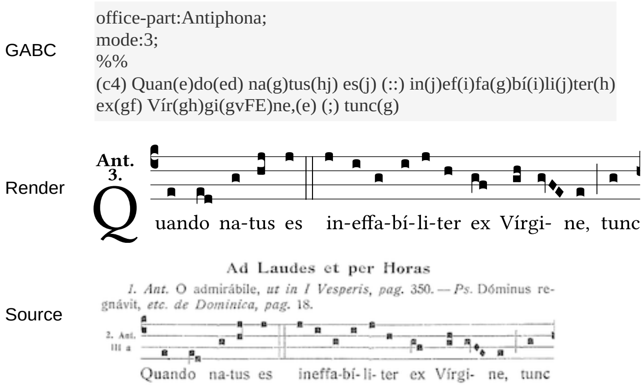
ABSTRACT

Medieval music has been systematically studied by scholars for over a century. Nowadays, extensive databases exist that contain transcriptions of Gregorian chants in GABC format. These databases often include additional media such as images of the original manuscripts or audio recordings of the chants. In this paper, we propose an audio-to-score alignment system to synchronize a chant in GABC format with its corresponding audio rendition. First the GABC score is converted into MIDI using fixed values for tempo and note duration (markings like punctum mora and episema are taken into account). The MIDI score is then aligned to the audio by computing a full pairwise distance matrix between audio and score sequences, and using a version of offline Dynamic Time Warping, which is very efficient in terms of memory and CPU. As a result, each single square-note (and its Latin syllable) is assigned a precise time within the audio performance. Our system demonstrates strong resilience towards variations in tempo and note duration, but unfortunately does not work with unison chants.

Keywords— medieval music, gregorian chant, gabc, audio-to-score alignment, dynamic time warping.

1. INTRODUCTION

The origins of Gregorian chant can be traced back to the 8th century. Early manuscripts recorded only the Latin texts, but over time sketches of the melodies began to appear in the form of *neumes*, symbolic figures inserted to denote melodic contours. These neumes were later accompanied by lines representing pitches, which soon evolved into the four-line square notation still used in chant books today. This notation is the predecessor of the modern five-live staff [1][2].



```
office-part:Antiphona;
mode:3;
%%
(c4) Quan(e)do(ed) na(g)tus(hj) es(j) (::) in(j)ef(i)fa(g)bí(i)li(j)ter(h)
ex(gf) Vír(gh)gi(gvFE)ne,(e) (;) tunc(g)
```

GABC
Render
Ant. 3.
Q uando na-tus es in-effa-bí-li-ter ex Vírgi- ne, tunc
Source
Ad Laudes et per Horas
1. Ant. O admirábile, ut in I Vesperis, pag. 350. — Ps. Dóminus regnávit, etc. de Dominica, pag. 18.
3. Ant. III a
Quando na-tus es ineffa-bí-li-ter ex Vírgi- ne, tunc

Figure 1. *Quando natus est* chant in GregoBase. Chant is stored in GABC format, with its rendered version and source book image**.**

Systematical study of Gregorian chant has been active for over a century [3]. With the advent of computers and the internet, tools emerged to digitize and preserve this early music. The main tool in this field is the Cantus Index [4], an online index of chants that provides their full text and a *Cantus ID* to identify the same chant across different databases and sources. Several databases are indexed in this catalogue, the largest one being the Cantus database [5]. This archive of chants contains about 500,000 entries, which include source (manuscript and its location), Cantus ID, image of the source, and more. Around 13% of these chants have their melodies transcribed using *Volpiano* notation, a specialized typeface designed for notating plainchant [2].

GregoBase is an online corpus that hosts around 9,000 chants of the Gregorian repertoire [6]. These chants are presented in the same way as in modern chant books like the *Liber Usualis:* modern four-line square notation with elements such as breathing marks, varied note shapes and clef changes [2]. These transcriptions are stored in GABC

[1] ***Contact author**: pvera@ujaen.es*

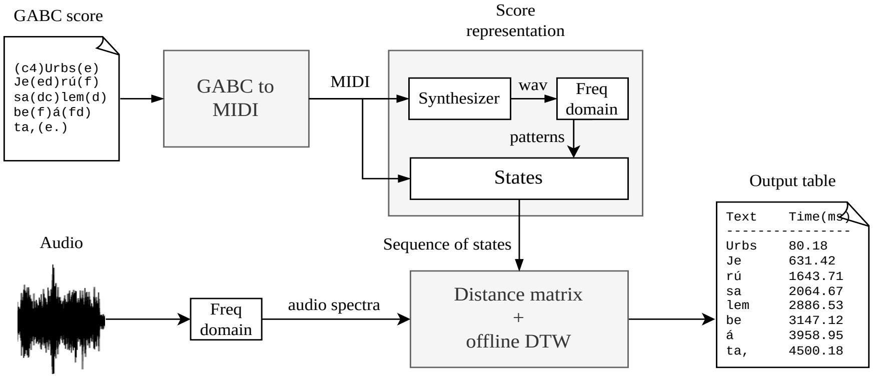


Figure 2. Block diagram of the proposed audio-to-score alignment method for Gregorian chant**.**

format, an elaborate TEX style notation for representing Gregorian chant using plain text, and part of the *Gregorio* project [7]. An example of chant written in GABC (with its rendered version and source) is illustrated in Figure 1. Regarding audio data, the commercial mobile application *Neumz* offers 7,000 hours of recordings of Gregorian chant [8]. Each audio is played in sync with its square-note score in the screen, which is a rendered version of a GABC transcription.

Given the amount of information stored in these databases, more and more research has been focused on providing computational methods to process and study Gregorian chant. One example is *transcript alignment*, whose purpose is to associate the transcribed chant text with its position in the manuscript image [9]. In this work, we focus on *audio-to-score alignment*, i.e., associating each square-note (and Latin syllable) of the chant to a position in an audio rendition. Even though score alignment methods for classical music have been around for almost four decades [10], to our knowledge, no previous systems have been applied to vocal plainchant. In the *Neumz* application, this alignment labor is done manually.

In this paper, we propose an audio-to-score alignment system designed to synchronize chants in GABC format with their corresponding audio representations. The method is an adaptation of the algorithm in [11], originally designed for classical music. Initially, we convert the GABC score into MIDI format while considering predefined values for tempo and note duration. Next, we align the MIDI score with the audio by calculating a pairwise distance matrix between the audio and score sequences. We employ an efficient version of offline Dynamic Time Warping (DTW), which optimizes memory and CPU usage. The algorithm assigns precise timing to each square-note within the audio performance.

In the next section, we describe our approach in detail. In section 3, our experiments on score-audio pairs taken from *Neumz* are discussed.

2. ALIGNMENT METHOD

The block diagram of the proposed method is shown in Figure 2. The system has two inputs: a plain text file containing the score in GABC and an audio file. The output is a table associating each Latin syllable (or note) with a time position. The alignment is done offline, as future audio information is used during the process.

2.1. GABC to MIDI conversion

The first step is to convert the score from GABC to MIDI. For this we use the open source tool *gabctk* [12]. This software parses the GABC file and assigns a time position and duration to each square note. By default, a tempo of 165 beats per minute (bpm) is used, and each note is given a duration of 1 beat. Specific musical notations that alter the duration of notes are also taken into account, such as *episeme* (note is sung slightly longer or with a slight emphasis, so is set to 1.7 beats), *quilisma* (adds a trill or vibrato effect, involving a prolongation of the previous note, which is set to 2 beats) and *punctum mora* (tipically doubles the note it affects, but is set to 2.3 beats).

The Latin syllables are also extracted and inserted into the MIDI file as meta-events with the timing of their first note. Notice that a single syllable can be sung on one or more notes. An example of a GABC to MIDI conversion is illustrated in Figure 3.

2.2. Score representation

Given the score in MIDI format, a preprocessing stage is carried out to convert it into a suitable format for alignment. The score is represented as a sequence of *states*, where each state is a portion of the score delimited by adjacent onsets or offsets. Since plainchant is essentially monophonic, states often consist in a single note. Each state n is composed by

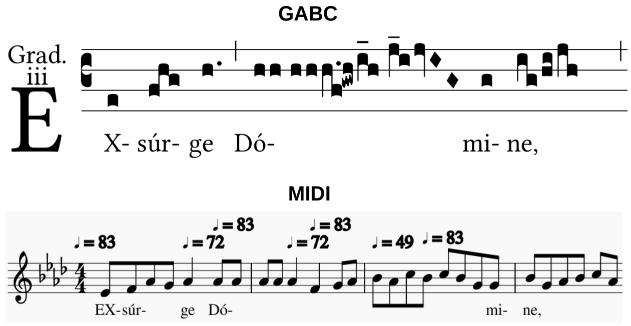


Figure 3. Conversion of a chant encoded in GABC to MIDI format using gabctk. Notes are given a duration of 1 beat at 165 bpm, with some markers changing the duration.

four parameters: start time, end time, Latin text syllabe and a spectral pattern $\mathbf{u}_n$.

To compute the spectral patterns $\mathbf{u}_n$, the score is converted to a waveform using a synthesizer and then transformed to the time-frequency domain. All frames with the same combination of notes are stacked into a matrix, which is then reduced to a single pattern using a Non-negative Matrix Factorization (NMF) model with one basis vector. The process is repeated for all combinations present in the score. Note that different states are given the same spectral pattern if they have the same combination of notes.

To allow the alignment with DTW, the score is treated as a sequence of time frames $U = \{\mathbf{u}_{n(1)}, \ldots, \mathbf{u}_{n(\tau)}, \ldots, \mathbf{u}_{n(\Gamma)}\}$, where $\mathbf{u}_{n(\tau)}$ is the pattern corresponding to score frame τ and Γ is the number of score frames. Here, we denote the state associated to score frame τ as $n(\tau)$.

2.3. Cost matrix

The alignment module is responsible for determining the position of the score in the audio. Formally, the goal is to find the score time τ that corresponds to each audio time $t \in [1, T]$. For this task, the system computes a comparison measure between each incoming audio frame, denoted as $\mathbf{v}_t$ in the frequency domain, and each element in U, and then estimates the best alignment path through the comparison matrix.

The cost measure $d(\tau, t)$ between $\mathbf{v}_t$ and each position τ in the score is given by

$$d(\tau, t) = D_\beta \left(g_{\tau,t} \mathbf{u}_{n(\tau)}, \mathbf{v}_t \right), \tag{1}$$

where $D_\beta(\cdot, \cdot)$ is the β-divergence function, $\beta = 1.5$ and $g_{\tau,t}$ represents the gain of $\mathbf{u}_{n(\tau)}$ that minimizes its β-divergence to $\mathbf{v}_t$. This gain can be computed as:

$$g_{\tau,t} = \frac{\left\| \mathbf{v}_t \circ \mathbf{u}_{n(\tau)}^{(\beta-1)} \right\|_1}{\left\| \mathbf{u}_{n(\tau)}^{\beta} \right\|_1}. \tag{2}$$

The cost measure in Eq. 1 can be interpreted as the minimum β-divergence between $\mathbf{v}_t$ and $\mathbf{u}_{n(\tau)}$ up to amplitude differences. It is inspired by the NMF model in [13], where only a single basis can be active at each frame.

2.4. Alignment

The alignment path trough the matrix $d(\tau, t)$ is determined by an implementation of offline DTW. From $d(\tau, t)$, the DTW *accumulated cost matrix* D is computed using the following recursion:

$$D(\tau, t) = \min_{j,i} \left\{ \begin{array}{l} D(\tau - j, t - 1) + d(\tau, t)\omega_j \\ D(\tau - 1, t - i) + d(\tau, t)w_i \end{array} \right\}, \tag{3}$$

where j and i are the step length in each dimension, whose values are the integers in the range $j \in [1, J]$ and $i \in [1, I]$. J and I are the maximum allowed step lengths in the score and audio axes, respectively. Our system uses $I = J = 4$, which allows the audio tempo to be between 1/4 and 4 times the MIDI tempo. In reality, Gregorian chant follows a melodic-verbal rhythm, where syllable and note duration varies depending on context and importance within the word/phrase. In our experiments, however, we found this settings to be robust enough in practice. The weight ω_j represents the cost associated with step $(j, 1)$, and w_i is the cost of step $(1, i)$. We use $\omega_j = 1$ and $\omega_i = i$.

Accordingly, the best step $(j, 1)$ or $(1, i)$ for each cell (τ, t) is stored in a matrix P as follows:

$$P(\tau, t) = \arg\min_{j,i} \left\{ \begin{array}{l} D(\tau - j, t - 1) + d(\tau, t)\omega_j \\ D(\tau - 1, t - i) + d(\tau, t)w_i \end{array} \right\}. \tag{4}$$

The matrix P allows the alignment path to be constructed from the end of the audio, in a traceback stage. In ordinary DTW, the last pair of the alignment path is (τ_{min}, T), where $\tau_{min} = \arg\min_\tau D(\tau, T)$, and then the path is retrieved backwards by indexing P. In our implementation, however, instead of using the whole audio, the alignment result for each frame t is backtracked from 40 seconds in the future. This allows us to limit memory usage even for very long audio without loss of accuracy, at the cost of a slight increase in CPU usage (the backtrack path is computed for each t). An efficient implementation of our DTW variant is described in [14]. Finally, from the alignment path, the score state (and hence the note and Latin syllable) corresponding to each audio time can be determined.

3. EXPERIMENTS AND DISCUSSION

We evaluated our method against five manually annotated chant recordings included in the Neumz application. These recordings are high quality performances by the nuns of the Abbey of Notre-Dame de Fidélité of Jouques. The sound is recorded live in the Abby, with some reverberation and ambient noise, sometimes accompanied by an organ. The five chants tested are, as named in neumz: *Quando natus est* (4 minutes), *Exsurge Domine fer opem* (3.5 minutes), *Urbs Jerusalem beata* (3 minutes), *Vocem jucunditatis* (4 minutes) and *Regem Confessorum Dominum* (7 minutes). The annotations consist of a list of verse lines, each associated with a point in the audio. As a result, we could only evaluate the alignment of the verses, even though our method operates at note level. We calculated the alignment accuracy as the percentage of verses that deviated less than a threshold (or tolerance window) from the reference alignment. The algorithm was run with the same signal analysis parameters as in [14].

The results of our evaluation are shown in Table 1. Tolerance windows of 0.5s and 1s were used. The maximum detected deviation is also shown in the last column.

Table 1. The per-verse accuracy of our score-to-audio alignment. Ground-truth was manually generated.

Chant	#Verses	Acc (0.5s)	Acc (1s)	Max dev.
Quando	30	90%	96.6%	1.8 s
Exsurge	8	75%	100%	0.9 s
Urbs	31	80.6%	100%	0.9 s
Vocem	13	70%	77%	1.7 s
Regem	43	65.1%	95.4%	1.2 s

The method yields results of sufficient quality to enable audio reproduction in sync with the musical score. To provide a more subjective evaluation, we generated a subtitle file with the synchronized Latin text, and it was observed that nearly all syllables were correctly aligned at first glance. Most errors occurred in unison segments where the same pitch was sustained over several syllables. Since our method is pitch-based, it is unable to detect individual syllables in such cases. The method seems to be resistant to background sounds and handles silent pauses between phrases well. Although plainchant scores do not specify precise tempo or note durations, the system was able to align the audio with good accuracy using our settings.

4. ACKNOWLEDGMENTS

We thank John Anderson from Neumz for providing the recordings/scores used in our experiments. This work was supported by the Regional Government of Andalucıía under the PAIDI 2020 Framework Programme through grant P18-TPJ-4864 and the REPERTORIUM project. Grant agreement number 101095065. Horizon Europe. Cluster II. Culture, Creativity and Inclusive society. Call HORIZON-CL2-2022-HERITAGE-01-02.

5. REFERENCES

[1] H. Strayer, “From Neumes to Notes: The Evolution of Music Notation,” *Musical Offerings*, vol. 4, no. 1, pp. 1–14, 2013.

[2] B. Cornelissen, W. Zuidema, and J.A. Burgoyne, “Studying large plainchant corpora using chant21,” in Proc. of the 7th International Conference on Digital Libraries for Musicology, (Montréal, Canada), pp. 40–44, 2020.

[3] J. Stinson, “Medieval Music on the Web: Musical Resources for the 21st Century,” *Australian Academic & Research Libraries*, vol. 28, no. 1, pp. 65–74, 1997.

[4] D. Lacoste, and J. Koláček, “Cantus index: Online catalog for mass and office chants,” URL: http://cantusindex.org/.

[5] D. Lacoste, “The cantus database: Mining for medieval chant traditions,” *Digital Medievalist*, vol. 7, 2012.

[6] O. Berten and contributors, “Gregobase: A database of gregorian scores,” URL: https://gregobase.selapa.net/.

[7] Gregorio, “The Gregorio project,” URL: http://gregorio-project.github.io/index.html.

[8] Neumz, URL: https://neumz.com.

[9] T. de Reuse, and I. Fujinaga, “Robust transcript alignment on medieval chant manuscripts,” in Proc. of the 2nd Int. Workshop on Reading Music Systems, (Delft, The Netherlands), 2019.

[10] A. Arzt, *Flexible and Robust Music Tracking*, Ph.D. dissertation, Johannes Kepler University Linz, 2016.

[11] F. J. Rodríguez-Serrano, J. J. Carabias-Orti, P. Vera-Candeas, and D. Martinez-Munoz, “Tempo driven audio-to-score alignment using spectral decomposition and online dynamic time warping,” *ACM Trans. Intell. Syst. Technol*, vol. 8, no. 2, pp. 1–20, 2016.

[12] Gabctk, URL: https://github.com/jperon/gabctk.

[13] J. J. Carabias-Orti, F. J. Rodríguez-Serrano, P. Vera-Candeas, F. J. Cañadas-Quesada, and N. Ruiz-Reyes, “Constrained non-negative sparse coding using learnt instrument templates for realtime music transcription,” *Engineering Applications of Artificial Intelligence*, vol. 26, no. 7, pp. 1671–1680, 2013.

[14] P. Cabañas-Molero, R. Cortina-Parajón, E.F. Combarro, P. Alonso, and F.J. Bris-Peñalver, “HReMAS: Hybrid real-time musical alignment system,” *The Journal of Supercomputing*, vol. 75, pp. 1001–1013, 2019.