Modeling and Editing Cross-Modal Synchronization on a Label Web Canvas

Louis Garczynski, Mathieu Giraud, Emmanuel Leguy, Philippe Rigaux

Modeling and Editing Cross-Modal Synchronization on a Label Web Canvas

Louis Garczynski	Mathieu Giraud	Emmanuel Leguy	Philippe Rigaux
Université de Lille	Université de Lille	Université de Lille	CNAM
France	France	France	France
louis.roc@gmail.com	mathieu@algomus.fr	manu@algomus.fr	philippe.rigaux@lecnam.net

Abstract

We present how the open-source web framework Dezrann enables to hear, study, and annotate music by interacting with *synchronized views* such as image scores, rendered scores, videos, and representations of audio as waveforms or spectrograms. We encode as *unit conversions* the cross-modal synchronization between these music representations. Composing these conversions allows us to conveniently translate musical time, audio time, and graphical positions, as well to synchronize annotation labels across views. Usages of the Dezrann platform or of some of its components include corpus annotation for musicology or computer music research, music education, as well as collaborative score edition and correction in the CollabScore project.

Keywords: *music annotation, music representations, web, cross-modal synchronization, score, audio*

1. Introduction

Analyzing music can be viewed as "casting light upon music" (Bent 1987). Music analysis is well-established on *scores* in common music notation. Music analysis builds upon *music annotation* of some elements in the score (patterns, harmonies, similarities, form, ...). These elements can be annotated while considering the score as Nattiez' *niveau neutre* (Nattiez, 1975), but also with an historical, comparative or aesthetic perspective (Caplin et al., 2009). We discussed before that some music annotation can be modeled as putting *annotation labels* on the score at various positions, possibly with a duration, and sometimes drawing relations between these labels (Giraud, 2018).

Music annotation also helps other music-related activities, such as teaching, performing or even composing music. The "music analyst" can thus be a researcher in music theory or in computer musicology, a teacher, a student, a performer, a composer, a music lover. They may engage with music itself through hearing or feeling, but also with scores (or

with any other music representations) through reading, annotating, and analyzing. Can they use numeric tools to help or to foster these activities?

Web platforms for music annotation. There are many web platforms to render, and sometimes to annotate, scores in common Western music notation, both academic (Hoos 1998; Couprie 2008; Pugin 2014; Carey 2016; Lepetit-Aimon 2016; Sapp 2017) and commercial (including flat.io, jellynote.com, notezilla.io). Some of these platforms are designed to analyze music and share music annotations or analyses. Within the Dezrann project (Giraud 2018), we aim at providing a cross-modal platform for easy music annotation and analysis for everyone, including for people with limited programming background but also for people needing to annotate large music corpora. Music is presented either on several *views* – as scores or waveforms/spectrograms. The user creates, updates and edits *labels* on any of these views.

Synchronizing sources. Any cross-modal platform for music annotation faces the problem of music *representation and encoding,* and especially the link between *sources* with different representations – including audio, symbolic, image, and annotations.

As far as possible, generic and portable standards should be used to model links between documents. The W3C Web Annotation Data Model[1] enables linking towards any resource. The IIIF Image API 3.0 specification[2] allows anchors to any region in an image.

On the image side, Optical Musical Recognition (OMR) is a very active field of research (see review in (Calvo-Zaragoza 2020)). Beyond OMR, some studies addressed image-score alignment. Transcribing music often needs expert and/or crowd correction or raw input (Burghart 2017; deGroot Maggetti 2020).

On the audio side, software like Sonic Visualiser[3] or Audacity[4] allows to tap *markers* -- that can be beats, measures, or noteworthy points -- and to further edit these markers, but without any notion of (symbolic) musical time. Full-featured Digital Audio Workstations (DAW) such as Ableton, Logic, or Ardour[5] do model measure/beat information, and sometimes offer a visualization of a *grid* of the musical time. A typical DAW task is to *move* or sometimes *stretch* audio to fit such a pre-existing grid. On the other hand, for some applications such as score following, analysis or more generally research in digital humanities, the audio may be given, and one goal may be to synchronize that audio (without altering it) with a (symbolic) music representation. Synchronization between audio and musical time can be either as simple as two start/end points, a few points, a systematic annotation of beats and/or measures, or a combination of these -- for example start/end points with a few more synchronization points at which the tempo is changing.

Audio-score alignment and score following are also very active fields of research (Ewert 2009; Dorfer 2018; Mueller 2019; Thickstun 2020). Common methods are dynamic

[1] https://www.w3.org/TR/annotation-model
[2] https://iiif.io/api/image/3.0/#41-region
[3] https://www.sonicvisualiser.org
[4] https://www.audacityteam.org
[5] https://ardour.org

time-warping algorithms, or, today, deep learning approaches. Possible strategies are to align two audio sources (one generated from the score), usually by their chromas, or to directly align chromas with the score. Libraries such as librosa[6] implement some of these algorithms. The quality of these alignments is improving but may require expert correction. Repeats (and jumps) are also challenging to model and to detect (Shan 2020). Anyway, several recent studies released significant corpora with audio/image/videos aligned to scores or other symbolic music information (Balke 2018; Kosta 2018; Weiß 2021).

Contents. We do not address here automatic synchronization issues but rather focus on how to conveniently *model, edit, and use* such synchronization information, either starting from scratch or correcting an automatic synchronization information. Hence the synchronizations could be "perfect", but will often be "approximate", and a platform has to do its best to "degrade gracefully" the available information: An approximate synchronization still enables many applications such as score following or annotations.

We present here a simple way to model cross-modal synchronizations, with interpolation, to link several music representations (Section 3). Within a complete refactor of earlier version using Vue.js, the new Dezrann framework enables the interaction with a *label canvas*, scores backends (Verovio/Lilypond), audio/video backends (HTML5, YouTube) through both a full *annotation platform* and an independant *component to edit synchronization files* between audio and symbolic representations of music (Section 3). We conclude by discussing current and future usages of these tools (Section 4).

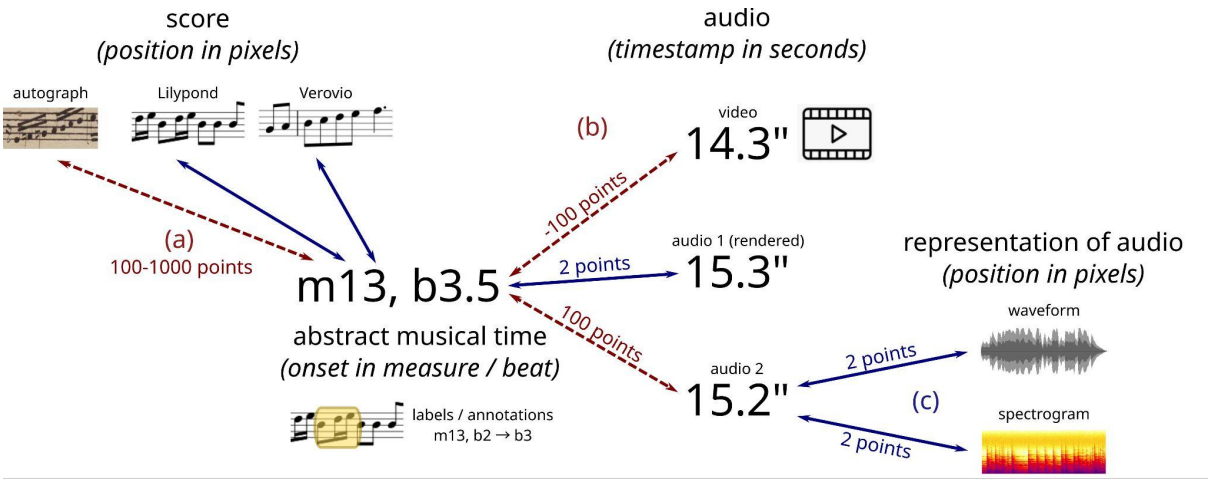

Figure 1. ConvUnits describe synchronizations between (a) image/musical time, (b) audio time/musical time, and (c) image/audio time. Plain arrows show ConvUnits that are either trivial (2 synchronization points) and/or given by the rendering software, whereas dashed arrows show ConvUnits that could be inferred through (a) OMR or (b) score/audio alignment, but that could also be edited/corrected within a synchronization editor.

[6] https://librosa.org

2. Synchronizing through Conversions between Units

We distinguish here *symbolic scores* (structured encoding such as MEI or other formats), *rendered scores* (graphical representation of the former, rendered by some rendering software), and *image scores* (scanned printed scores or autographs).

On computational music analysis applications, we would like that *musical time* (symbolic position in measure and beats, such as “measure 13, beat 3.5”) plays the *pivotal role*, both to browse synchronized scores and to annotate some of their elements. Such a symbolic musical time is relevant for most (but not all) of the notated Western music. An *annotation label* may thus be given as “measure 13, beat 2 to beat 3”, with additional staff and/or other data information. However, the musical time can not be easily extracted, neither from an image file representing a score, nor from an audio file.

But *sources* representing a different aspect of music, such as (symbolic representations of) scores, images, audio, or videos, can be *aligned*. We model the *conversion* between one unit of measurement to another one into an abstract object that we call **ConvUnit**. This enables both audio and image synchronization, allowing to refer to the musical time even when annotating or playing from other representations.

A ConvUnit is a *sorted list of synchronization points* targeting two sources. The ConvUnits encodes the following conversions (Figure 1):

- *image/musical time* (X(Y)-positions/onset, (a)): For rendered scores, this conversion may be provided by the rendering software. On scanned scores, it may be provided by OMR software and/or corrected or entered by human input. The current implementation of ConvUnits uses *X-positions* for score images, identifying positions in continuous single-staff music that may be horizontally scrollable.

- *audio time/musical time* (timestamp/onset, (b)): For rendered audio, this conversion is usually trivial: Even when there are tempo changes, the audio software can provide such data. For any other audio file, the conversion can be produced by audio/score alignment methods and/or corrected or entered by human input.

- *image/audio time* (X(Y)-positions/timestamp, (c)): Mapping the X-position of any linear representation of music -- waveform, spectrogram -- to the audio timestamp is most of the time a trivial linear conversion requiring only start/stop points. This is the only conversion needed for music applications that considers only an audio file. In this case the musical time is useless.

Focusing on *conversions between units* rather than on the *units* themselves ease flexible *chains of conversions*: Any value can be converted to any other, possibly using interpolations. As example, composing the (a) and (b) ConvUnits produce a combined X(Y)-positions/timestamp conversion for score following or other applications. We now use the ConvUnits shown in the Figure 1, but other ConvUnits could be added and the conversion could be generalized to any tree structure, or even to any graph structure provided preference rules for conversion paths.

3. A Canvas Framework for Synchronized Views

This section details how we implement features on a set of synchronized views through ConvUnits, with possibly annotation labels that can be moved/edited on different views.

3.1 Backends

Audio/Video backends. The audio sources are displayed either through the HTML5 `<audio>` tag, or through the `youtube-player` plugin. Both enable to get or to set the playback position that will then be converted through the ConvUnit audio time/musical time. Waveforms and spectrograms are generated with SoX[7].

Score backends. Two backends render scores along with the ConvUnit image/musical time. As an example, we report sizes on the madrigal Di nettare amoroso by *Di nettare amoroso ebro* by Luca Marenzio (uncompressed MEI of 1.4 MB, 1600+ notes in 133 measures).

- **Verovio** (Pugin 2014; Sapp 2017). Starting from MEI files (that can be converted from other files), the backend outputs either a rasterized `.png` files or a vector `.svg` canvas (Figure 3, bottom). The former is more efficient for large scores (743 KB), and the latter is a bit larger (1.8 MB) but enables arbitrary zoom. Moreover, on the `.svg`, events can be attached to notes and support additional interaction that could provide a ConvUnit mechanism and way more. This increases again the complexity of the DOM: We found that the most efficient way to handle large scores (including fast scrolling through entire scores), even with the `.svg`, was to keep a static ConvUnit list (here 66 KB, uncompressed, with 562 different offsets).
- **Lilypond**[8]. Lilypond `.ly` files were first generated with music21, but the support of Lilypond is not optimal in music21[9]. They can also be generated through `MusicXML2ly`, after conversion into `XML` by music21. The ConvUnit list is obtained through patching the `.ly` file in order to output the position of every note and bar.

3.2 Synchronizing Views on a Behavior-based UI

The front interface is based on the Vue.js 2.6 framework[10] with Vuetify material design UI elements[11].

Behaviors. Complex graphs tend to have ridiculously large code space. Because any single action such as a click can have dozens of different effects depending on where and when it happens. Instead of segmenting the codebase on events, we segment the codebase as a *list of features*, which can all be toggled depending on the context. The user interface (UI) is

[7] http://sox.sourceforge.net
[8] https://www.lilypond.org
[9] https://github.com/cuthbertLab/music21/issues/433
[10] https://vuejs.org
[11] https://vuetifyjs.com

thus based on *renderless behavior slots*[12]: Components expose behaviors that can be inherited, customized, or detailed.

Behaviors rarely are linked to a UI element, because they live and die in the canvas. A behavior is rather a link between an event (a click, another behavior finishing its job, etc.), and an action (deleting a label, moving a label, etc.). Some behaviors further display an element in the canvas, such as the music cursor. In that case, the event is "audio player updating its timestamp", and the action is "update the cursor in the canvas". Developing with such behaviors improves code reusability and portability. One defines several (possibly inherited) behaviors enabled in different situations, such as desktop/mobile usage, simple/advanced modes, or active/inactive depending on the availability of audio or the current selection. This is very close to how humans think about a codebase, a list of features to be implemented, and sometimes removed.

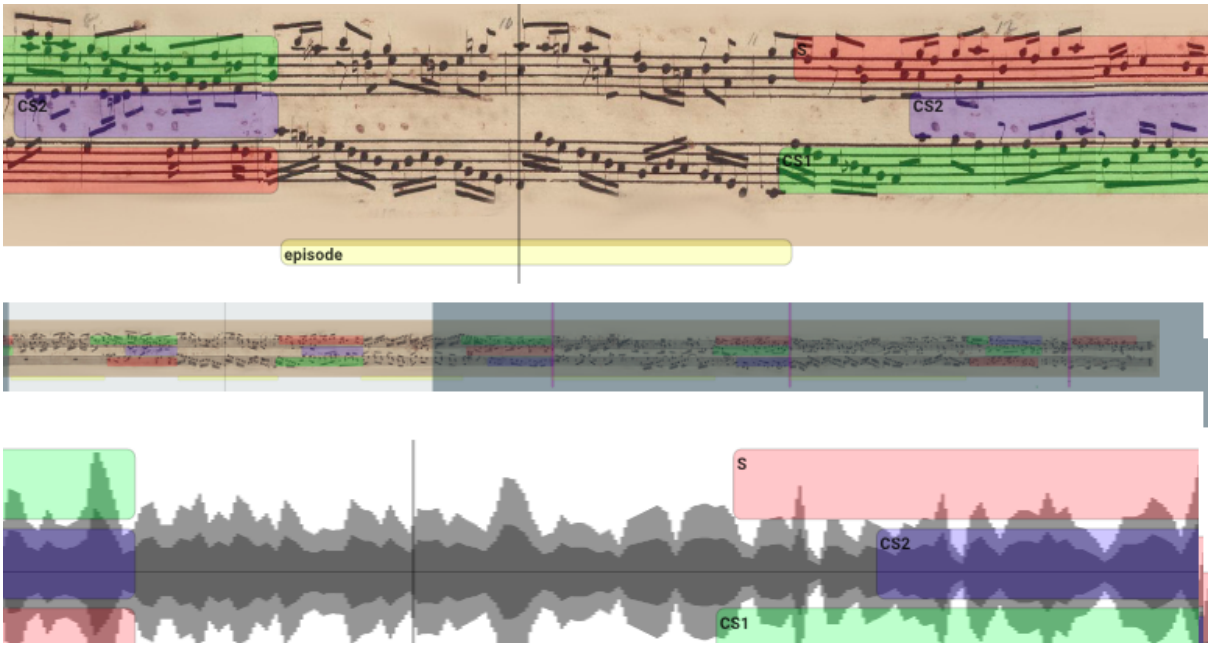

Figure 2. Autograph of the fugue in C Minor BWV 847 by J.-S. Bach (D-B Mus.ms. Bach P 202) synchronized with a performance by Kimiko Ishizaka (bottom, http://welltemperedclavier.org) as well as fugue annotation labels from (Giraud 2015).

Views. Each source is represented on a view, possibly with labels. The playback cursor is always synchronized between views with the ConvUnits. The synchronizations can be further used in two ways:

- The common usage of such an application is to *keep all the views synchronized*. Scrolling any representation of the music, playing the music, or creating or modifying a label on any view updates the other views and their labels, using the ConvUnits. This enables both playback applications while browsing various sources (rendered score, autograph, ...) as well as analysis applications (Figure 2).

[12] https://www.digitalocean.com/community/tutorials/vuejs-renderless-behavior-slots

- Conversely, to *edit* the synchronization, one typically needs to update the synchronization data on one source while freezing it on the other. We propose a synchronization editor that displays the ConvUnit list. The editor is currently implemented for the audio/musical time synchronization (Figure 3). In the *edit mode*, any synchronization point can be added or updated, either on the representation of audio or on the score. In the *tap mode*, one can tap points at regular intervals such as measures. The modes can be mixed: The user can thus synchronize a score starting with two start/stop points, adding manually a few points when the tempo changes, tap a few sections where the synchronization is more elaborated, and finally correct some of these points as needed.

The components offer import and export features for analysis labels but also for the synchronization itself.

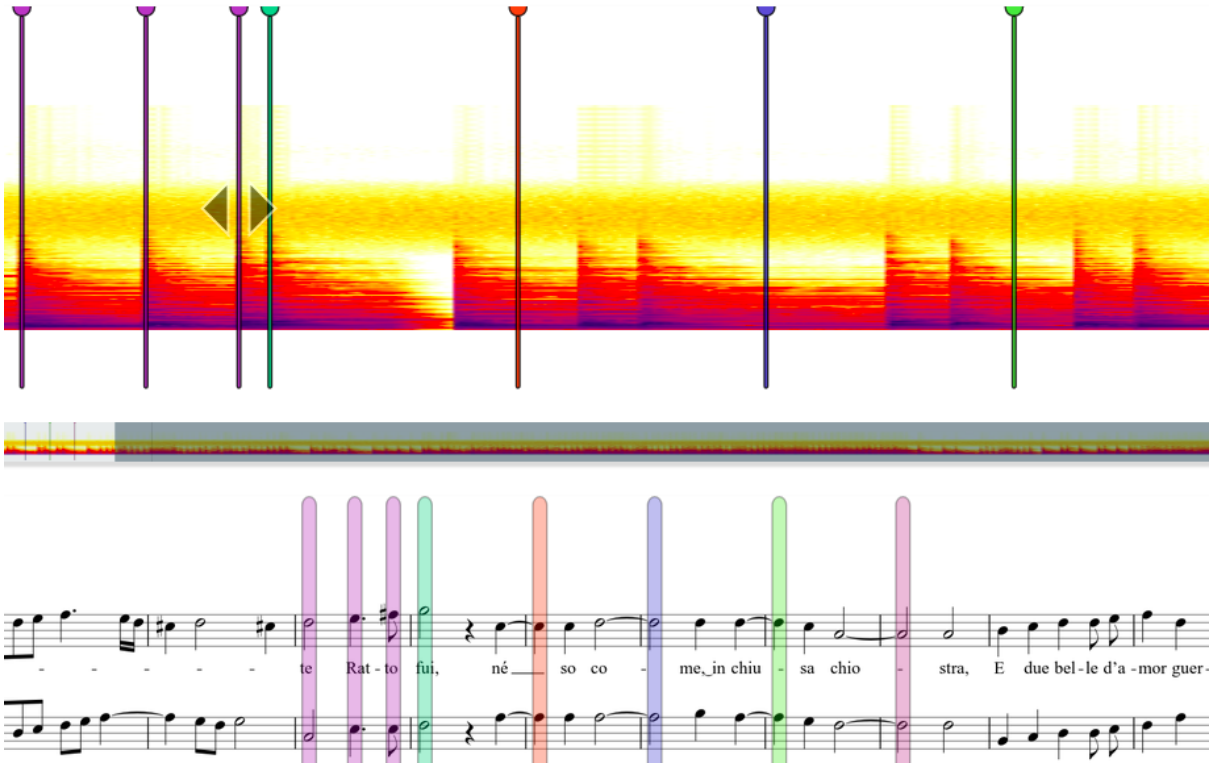

Figure 3. Rendering with Verovio the madrigal *Di nettare amoroso ebro* by Luca Marenzio (1587), editing the synchronization against the spectrogram of an audio file. The colors help to identify the corresponding synchronization points. Regular points (such as here on bars with syncopations) are added in the tap mode. The synchronization is refined with the edit mode, for example on slow-down before cadences – the user is here updating the point on the last eighth note before the end of a phrase (Rat-*to* fui).

4. Applications and Discussion

The Dezrann platform and the prototype synchronization editor are available at www.algomus.fr/dezrann, and the code is released under the GPLv3+ license. We discuss here the usages of these software, focusing on the added value of synchronization between different sources.

The Dezrann platform is used for annotation of corpora for ***research in musicology and computational musical analysis*** – whether in support of manual annotation, in visualization of automatic or semi-automatic analysis results and in the evaluation of these results. The platform was used in the encoding of corpora including patterns, rhythms, harmonies, texture and form: Such corpora distributed total to more than 3000 labels and are openly distributed on www.algomus.fr/data and through Dezrann. Some music analysts work only on scores, and may not benefit from other synchronized sources. Conversely, the presentations of other sources and the ability to edit the synchronizations is interesting for music historians or researchers on interpretations. Research (and practice!) of Folk music is also mostly based on field recordings that also benefit from synchronization capabilities.

Music pedagogy seeks the engagement of children in active listening (Vitale 2021). Dezrann is also thus intended for ***education*** in music and computer music, whether in specialized courses in music schools (such as harmony or analysis) or in music education in primary or secondary schools. Non-sight-readers, or people learning or improving their skills in music (sight-)reading, benefit from switching between scores and audio/video files – but the synchronization should be prepared by the teachers.

In 2021/2022, a pilot program gathers classrooms in five *secondary schools* (ages 10-16) (Sauda 2022). Each classroom hosts between four and six 1-hour sessions on the platform, partly helped by a mediator conveying music (and computer music!) discourse to the children. Most of these children do not have any specialized music education – and do not read scores at all. For them, the audio (or the video) is the most informative source, or even the only one. In the first session, the mediator introduces a piece, chosen by the teacher from a catalog of a few pieces, and uses the platform to hear and annotate music elements. Note that the focus is not on the platform – but rather on actual music concepts on which the children are working, the platform being as "transparent" as possible. In the following sessions, the teacher freely selects any piece. The mediator or the teacher led these sessions, aiming at improving both children's autonomy while listening to music – alone or in groups of two or three – and also to share and discuss their work.

We attended seven of these sessions, observing both children and teacher usage. Children manage both to be autonomous in their analysis and are eager to report and share their analysis with others, thus improving their engagement (Sauda 2022). During the sessions, children themselves proposed new usages, including the possibility that each of them chooses a piece to analyze and presents it to the classroom with their analysis, thus

improving their engagement again. We also organized two feedback meetings with teachers. Following these reports, ergonomy was improved, bugs were corrected, and functionalities were added or scheduled.

Finally, through the synchronization component, the platform allows future usage in ***collaborative score correction***. The CollabScore project will use some of the techniques presented here to build a cross-modal demonstrator: Given a symbolic score obtained from a (possibly faulty) OMR process, synchronize both the obtained encoding (MEI) and the original image to allow for crowd-sourced corrections.

Modeling and displaying synchronizations between representations of music – as well as being able to update these cross-modal synchronizations – help thus research, education, and practice of music. Perspectives include extending the synchronization components for audio/video, allowing to flexibly edit the synchronization on score images – with also bounding boxes based on (possibly partial or faulty) OMR results. For this, future implementations of ConvUnits will target both rational numbers and also non-numeric values to better model repeats, and extend the handling of Y-positions for bounding boxes.

Acknowledgements

We thank all users of the Dezrann platform for their usage and their feedback. We thank the Sciences Infusent program of the Université de Lille for their support in the pilot program with music education classrooms. We also wish to thank the Mésocentre de Lille for their computing resources and the hosting of the platform, and we gratefully acknowledge support from the ANR CollabScore ANR-20-CE27-0014 project and by the CPER MAuVE (ERDF, Hauts-de-France).

References

Balke, S., Dittmar, C., Abeßer, J., Frieler, K., Pfleiderer, M., & Müller, M. (2018). Bridging the gap: Enriching YouTube videos with jazz music annotations. *Frontiers in Digital Humanities* , 5:1.

Bent, I. & Drabkin, W. (1987). *Analysis*. W. W. Norton & Company.

Burghardt, M. (2017). Allegro: User-centered design of a tool for the crowdsourced transcription of handwritten music scores. In *Digital Access to Textual Cultural Heritage conference (DATeCH 2017)*.

Calvo-Zaragoza, J., Jr., J. H., & Pacha, A. (2020). Understanding optical music recognition. *ACM Computing Surveys*, 53(4).

Caplin, W. E., Hepokoski, J., & Webster, J. (2009). *Musical Form, Forms & Formenlehre – Three Methodological Reflections*. Pieter Bergé, Leuven University Press.

Carey, B. & Hajdu, G. (2016). NetScore : An image server/client package for transmitting notated music to browser and virtual reality interfaces. In *International Conference on Technologies for Music Notation and Representation (TENOR 2016)*.

Couprie, P. (2008). iAnalyse : un logiciel d'aide à l'analyse musicale. In *Journées d'Informatique Musicale (JIM 2008)*, pages 115–121.

deGroot Maggetti, J., de Reuse, T. R., Feisthauer, L., Howes, S., Ju, Y., Kokubu, S., Margot, S., Nápoles López, N., & Upham, F. (2020). Data quality matters: Iterative corrections on a corpus of Mendelssohn string quartets and implications for MIR analysis. In *International Society for Music Information Retrieval Conference (ISMIR 2020)*, pages 432–438.

Dorfer, M., Henkel, F., & Widmer, G. (2018). Learning to listen, read, and follow: Score following as a reinforcement learning game. In *International Society for Music Information Retrieval Conference (ISMIR 2018).*

Ewert, S., Muller, M., & Grosche, P. (2009). High resolution audio synchronization using chroma onset features. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2009)*, pages 1869–1872.

Giraud, M., Groult, R., Leguy, E. & Levé, F. (2015). Computational fugue analysis. Computer Music Journal, 39(2).

Giraud, M. (2018). Using Dezrann in musicology and MIR research. Digital Libraries for Musicology (DLfM 2018).

Giraud, M., Groult, R., & Leguy, E. (2018). Dezrann, a web framework to share music analysis. In *International Conference on Technologies for Music Notation and Representation (TENOR 2018* , pages 104–110.

Hoos, H. H., Hamel, K. A., Renz, K., & Kilian, J. (1998). The GUIDO music notation format. In *International Computer Music Conference (ICMC 1998)*, pages 451–454.

Kosta, K., Bandtlow, O. F., & Chew, E. (2018). Mazurkabl: score-aligned loudness, beat, expressive markings data for 2000 Chopin mazurka recordings. In *International Conference on Technologies for Music Notation and Representation (TENOR 2018)* , pages 85–94.

Lepetit-Aimon, G., Fober, D., Orlarey, Y., & Letz, S. (2016). INS core expressions to compose symbolic scores. In *International Conference on Technologies for Music Notation and Representation (TENOR 2016)*.

Mueller, M., Arzt, A., Balke, S., Dorfer, M., & Widmer, G. (2019). Cross-modal music retrieval and applications: An overview of key methodologies. IEEE Signal Processing Magazine, 36(1):52–62.

Nattiez, J.-J. (1975). *Fondements d'une sémiologie de la musique*. Dufrenne.

Pugin, L., Zitellini, R., & Roland, P. (2014). Verovio: A library for engraving MEI music notation into SVG. In *International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 107–112.

Sapp, C. S. (2017). Verovio Humdrum Viewer. In *Music Encoding Conference (MEC 2017)*.

Sauda, A., Giraud, M., & Leguy, E. (2022). Soutenir en classe l'écoute active, l'autonomie et l'échange en analyse musicale avec la plateforme web Dezrann. Upcoming, in *Journées d'Informatique Musicale (JIM 2022)*.

Shan, M. & Tsai, T. (2020). Improved handling of repeats and jumps in audio-sheet image synchronization. In *International Society for Music Information Retrieval Conference (ISMIR 2020).*

Thickstun, J., Brennan, J., & Verma, H. (2020). Rethinking evaluation methodology for audio-to-score alignment. arXiv preprint arXiv:2009.14374

Vitale, J. L. (2021). The Importance of active listening and reflection in the music classroom The Canadian Music Educator, 62(2), pages 38–42.

Weiß, C., Zalkow, F., Arifi-Mü ller, V., Mü ller, M., Koops, H. V., Volk, A., & Grohganz, H. G. (2021). Schubert Winterreise Dataset: A multimodal scenario for music analysis. *Journal on Computing and Cultural Heritage*, 14(2).