Computer-Assisted Measure Detection in a Music Score-Following Application

Eran Egozy
Massachusetts Institute of Technology
Cambridge, MA, United States
egozy@mit.edu

Ian Clester
Georgia Institute of Technology
Atlanta, GA, United States
ijc@gatech.edu

***Abstract*—ConcertCue is a web application that synchronizes content delivery with live music. During a concert, a human operator supervises a score-following process to ensure it stays in sync. To aid the operator, the system displays a scanned copy of the score and highlights the current estimated measure, which requires the system to know the visual location of each measure on each page.**

In this paper, we present a fast, accurate OMR algorithm for identifying systems and barlines in a musical score and an integrated user interface that allows for quick OMR verification and correction. The use of this system as a part of ConcertCue has accumulated over 1,000 pages of human-verified annotations, which we plan to release as a dataset in the near future.

***Index Terms*—Real-time Score Following, Optical Music Recognition, Measure Detection, User Interface**

I. Introduction and Related Work

Score following is the task of continuously estimating the position of a live musical performance relative to a predetermined reference track. The musical performance is typically captured as a live audio signal, while the reference track might be a preexisting audio recording of the same music [1] or a symbolic representation such as a MIDI file or digital score [2]. Applications of score following include automatic accompaniment [3], automatic page turning [4], and synchronized content delivery to audiences during live music [5].

ConcertCue,[1] a web application that we have developed over the past several years, is an example of the latter. A timeline of events (snippets of content including text, images, and animations) is authored to align with important musical moments in a particular piece. When the piece is performed, these events are displayed on target devices such as personal mobile devices or screens in the concert hall at exactly the right musical times. While this real-time alignment may happen automatically, in practice, a musically-literate human operator must be present during the performance to ensure that the system is working properly and to correct any synchronization errors that might occur.

As shown in Fig. 1, the ideal environment for the operator is an application that displays the music score of the piece being performed while also indicating the estimated real-time position of the live music overlaid on top of the score. In this way, the operator has a full understanding of "where the system is" relative to the music and can make adjustments as needed via a suitable user interface.

[1]See https://concertcue.org and https://concertcue.com

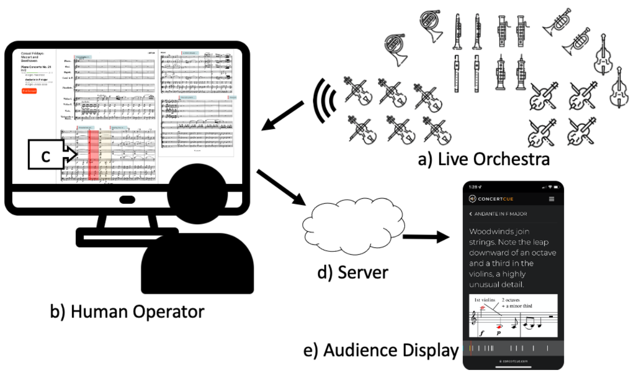

a) Live Orchestra
b) Human Operator
c
d) Server
e) Audience Display
Woodwinds join strings. Note the leap downward of an octave and a third in the violins, a highly unusual detail.

Fig. 1. ConcertCue System Overview. Audio of a performing orchestra (a) is sent to a human operator (b) who ensures that score following behaves correctly. A score is displayed with the current measure highlighted in red (c), which can be adjusted by the operator as needed. Timing data is streamed to a server (d) which in turn synchronizes content on the audience display device (e) during the performance.

To enable this functionality, a digitized score of the piece must be properly annotated so that the application can create a mapping between musical time (the measure and beat of the music) and its 2D location in the digitized score. Optical Music Recognition (OMR) makes this annotation task feasible and much faster than manual annotation. While the full complement of OMR tasks includes a complete understanding of the music score (such as notes, rests, dynamic markings, and articulations [6]), we are only interested in the early stages of an OMR system: identifying the locations of the systems of a score and the measures within each system (such as [7]).

II. Implementation

An overarching goal in implementing ConcertCue is to develop powerful and easy-to-use tools for content creators. As such, the entire platform is a web-based client-server app with all its inherent advantages (e.g., works anywhere on any computer browser, zero installation, easy deployment and software updates). Much work went into user interface (UI) and user experience design in the two primary usage domains: pre-concert content authoring, and real-time operation. In both cases, the most important UI element is the digital representation of the score and its associated annotations.

A. *Score Annotation Tool*

Fig. 2 shows the score annotation tool. A user begins by uploading a PDF of the entire score for a piece of music. The server initiates an automatic OMR process that estimates the locations of all system bars on every page. An overview of the algorithm is described below in section II-B.

Processing a single page may take anywhere from 2-20 seconds, largely depending on the CPU resources available to the server. As soon as the first page has been processed, it is displayed in the editor so that the user can verify the system and measure positions and make corrections if necessary. Thus, the user can immediately start working while the server continues to process subsequent pages.

Our UI supports multiple systems per page and assumes that all barlines belonging to a single system share the same y coordinates.[2] This enables the user to easily add, delete, or move a barline by dragging it to adjust its x coordinate, rather than specifying a rectangle for each measure separately.

Ideally, the OMR process correctly identifies all systems of a page and the barlines within each system. To aid the user in quickly assessing the automated estimation process, we employ a pseudo-random coloring scheme for displaying measures. Measure colors are chosen such that adjacent measures are tinted with highly contrasting hues. Barlines are displayed as blue lines (see Fig. 2). Both measure tinting and barlines can be toggled on or off. Typically, a page can be verified by a musician within a few seconds.

If the OMR process incorrectly identifies the systems of a page (for example, by accidentally combining two separate systems into one), the barlines of these incorrect groupings will likely be incorrect as well. At this point, the user can correct an erroneous system's y coordinates using the drag handles and then initiate a partial OMR task that only re-estimates barlines within the selected system. This feature makes quick work of fixing an early error (system misidentification) which would otherwise render the whole page incorrect. After correcting the system annotations, barlines can be re-estimated, producing the correct result much more quickly than by manually correcting every barline.

In addition, the user can easily number the measures of a score by setting a number for a particular measure. The editor automatically numbers the following measures sequentially until the end of the piece, or until a new measure number is entered. This case is useful, for example, if a multi-movement score begins renumbering measures at the start of each movement.

B. *OMR Implementation*

Our OMR implementation uses a statistical image processing approach based on assessing the probability of long,

[2]While this assumption may not be strictly true due to page skew or other scanning artifacts, it works well enough for this application and simplifies the UI, as barline y coordinates are inferred from the containing system.

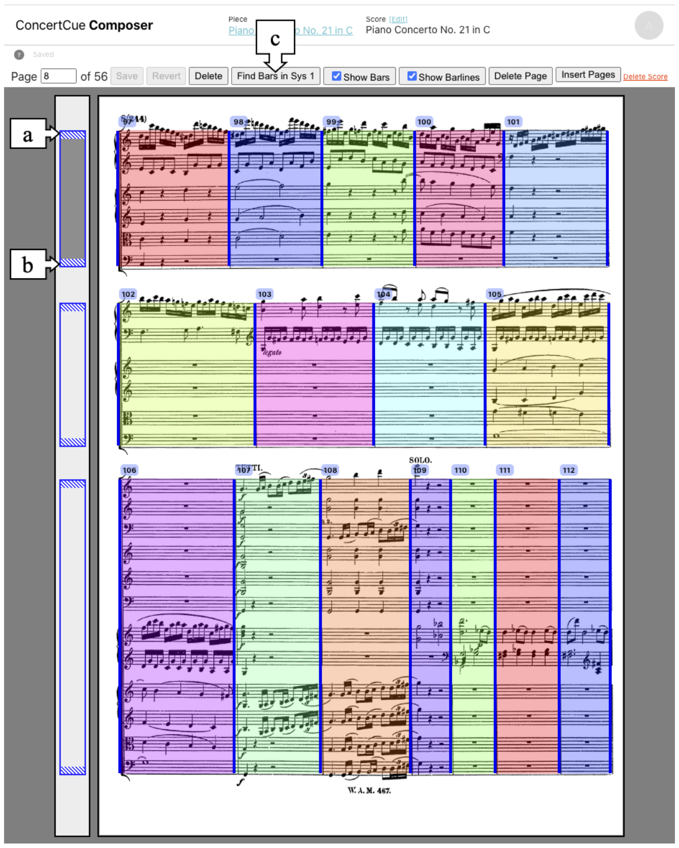


Fig. 2. Screenshot of the score annotation tool. Measure tinting is guaranteed to have high visual contrast between adjacent measures. Barlines are overlaid as blue lines. Shown above, the first system is selected. The system grab handles (a, b) let the user modify a system's vertical bounds, which also dictate the vertical extents of the barlines. A single button click (c) re-estimates barlines of the currently selected system, or of the entire page if no system is selected.

mostly-continuous horizontal lines (staff lines) at a given y coordinate and of mostly-continuous vertical lines (barlines) at a given x coordinate. This approach is similar to the strategy taken by [8] though it was developed independently. The algorithm happens in three phases.

First, the full image is rotated so as to maximize the likelihood of perfectly horizontal and vertical lines. This process removes whatever slight rotation may exist if the original score was scanned by hand. Removing rotation greatly improves accuracy in the rest of the process.

Next, staff lines are detected. The image is processed with a median filter to enhance horizontal lines and then averaged across the x-axis to produce a staff-line likelihood curve, where peaks correspond to staff line locations along the y-axis. See Fig. 3. Then, intra-staff-line spacing is measured. Small gaps are likely to be spaces between lines of a single staff while large gaps are likely to be spaces between distinct staves. Once staves are identified, they are grouped into systems. In a typical multi-system score, the left-most edge of the score has a vertical line connecting staves into a single system. Therefore, undisturbed white space between staves usually

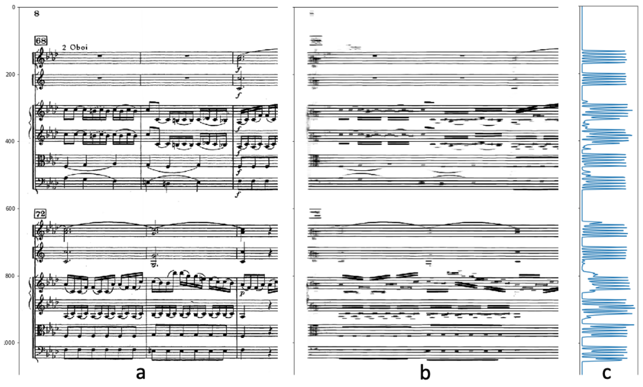


Fig. 3. Staff line identification. The original image (a) is passed through a median filter (b). Averaging across the filtered image produces a staff line likelihood curve (c). Peaks in this curve correspond to staff line locations.

indicates the beginning of a new system.

Finally, each system is analyzed for the likelihood of barlines within its vertical bounds. Since all barlines of a single system are vertically aligned, they tend to reinforce each other. A similar median-filtering approach along the y-axis is used to create a barline likelihood curve. Once the x-coordinates of barlines are estimated, they are used in combination with the system bounds to generate a listing of system measures.

C. Other Tools

The rest of the application comprises basic content management (e.g., creating concerts, pieces, movements), and other editors needed to complete data entry. For example, the cue editor is used to attach cues (HTML snippets with text and images) to a location in the score. A score location is defined as a measure number and a fractional position within that measure in the range $[0, 1]$. The cue editor is shown in Fig. 4. Cues are easily added, modified, or deleted from the score with mouse clicks and drags.

During a live performance, the score is displayed and its pages are automatically scrolled into view based on the estimated current location in the piece. The current measure is highlighted in red (see Fig. 1) using the annotations created by the user (section II-A) with the aid of our OMR algorithm (section II-B). The operator monitors the progress of the estimated location and make manual adjustments as needed.

III. Evaluation

The measure detection task was evaluated against a corpus of fourteen typeset scores totalling 1,117 pages of music. Eleven scores were scanned from printed copies, while the other three were exported from engraving software and thus have no scanning artifacts. There are no hand-written scores. This ground-truth set was created and human-verified using the score annotation tool described above. The data for system barlines are stored as normalized page coordinates. In other words, x-coordinates are in the range $[0, 1]$ relative to page width, and y-coordinates are similarly relative to page height.

TABLE I
Barline estimation results. Type S is hand-scanned score. Type E is exported directly from engraving software

Score	Type	Pages	Barlines	F-Score
Beethoven Rondino	S	8	142	0.996
Adams Doctor Atomic	S	97	826	0.987
Debussy La Mer	S	137	845	0.987
Debussy Nocturnes	S	120	666	0.977
Haydn 49	S	23	503	0.983
Liszt Prometheus	S	66	516	0.976
Maskats My River	E	66	387	0.996
Mozart Piano Concerto 24	S	64	1049	0.980
Prokofiev Violin Concerto 2	S	85	981	0.931
Ruehr Cosmic Cowboy	E	41	320	1.000
Ruehr Lucy	E	42	406	1.000
Schumann 3	S	67	1290	1.000
Thomas Prayer Bells	S	44	234	0.998
R-Korsakov Scheherazade	S	257	1888	0.952
Total		1117	10053	0.976

Our evaluation considers an estimated barline to match a ground-truth barline if all normalized coordinates fall within a threshold of 0.01. This metric produces an F-score of 0.976 for the entire corpus. See Table I for details.

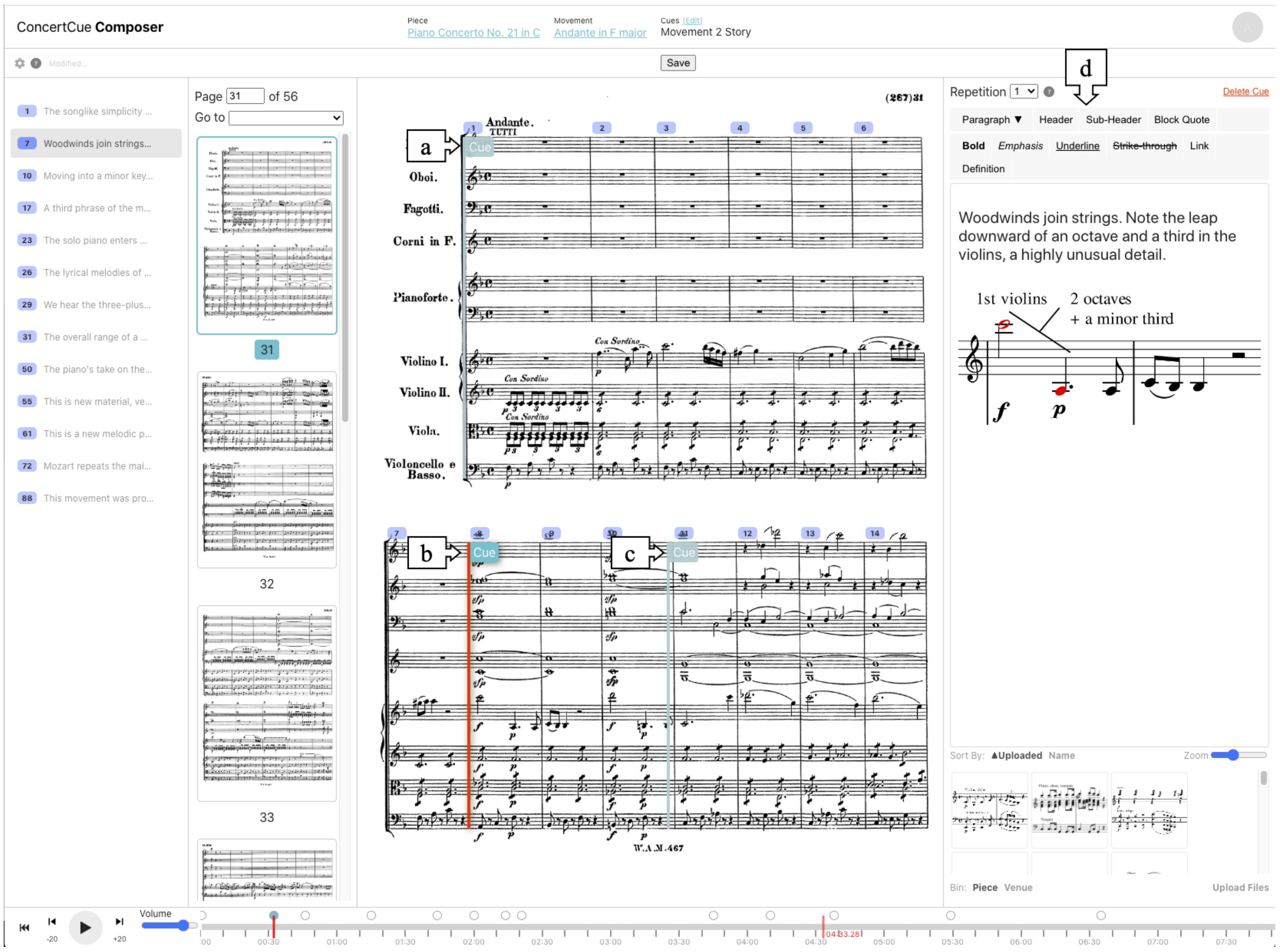
IV. Discussion and Future Directions

The innovation of displaying the music score with overlaid annotations seems simple enough, but it has proven to be incredibly effective for running live concerts. We recently used ConcertCue to display supertitles in opera productions. Typical production workflows without ConcertCue involve an operator looking at a printed score on a desk while also operating a computer with a PowerPoint-style presentation to advance the displayed text at the right musical time. This method is awkward and stressful. Using ConcertCue, the operator looks only at the computer screen and sees all pertinent information in one place.

We have had several users (staff at symphony orchestras and regional opera houses) use the full system to upload scores and verify and correct OMR results. Universally, the sentiment is that the score annotation tools are easy to use, even in cases with long (100+ page) scores. Some even report the task being fun. We believe this is due to the OMR system performing quite accurately in the vast majority of cases, and the visualization and editing tools being user-friendly.

As we continue to use ConcertCue, we will accumulate more human-verified scores with system and measure annotations and hope to release this growing dataset in the near future. Importantly, the majority of these scores are orchestral scores, whereas a number of existing datasets tend to focus on piano pieces or non-orchestral works.

We plan to continue refining the OMR algorithm and report results against this and other datasets (see [7]). An additional area of interest is to automate the measure numbering system using numeric OMR, specifically targeting measure numbers, as motivated by [9].



Woodwinds join strings. Note the leap downward of an octave and a third in the violins, a highly unusual detail.

Fig. 4. Screenshot of the cue editor, showing three cues attached to various locations in the score (a, b, c), with (b) currently selected. Cue content is authored in (d).

Acknowledgment

The authors would like to thank Nathan Gutierrez for his contributions to ConcertCue design and implementation.

References

[1] S. Dixon, "Live tracking of musical performances using on-line time warping," in *Proceedings of the 8th International Conference on Digital Audio Effects*, vol. 92, 2005, p. 97.

[2] A. Cont, "A coupled duration-focused architecture for real-time music-to-score alignment," *IEEE transactions on pattern analysis and machine intelligence*, vol. 32, no. 6, pp. 974–987, 2009.

[3] A. Cont, J. Echeveste, J.-L. Giavitto, and F. Jacquemard, "Correct automatic accompaniment despite machine listening or human errors in antescofo," in *ICMC 2012-International Computer Music Conference*, 2012.

[4] A. Arzt, G. Widmer, and S. Dixon, "Automatic page turning for musicians via real-time machine listening," in *ECAI 2008*. IOS Press, 2008, pp. 241–245.

[5] M. Prockup, D. Grunberg, A. Hrybyk, and Y. E. Kim, "Orchestral performance companion: Using real-time audio to score alignment," *IEEE MultiMedia*, vol. 20, no. 2, pp. 52–60, 2013.

[6] J. Calvo-Zaragoza, J. H. Jr, and A. Pacha, "Understanding optical music recognition," *ACM Computing Surveys (CSUR)*, vol. 53, no. 4, pp. 1–35, 2020.

[7] M. Kletz and A. Pacha, "Detecting staves and measures in music scores with deep learning," in *Proceedings of the 3rd International Workshop on Reading Music Systems*, J. Calvo-Zaragoza and A. Pacha, Eds., Alicante, Spain, 2021, pp. 8–12. [Online]. Available: https://sites.google.com/view/worms2021/proceedings

[8] F. Zalkow, A. V. Corrales, T. Tsai, V. Arifi-Müller, and M. Müller, "Tools for semi-automatic bounding box annotation of musical measures in sheet music," in *Demos and Late Breaking News of the International Society for Music Information Retrieval Conference (ISMIR)*, 2019.

[9] A. Pacha, "The challenge of reconstructing digits in music scores," in *Proceedings of the 3rd International Workshop on Reading Music Systems*, J. Calvo-Zaragoza and A. Pacha, Eds., Alicante, Spain, 2021, pp. 4–7. [Online]. Available: https://sites.google.com/view/worms2021/proceedings