Chapter 8

Applications

Using the methods presented in the previous chapters of this thesis, we are able to calculate temporal alignments for given scanned sheet music documents and corresponding audio recordings. Furthermore, we are able to identify all documents that correspond to the same piece of music. Since the proposed methods are mostly automatic and require only little manual interaction, alignments and links between the documents can be obtained for large collections of music documents. In this chapter, we present applications and user interfaces for presenting the music documents and explicitly making use of the available temporal alignments and links. A main concept in the presentation of the documents to the user is that only the high quality input documents are used for presentation. All the intermediate data, especially the error-prone symbolic score data obtained via OMR, are never directly visible to the end-user. The target use case of the presented applications is that of a digital music library where users can access the digitized collection of sheet music scans and audio recordings. This scenario is being pursued as part of the PROBADO project, see also Section 4.1. A prototypical user interface has been designed and implemented. This user interface and the functionalities it offers are presented in the individual sections of this chapter. Beyond that, an introductory demo video of the interface can be found on the website of the author. Section 8.1 introduces the score viewer interface for displaying and interacting with the scanned sheet music documents. In Section 8.2, we present the audio viewer interface for visualizing and interacting audio CD collections. Both interfaces can then be used for presentation of and navigation in the content across the domains of sheet music and audio, which is described in Section 8.3. If more than one audio recording is available for a single piece of music, the user can switch between the available interpretations as described in Section 8.4. Finally, the interface can be used to perform content-based retrieval across the domains, which is presented in Section 8.5.
8.1 Score Viewer

The score viewer displays sheet music in form of a 3D virtual book, see Figure 8.1. The advantage of this style of display is that it resembles the original look and feel of the sheet music book that is represented by the given individual scans for each page. This makes interacting with the content more natural and intuitive for the user. The view on the book can be zoomed and scrolled freely and the tilt angle can be adjusted to suit the viewers preference. The view settings selected in Figure 8.2 show a more close-up view of the right page shown in Figure 8.1. A translucent yellow box is used to indicate the current playback position in score book on a bar-wise level. The features related to playback are discussed in more detail in Section 8.3. The user can navigate freely through the whole sheet music book. To this end, pages can be turned by clicking the left or right borders of the book using the mouse pointer. A bookmark at the side of the book (upper right side in Figure 8.1) always points to the current playback position. If the user has navigated away from the current playback position, clicking this bookmark causes the viewer to instantly go back to the page of the current playback position.

At the top of the score viewer, a title and other meta information about the current score track are displayed. Control fields for the track number, page number, and the performance bar number in the track are shown at the bottom. In addition to displaying the current playback position in the book, these controls can be used to navigate to the previous or next track, page, or bar, or to enter a specific track number, page number, or bar number using the computer keyboard. A click on the icon at the top right opens the so-called track start browser, which is depicted in Figure 8.3. Similar to the table of contents shown in Figure 6.7, the track start browser shows a list of all score tracks represented by their title and their

\[\text{http://www.iai.uni-bonn.de/~fremerey/}\]
8.1. SCORE VIEWER

Figure 8.2. The view on the sheet music book displayed in the score viewer can be zoomed and scrolled freely.

Figure 8.3. The track start browser allows quick access to all the score tracks contained in the sheet music book.
first grand staff in the score. Note that using the spatial information obtained from the OMR results, the image data for these excerpts are extracted from the scans automatically. Using the track start browser, the user can quickly navigate to any of the score tracks contained in the sheet music book. Additional controls for playback and interpretation are found in the top left and bottom right corners. These will be explained in the subsequent sections of this chapter.

8.2 Audio Viewer

The audio viewer employs a similar design as the score viewer to allow the interaction with audio CD collections, see Figure 8.4. A virtual CD case is displayed at the upper left part to give the user the feeling of interacting with a real physical medium. The case can be magnified and turned around to display the backside of the cover. A list of CD tracks showing a title and duration is displayed on the right side together with a selector for the disk number. Using these controls, the user can browse through all disks and tracks that are part of the collection.

The acoustic content of the current audio track is visualized in the center part of the audio viewer. The visualization shows a time-frequency representation of the acoustic content. Time is represented on the horizontal axis and frequency is represented on the vertical axis. The frequency axis uses a logarithmic scale, which means that octaves are equally spaced. A vertical line with fixed position at the center of the visualization area indicates the current playback position. Accordingly, during playback, the visualized content moves from right to left. A spot with bright color in the visualization means that the corresponding frequency is present with high amplitude at the corresponding time. Areas with dark color indicate that the corresponding frequencies are present with only low amplitude or not present at all. This visualization is computed using a short-time Fourier transform in combination with binning and thresholding strategies. The result looks somewhat similar to a piano-roll display, which
8.3 CROSS-DOMAIN PRESENTATION AND NAVIGATION

Figure 8.5. Illustration of three different zoom levels for the visualization of audio content used in the audio viewer. A vertical line at the center of the visualization marks the current playback position. (a) The regular zoom level gives an overview of about one minute of audio content. (b) Same playback position as in (a) but zoomed in, so that only a few seconds of audio content are displayed. (c) Same playback position as in (a), but with maximum zoom setting. The horizontal lines seen in the visualization can be interpreted as a picture of the instantaneous frequency content.

is often used to visualize MIDI data, see for example [86, 76]. However, in contrast to a piano-roll display, a single long note that is played in the performance is usually displayed as multiple horizontal lines, because in this rather simple implementation harmonics of the fundamental frequency are not suppressed as in more sophisticated approaches like [127]. The time axis of the visualization can be scaled to different zoom levels by the user. In the regular setting shown in Figure 8.5(a), the visualization gives an overview of about one minute of the acoustic content surrounding the current playback position. In the maximum zoom setting, which can be accessed by clicking the double plus symbol in the viewer, the visualization corresponds to the instantaneous frequency content at the playback position, see Figure 8.5(c).

Controls at the bottom section of the audio viewer offer a similar functionality than those used in the score viewer. Here, instead of score track, page, and bar number, the controls allow access to the audio track, disk number, and track number on the disk. In both the score viewer and the audio viewer, controls at the bottom right can be used to start and stop the playback.

8.3 Cross-Domain Presentation and Navigation

A key concept of the presentation of the data to the user is that the score viewer and audio viewer are linked together through the sheet music-audio synchronization. Instead of acting
Figure 8.6. Illustration of simultaneous presentation of musical content in the score viewer and audio viewer. The score viewer (left) shows the current playback position as a yellow overlay box over the current bar in the sheet music book (near the top right corner of the book). The audio viewer (right) shows the corresponding position in the visualization of the acoustic content and on a slider representing the timeline of the complete track. While playback time advances, the content in the visualization moves towards the left, and the highlighted bar in the sheet music book progresses throughout the book. Pages in the book are turned automatically.

as just a viewer for sheet music and a player for audio recordings separately, the linked combination of the two appears more like two different windows of presentation of and interaction with the same piece of music. Since a single piece of music is presented and can be accessed in two different domains, i.e., the sheet music domain and the audio domain, we use the attribute cross-domain to describe the applications that emerge from the combination of the two.

A typical first application is cross-domain presentation, which, in our case, means letting the user follow the score while simultaneously listening to the audio playback. Since the software is aware of the current playback position in the sheet music, it can assist the user in following the performance by continuously updating the highlighted bar in the sheet music book according to the playback position, see Figure 8.6. The score viewer offers a special mode tagged “score following”. If this mode is active, the viewer automatically controls page turning and scrolling to keep the currently played bar visible in the sheet music book. The score following mode is turned off automatically if the user navigates away from the current playback position by manually turning a page.

A second application that makes use of the sheet music-audio synchronization is cross-domain navigation. Since the sheet music domain and the audio domain are linked, changing the playback position in one domain also affect the playback position in the other domain. A very useful application of this is to navigate to a particular position in the audio recording by selecting the corresponding bar in the sheet music. The compact visual representation of the music that is given by the sheet music makes finding a particular position of interest in a piece of music much easier and faster than searching for the corresponding time position in the audio recording. Using the score viewer, the user can navigate to any position in the piece of music by clicking the target bar with the mouse pointer. The audio viewer will, then, update the playback position in the audio recording accordingly. Note that a bar in the notation bar sequence of the sheet music can occur more than once in the performance. If
8.4. SWITCHING INTERPRETATIONS

In many cases, more than one interpretation or recording of a given piece of music is available. If all of these interpretations are synchronized to the same reference sheet music representation, the synchronizations can be used to enable a mechanism for convenient switching of interpretations during playback without changing the playback position. To this end, the score viewer offers access to an interpretation selector at the upper left corner, see Figure 8.8. The user can change the interpretation by selecting one of the available options from the list. The player, then, remembers the current bar and occurrence number, loads the target

Figure 8.7. The highlighting box of the bar under the mouse cursor is split into three horizontal slices indicating that this bar occurs three times in the performance. By clicking on the second slice from the top, the user directs the player to jump to the second occurrence of the bar in the performance.
CHAPTER 8. APPLICATIONS

Figure 8.8. If more than one audio interpretation of the current score track is available, the interpretation can be switched using a list that is opened by a click on the icon at the top left of the score viewer.

interpretation, and starts the playback at the time position that corresponds to the remembered bar and occurrence number in the new interpretation. Using this mechanism, the same sections of different interpretations can be compared conveniently without having to face the distraction of having to search for the right time position in the different recordings.

8.5 Cross-Domain Retrieval

Besides for presentation and navigation, the user interface presented in this chapter can also be used for content-based retrieval. To this end, the user can choose a section of the performance as a query by selecting a sequence of performance bars from the sheet music. Bars in the sheet music are selected in the same way as text in a standard text editor. The user clicks-and-holds on the first bar of the intended section and drags the pointer to the last bar of the intended selection. The selected bars are highlighted in translucent red color with the indices of the bars in the performance bar sequence being indicated at the center of each bar, see the left part of Figure 8.9. A right-click on the selected bars opens a context menu, from which the user selects the option “Query Selection using Audio Matching”. This triggers a content-based search using the audio data of the performance corresponding to the selected bars as a query. In the example shown in Figure 8.9, the query consists of the first theme of the Rondo of Beethoven’s Piano Sonata No. 8, Op. 13, “Pathetique”, which appears four times throughout the piece. The database that is used to search in is an index created from the audio data of all audio CD collections that are available through the system. For the content-based retrieval, a variant of a search technique called audio matching, which in its approach and output is somewhat similar to using subsequence dynamic time-warping as described in Section 3.5, is used. Details on audio matching techniques can be found in [75].
8.5. CROSS-DOMAIN RETRIEVAL

Figure 8.9. Left: A sequence of performance bars is selected in the sheet music book to formulate a content-based query. Right: The query results are displayed as regions highlighted in green color in the book and as bookmarks at the side of the book.

Figure 8.10. The query results of the content-based retrieval are displayed as a list of timelines with matches indicated as green boxes.

As a result, a set of matches is obtained, with each match specifying an audio track, a temporal region in that audio track, and a ranking value that expresses how similar the specified region is to the query. The matches that are found in the currently loaded audio CD collection are displayed as corresponding highlighted regions in the sheet music book, see the right side of Figure 8.9. Additionally, each match gets its own bookmark at the side of the book and is displayed as an entry in a result list on the left side of the score viewer. The complete set of matches, i.e., the matches that were found in any of the available audio CD collections, can be displayed in a separate window, which is shown in Figure 8.10. Several styles of visualizing the matches are possible. The figure shows the so-called “Timeline List View”. Here, each audio track that contains matches is displayed as an entry in a list. Each entry shows the title and meta information of the audio track and a set of green rectangular boxes on top of a blue bar indicating the position and duration of the matches on a timeline representing the audio track. The list is sorted by the ranking value of the top match contained in each entry. The ranking value is reflected in the transparency and color saturation of the rectangular boxes. Fully opaque and saturated green boxes, e.g., the first two boxes in the top row
of the illustration in Figure 8.10 indicate matches with very high similarity to the query. When hovering the mouse cursor over one of the matches, a pop-up frame displays additional information such as the temporal position of the region boundaries in the audio track and the ranking value. Clicking on a match opens the corresponding audio track in the audio viewer and starts playback from the starting position of the match. Furthermore, a corresponding sheet music book is displayed in the score viewer with the matches being highlighted as shown in the right part of Figure 8.9. In the top row of our example result list depicted in Figure 8.10 all four occurrences of the queried first theme of the Rondo of Beethoven’s Piano Sonata No. 8, Op. 13, “Pathétique” have been found. Since the audio matching technique used for the content-based retrieval is capable of finding similarities across different interpretations of the same piece of music, and the audio database used in the example contains more than one interpretation of this piece, the following lines reveal the same four matches in different interpretations of the same piece.
Chapter 9

Conclusions

As a result of the work conducted in this thesis, we have seen that automatic bar-wise synchronization of scanned sheet music and audio recordings of Western classical music can be achieved. Furthermore, we have presented appealing end-user applications such as assisted score reading, visual navigation in audio recordings through the sheet music, switching of interpretations, and content-based search of sections in the score. However, in the scenario of building up a large-scale digital music library that offers synchronization between sheet music and audio recordings, several practical issues arise. The three main issues are the quality of the OMR results, the segmentation and identification of individual movements or songs in the sheet music and audio documents, as well as differences in the global structure between pairs of representations of the same movement or song. When restricting to certain types of music and levels of print quality, the issues can be solved in a way that minimizes the required manual interaction for organizing the data. Nevertheless, some issues still remain. In particular, large variations in tempo throughout a piece of music, transposing instruments, and special cases such as cadenzas, optional sections, missing sections, or alternative sections (COMA) may cause problems that require manual editing. Even though the approaches developed and discussed in this thesis seem promising and experiments deliver encouraging results, the system has yet to be put into practice in the targeted real-world environment to prove its usefulness and sustainability – a challenge that is currently being worked on.

This work has laid the foundation for many interesting future directions and further research. An obvious approach to encounter the issues of missing tempo information and tempo changes would be to identify and interpret the tempo directives found in the sheet music scans. Since the OMR results of SharpEye 2.68 are not reliable enough, one either has to incorporate other OMR software, or develop methods specifically tailored to the task that directly work with the scanned material itself. Here, one could incorporate knowledge about the expected position of tempo directives relative to the (already known) grand staffs. A similar approach could be taken to encounter the issue of transposing instruments in orchestral scores. Here, what needs to be done basically is to determine which staffs are played by which instrument group. A mapping between staffs and instrument groups is usually indicated by text labels next to the first grand staff of the score of an orchestral piece, which could be identified and mapped to a known database of instrument group names and corresponding transpositions. However, the
CHAPTER 9. CONCLUSIONS

The task still includes many non-trivial problems to solve, such as individual instrument groups phasing out and back in the grand staffs off the orchestral score depending on whether or not they have notes to play during that part of the piece. Different conventions found in different scores on how to label the remaining instrument groups in such cases makes the task particularly hard.

To account for the missing information about important symbols and jump markers such as segnos and brackets for repeats with alternative endings, a straightforward approach would be to incorporate the results of a second OMR software that is capable of recognizing these symbols (e.g., SmartScore X). However, here one has to keep in mind that merging results from different OMR programs is far from trivial [71, 17]. Even though we were able to pragmatically fix some recognition errors of OMR using simple heuristics or sets of rules, a much better solution would be to incorporate more higher-level rules in the OMR software itself. Most OMR programs seem to make only very little use of the strong interdependencies that exist between the musical entities of sheet music. For example, the OMR engine used by SharpEye performs the recognition for each page individually. This means that any semantics that exceed the scope of a single page, e.g., key signatures and time signatures, can not be considered in the recognition and inference process of the immediately following pages. Most of the common OMR approaches follow a fixed processing pipeline to get from lower level entities to higher level entities. Therefore, high-level entities do not influence the decisions on the lower levels. To address this shortcoming, first mechanisms were introduced in [57] that allow higher-level steps to give feedback to lower-level steps. The decisions on how to group and construct the musical entities when reading a printed score are strongly interdependent. Using these interdependencies, a human reader is able to quickly recognize the musical entities and their semantic meaning. But when seen in an isolated fashion, as it is often done in optical music recognition systems, most of these decisions can only be made with high uncertainty.

A very interesting but also very challenging direction of future work would be to find methods for automatically detecting COMA cases such as cadenzas. For simpler cases such as locating the boundaries of a cadenza in an audio recording for which the position in the score is known, this seems quite feasible. More general and advanced cases such as handling cadenzas or missing sections for which there is no evidence in the score data, however, might first require improvements in other issues like OMR quality and tempo estimation before a reliable automatic detection becomes possible. To encounter issues with differences in tempo, an interesting strategy might be to extend the concept of mid-level features for local content comparison to incorporate independency of tempo (or at least robustness against temporal differences to a high degree). From symbolic note events such tempo-independent features could easily be derived by simply assigning one temporal frame to each time position where there is at least one note onset. However, deriving such features from audio recordings seems much harder. Possible approaches would be to incorporate transcription or tempo estimation/beat tracking techniques like [70, 83, 69] to eliminate differences and variations in tempo. Unfortunately, especially for classical music, the results that can be obtained with these techniques may not be reliable enough to yield a lot of improvement at the present time.

In a different direction of future work, one might work towards making the sheet music-audio synchronization finer and more robust. Using techniques for improving the temporal resolution of the synchronization such as [43], one can probably achieve a reasonable accuracy.
on a sub-bar level such as individual notes. The chroma features, cost/similarity measures and DTW settings used in the synchronization process include a variety of parameters that could be tuned and optimized to achieve a higher robustness. Here, it would be especially interesting to measure the effect of different settings and OMR errors on the overall synchronization results. One might also try to replace the DTW approach with an approach based on HMMs to allow finding optimal parameters through machine learning based on training data.

As already discussed in Chapter 6, important future work would be to implement the string-based comparison of titles and headings to assist in the task of track segmentation and identification. An approach to improving the partial synchronization of sheet music and audio discussed in Chapter 7 that might be worth pursuing is to incorporate automatic structure analysis of audio recordings as an additional source of information. Finally, we want to point out that a successful sheet music-audio synchronization paves the way for more applications than those presented in Chapter 8. Here, a very interesting idea is to use the symbolic data that is available for each staff individually to perform a time-dependent staff-based EQ-ing or filtering to acoustically highlight or suppress individual instrument groups in multi-instrument recordings, as has already been proposed in [107].
Appendix A

Calculations

In this Section, we give the details on the calculation of the computational complexity for the iterative approach for reducing inconsistency described in Section 5.3. Note that the calculation presented here is just a rough heuristic estimate rather than a detailed analysis. Since we expect the predominant factor in the computational cost of our algorithm to be checking the data for rule violations when evaluating the inconsistency measure, we want to estimate how many of these evaluations to expect depending on the length of the data and the number of rules included in the rule set. Let $B$ be the number of bars in the given OMR data and $R$ be the number of rules included in the rule set. We assume that the cost $t_{\text{rule}}$ for checking the data for violations w.r.t. a single rule scales linearly with the length of the data, i.e., $t_{\text{rule}} = c_1 \cdot B$ for some constant $c_1$. For a single evaluation of the inconsistency measure we need to check the data against a total of $R$ rules resulting in a computational cost

$$t_{\text{eval}} = R \cdot t_{\text{rule}} = c_1 \cdot B \cdot R.$$  (A.1)

Let $V(k) = |V^k|$ be the number of violations obtained at the $k$-th iteration. To simplify our calculation, we assume that we start with $V(0) = K$ violations at the 0-th iteration and eliminate one violation with each iteration until no more violation is left after a total of $K$ iterations

$$V(k) = K - k, \quad k = 0, \ldots, K.$$  (A.2)

Note that this assumption usually does not hold in practice, because a single modification to the data can solve many violations at once but can also cause several new violations. However, it does seem reasonable to assume that, on the average, $V(k)$ linearly decreases with $k$. We approximate the number of candidate modifications obtained at the $k$-th iteration $M(k) = |M^k|$ as being proportional $V(k)$ by setting

$$M(k) = c_2 \cdot V(k)$$  (A.3)

for some proportionality constant $c_2$. In each iteration, we perform one evaluation of the inconsistency measure on $\Omega^k$ in Steps 2 to 3 of the algorithm, and perform $M(k)$ evaluations on the modified data $\Omega^k_{m}$ in Steps 6 to 7. Therefore, for the $k$-th iteration, we perform
1 + M(k) evaluations of the inconsistency measure. The total computational cost for all $K$ iterations can then be estimated as

$$t_{\text{total}} = \sum_{k=0}^{K} (1 + M(k)) \cdot t_{\text{eval}}$$

$$= \left( (K + 1) + \sum_{k=0}^{K} M(k) \right) \cdot t_{\text{eval}}$$

$$= \left( 1 + K + \sum_{k=0}^{K} c_2 \cdot V(k) \right) \cdot t_{\text{eval}}$$

$$= \left( 1 + K + c_2 \sum_{k=0}^{K} (K - k) \right) \cdot t_{\text{eval}}.$$  \hspace{1cm} (A.4)

The sum in Equation (A.4) can be calculated as

$$\sum_{k=0}^{K} (K - k) = \sum_{k=0}^{K} k$$

$$= K \cdot (K + 1) / 2$$

$$= 1/2 (K^2 + K).$$

Replacing the sum in Equation (A.4) with this result, we get

$$t_{\text{total}} = \left( 1 + K + \frac{1}{2} c_2 (K^2 + K) \right) \cdot t_{\text{eval}}$$

$$= \left( 1 + (1 + \frac{1}{2} c_2)K + \frac{1}{2} c_2 K^2 \right) \cdot t_{\text{eval}}.$$  \hspace{1cm} (A.5)

It is reasonable to assume that the amount of violations $K = V(0)$ after 0 iterations scales linearly with both the number of bars $B$ and the number of rules $R$. Therefore, we set

$$K = c_3 \cdot B \cdot R.$$  \hspace{1cm} (A.6)

for some constant $c_3$. Using Equations (A.1) and (A.6) in Equation (A.5), we get

$$t_{\text{total}} = \left( 1 + (1 + \frac{1}{2} c_2)K + \frac{1}{2} c_2 K^2 \right) \cdot t_{\text{eval}}$$

$$= \left( 1 + (1 + \frac{1}{2} c_2) c_3 \cdot B \cdot R + \frac{1}{2} c_2 c_3^2 \cdot B^2 \cdot R^2 \right) \cdot c_1 \cdot B \cdot R$$

$$= c_1 B R + (1 + \frac{1}{2} c_2) c_1 c_3 \cdot B^2 \cdot R^2 + \frac{1}{2} c_1 c_2 c_3^2 \cdot B^3 \cdot R^3$$

$$= \mathcal{O}(B^3 \cdot R^3).$$
Bibliography


Index

accumulated cost matrix, 29
alignment, see synchronization
alignment path, 27
alignment optimum, 28
audio, 9
  matching, 150
  structure analysis, 13
  thumbnailing, 13
track, 50
viewer, 146
automatic computer accompaniment, 9
bar, 25
  source, 121
target, 121
bitmap data, 17
block, 121
  boundaries, 121
  boundary indicators, 122
  feature, 121
chord, 71
chroma, 39
  features, 39
COMA, 62
  classes, 62
  entries, 62
connections, 131
  jump, 132
  regular, 131
content-based retrieval, 12
cover song identification, 13
cross-domain, 148
da capo, 88
dal segno, 88
database, 40
default bar sequence, 25
degeneration, 126
  horizontal, 127
vertical, 127
dynamic time warping, 29
  jump, 130
engraving data, 17
execution, 18
expected performance block index sequence, 124
expression, 44
feature blocks, 124
fine, 88
frames, 27
geometric music reading, 18
item, 44
jump, 121
  backward, 121
  external, 121
  forward, 121
  internal, 121
JumpDTW, 130
leaf-level
  track, 50
  work, 44
local cost
  matrix, 27
  measure, 27
manifestation, 44
  reference, 59
match, 12
  audio, 150
  curve, 10
matrix
  accumulated cost, 29
  local cost, 27
optimum predecessor position, 29
mid-level representation, 38
MRO files, 19
music transcription, 13
notation bar sequence, 25
note events, 17
off-line, 9
on-line, 9
optical music recognition, 18
optimum predecessor position matrix, 29
parent-level
work, 44
partial synchronization, 119
performance bar sequence, 48
performance block index sequence, 122
query, 12, 40
raster group, 71
rasterization, 18
reference manifestation, 59
rendering, 18
rule set, 76
score
block sequence, 121
following, 9
track, 50, 104
viewer, 144
score-performance matching, 9
scoring, 18
semantic interpretation, 18
sequence
default bar, 25
expected performance block index, 124
notation bar, 25
performance bar, 48
performance block index, 122
score block, 121
source bar, 121
staff signature, 101
step
decision cost, 134
weights, 30
symbolic, 9
score data, 17
synchronization
audio-audio, 9
cross-domain, 9
partial, 12, 119
score-audio, 9
sheet music-audio, 24
symbolic-audio, 9
symbolic-symbolic, 9
target bar, 121
track
audio, 50
identification, 50, 103
leaf-level, 50
score, 50, 104
segmentation, 50, 103
start browser, 144
typsetting, 18
vector shapes, 17
violations, 76
Viterbi algorithm, 32
voice, 71
waveform data, 17
work, 44
leaf-level, 44, 103
parent-level, 44, 103