Masataka Goto and Roger B. Dannenberg

Music Interfaces Based on Automatic Music Signal Analysis

New ways to create and listen to music

©ISTOCKPHOTO.COM/TRAFFIC_ANALYZER

Music analysis based on signal processing offers new ways of creating and listening to music. This article focuses on applications and interfaces that are enabled by advances in automatic music analysis. By using signal processing, some of these applications provide nonexperts the chance to enjoy music in their daily lives, while other applications apply signal processing to enhance professional music production and create new opportunities for composers and performers. Described in this article are the history and state of the art of music interfaces as well as its future directions that emphasize interactive music applications based on automatic music signal analysis.

Applications of music signal processing

Music understanding and music analysis is part of the human experience; whether the listener is a nonmusician casually enjoying music and tapping along with the beat or a professional making a formal analysis or transcription. As with many other human-oriented tasks, engineers and scientists have been inspired to formalize and automate aspects of human music perception such as identifying tempo, chords, melody, and repetition. Automatic music analysis capabilities have inspired research into new interfaces that take advantage of these novel possibilities. At the same time, applications have inspired new developments in signal processing for music listening and understanding. We have seen an explosion of new and exciting applications and interfaces. In this article, we explore some of the recent and emerging possibilities for music signal processing in music software.

Music signal processing goes by many names. Among these are *machine listening* (which also includes nonmusic signals) and *music understanding* (which emphasizes deep musical abstractions, e.g., patterns and structures, in contrast to shallower features such as pitch, loudness, and note-onset times). *Music content analysis* emphasizes the processing of signals (content) as opposed to metadata (often machine-readable text, such as file names, tags, web pages, or catalog entries). In this article, *music analysis* (i.e., music signal analysis) is used to refer to virtually any type of automatic (computational) music

Digital Object Identifier 10.1109/MSP.2018.2874360
Date of publication: 24 December 2018

recognition, detection, decomposition, classification, or understanding. Music analysis can identify music structure (chorus, section, and repetition) [1], [2], melody lines, chords, beat structure (beat and bar), drums [3], and so on.

Music interfaces may focus on a single musical piece or on collections of music, such as playlists, personal libraries or online catalogs. By means of a number of representative examples, this article explains how automatic music analysis can augment music interfaces; however, it considers only interfaces that focus on a single musical piece. The following three sections present different types of music interfaces based on playback navigation, customization, and music production, respectively.

Music interfaces for content-aware playback navigation

The traditional way of listening to music is hearing the piece from beginning to end. In the past, before it became possible to record music audio, one could only hear music at a live performance. When recording music became a reality, one could play a specific musical passage on demand, although manually controlling phonograph and tape players was often time consuming and inconvenient. The listener's ability to change the playback position almost instantly, with just the push of a button, only began recently with digital music on compact discs and the personal computer.

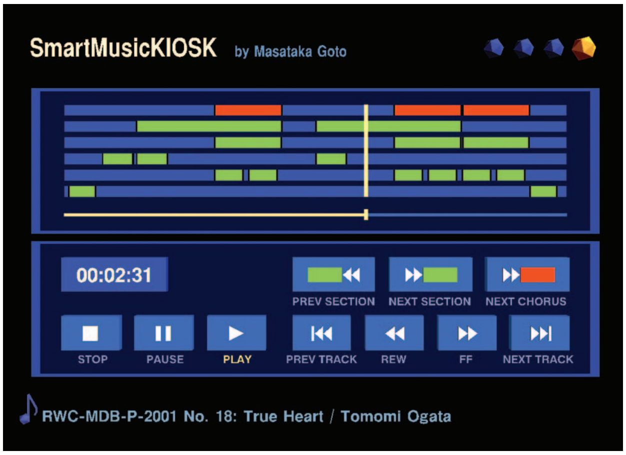
Interactive control of music playback is also a relatively recent development. Although digital music makes it easy for a listener to quickly jump from one song to another, only the fast-forward and rewind buttons can change the playback position within a musical piece. Even after media-player software on computers and portable digital audio players (e.g., MP3 players) appeared in the 1990s, music-listening interfaces remained unchanged except for a continuous playback slider. The total length of the slider corresponds to the length of a piece, and listeners can manipulate the slider to jump to any position in a song. However, listeners must use trial and error to search for a specific playback position.

Automatic music analysis based on signal processing addresses this problem by adding content-based navigation to conventional interfaces. Music interfaces that visualize music structure allow the listener to change the playback to logical positions. This approach is introduced in the "Music-Listening Interfaces Based on Automatic Music Structure Analysis" section of this article. Furthermore, when lyrics and music notation are aligned with audio signals, music interfaces display the lyrics or score in synchronization with the audio playback of a musical piece. As a result, new visual information about music content offers the listener a way to specify the playback position based on either lyrical or musical content. This approach is introduced in the "Music-Listening Interfaces Based on Automatic Music Synchronization" section.

Music-listening interfaces based on automatic music structure analysis

Automatic analysis of music structure improves conventional music-listening interfaces by using content-based playback navigation. The earliest of these works, introduced in 2003, is SmartMusicKIOSK [4], an intelligent music-listening station. In addition to the standard playback control buttons, SmartMusicKIOSK added a "jump to chorus" button and "jump to next/previous section" buttons, as shown in the lower window of Figure 1. SmartMusicKIOSK also extended the playback slider by visualizing the detected sections as the music structure. This visual representation, shown in the upper window of Figure 1, is called the *music map* and consists of chorus sections (the top orange row) and repeated sections (the five, lower green rows). In each row, colored sections indicate similar (repeated) sections. The music map helps a user decide where to jump to next, while each visualized section acts as a button to listen from the section's beginning.

The chorus and repeated sections are automatically determined by a signal processing method (RefraiD) [4] used for chorus-section detection, with a focus on popular music. First, a 12-dimensional feature vector, called a *chroma vector* [2], [4], is extracted from each frame of an input audio signal. Each element of the chroma vector corresponds to one of the 12 pitch classes (C, C#, D, D#, E, F, F#, G, G#, A, A#, and B) and the value of each element is the sum of magnitudes at frequencies of the pitch class over six octaves. In practice, this representation has been found to be robust with respect to changes in accompaniments, largely because its low dimensionality is enough to capture aspects of harmony and melody but not spectral details. The whole song is thus represented as a sequence of chroma vectors, i.e., a chromagram, and a pair of repeated sections is expected to have similar sequences of chroma vectors. RefraiD then calculates the similarity between all of the chroma vectors within the song and finds pairs of repeated sections whose similarity is higher than a certain threshold. This threshold is determined by an automatic threshold-selection method based on a discriminant criterion since the appropriate threshold varies for each song. To organize commonly repeated sections into groups and to identify both ends of each section, the pairs of repeated sections are integrated (grouped) by analyzing their relationships throughout the entire song. For example, three pairs of repeated sections, A and A', A' and A'', and A and A'', can be grouped on the basis of their relationships, even if one of the pairs is missing. Accordingly,



00:02:31

RWC-MDB-P-2001 No. 18: True Heart / Tomomi Ogata

FIGURE 1. The SmartMusicKIOSK interface [4]. A user can actively listen to various parts of a song, guided by the visualized music structure ("music map") in the upper window.

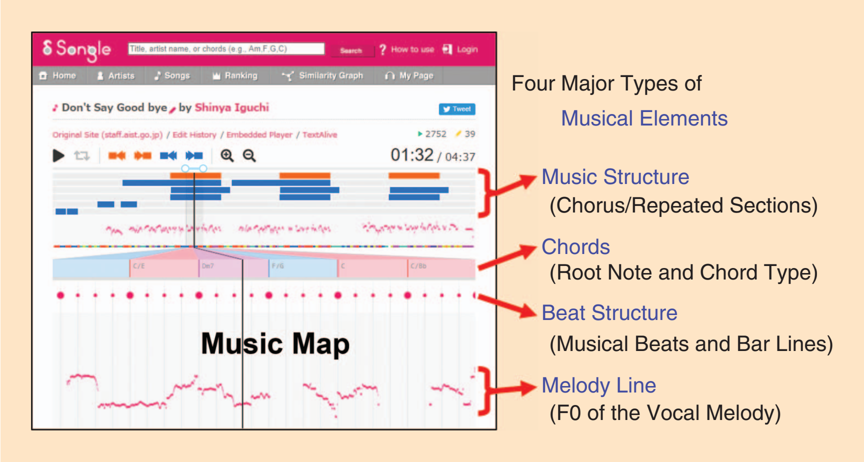

Songle
Title, artist name, or chords (e.g., Am,F,G,C)
Search
How to use
Login
Home
Artists
Songs
Ranking
Similarity Graph
My Page
Don't Say Good bye by Shinya Iguchi
Tweet
Original Site (staff.aist.go.jp) / Edit History / Embedded Player / TextAlive
2752
39
01:32 / 04:37
C/E
Dm7
F/G
C
C/Bb
Music Map
Four Major Types of
Musical Elements
Music Structure
(Chorus/Repeated Sections)
Chords
(Root Note and Chord Type)
Beat Structure
(Musical Beats and Bar Lines)
Melody Line
(F0 of the Vocal Melody)

FIGURE 2. An interface of the Songle web service [7]. Songle automatically analyzes songs publicly available on the web and visualizes them with an informative "music map," including four types of musical elements. It is also equipped with the SmartMusicKIOSK interface.

RefraiD obtains several groups of repeated sections as intermediate results (the five lower green rows in Figure 1). Finally, it selects the chorus sections from them by evaluating the possibility of being chorus sections for each group. This possibility is greater when its sections are repeated more frequently with higher similarity, are longer and more appropriately positioned.

The RefraiD method was sufficiently useful for detecting and playing back the chorus sections (the final output) during trial listening without any interface or visualization, but it was even more useful for visualizing the chorus sections along the playback slider, as shown in the top orange row on the music map of the SmartMusicKIOSK in Figure 1. We also found this method to be informative when visualizing the repeated sections in the five lower green rows in Figure 1, even when those sections were not final but intermediate results.

SmartMusicKIOSK thus augments within-song browsing and trial listening. A user can skip sections of a song that are of no interest by interactively changing the playback position while viewing the music map. This is an example of active-music-listening interfaces [5], which allow a user to enjoy music in more active ways than conventional passive music playback. Moreover, this interface can draw attention to music structures that are unknown to users. By enabling the user to listen to the chorus sections of a song in succession, the user can more accurately understand how lyrics and the arrangement change for each repetition of the chorus (as a reflection of the musicians' intent). SmartMusicKIOSK is not only an active interface, but also it is considered an example of augmented music-understanding interfaces [6] that facilitate a deeper understanding of music.

The interface concept of SmartMusicKIOSK is universal and can be used with other methods for music-structure analysis [1], [2]. Its interface is also versatile enough to be used with music structures annotated by humans even though the manual-annotation process is not scalable to a large music collection.

In fact, the SmartMusicKIOSK interface has already been implemented and made available for more than 1,200,000 songs on Songle [7], a public web service that was launched in 2011 and available at http://songle.jp free of charge. Songle enriches the music experience by providing an active, augmented music-listening interface. Through signal processing, Songle estimates not only the music structure but also the beat structure (beat and downbeat), melody line, and chords of songs available on the web and visualizes all of them (Figure 2). Given the wide variety of music available, one drawback of automatic analysis is that errors are inevitable. To overcome this, Songle provides a crowdsourcing interface that encourages users to correct errors in the estimated results by selecting from a list of alternatives or by providing an alternative annotation. The supplied corrections are then shared and used to immediately improve the user experience. Since Songle also provides an application programming interface (commonly known as API), the results of music analysis and human annotation can be used to develop music-driven applications such as robot dancing, stage lighting, and computer animation [8]. Songle therefore serves as an open showcase that demonstrates how people can benefit from signal processing-based music analysis and how interfaces can contribute to better music-listening experiences.

As previously mentioned in this section, a visual representation of music analysis results is key to changing traditional music interfaces into advanced interfaces with content-aware playback navigation. Another example is Dunya, a web-based application [9] that visualizes the pitch [fundamental frequency (F0)] contour of the melody line and its histogram as well as the waveform and spectrogram, with a focus on Carnatic music. By showing related recordings based on culturally specific similarity, it also allows a user to discover musically relevant relationships between different pieces.

Music-listening interfaces based on automatic music synchronization

Automatic synchronization of different representations of music, such as audio signals, lyrics, Musical Instrument Digital Interface (MIDI), and music scores, may also improve conventional interfaces with multifaceted, content-based music navigation and browsing. SyncPlayer [10], which is based on semiautomatic music synchronization procedures, is an early example of a music interface that provides users the opportunity to discover and explore multimodal representations of music. SyncPlayer's alignments between various music representations are computed in a preprocessing step and stored using suitable data structures. During the playback of a musical piece, it synchronously displays lyrics and a MIDI-based piano-roll view along with audio waveform and spectrogram. Time-aligned lyrics are shown in a karaoke-like display as the phrase currently being sung is highlighted. SyncPlayer has a lyrics search function that enables a user to submit a text-based query for lyrics that finds the corresponding audio. Time-aligned MIDI is generated by an automatic score-to-audio synchronization (alignment) method [11] and visualized in a piano-roll display. SyncPlayer first detects note onsets in the audio signal of a musical piece to obtain a score-like representation. This representation is then aligned with musical notes of a MIDI file by using a dynamic time-warping (DTW) algorithm.

This concept is further extended to the Score Viewer and Interpretation Switcher interfaces [11], [12], as shown in Figure 3. Score Viewer displays a time-aligned music score (scanned sheet music) that highlights the current bar. With a focus on Western classical music, spatial regions of the scanned sheet music are automatically synchronized with musically corresponding temporal sections within the audio recording. Score Viewer first extracts chromagrams (temporal sequences of the chroma vectors, as described in the previous section) from the results of optical music recognition of the sheet music. It then uses DTW to align those representations with chromagrams of the audio recordings. In classical music, recordings of different performers playing the same piece are often available. Interpretation Switcher automatically synchronizes those recordings and allows a user to seamlessly switch from one performance to another while continuing playback.

Score Viewer does not synchronize real-time audio input with the sheet music. To enrich the audience's experience of classical music concerts, however, real-time input is necessary. Another project, EU FP7 PHENICX (http://phenicx.upf.edu), developed and used an automatic, real-time audio-to-sheet-music synchronization method to track a live public performance of the Royal Concertgebouw Orchestra. During the performance, time-aligned sheet music was displayed for an audience in a concert hall in Amsterdam [13].

While Interpretation Switcher synchronizes different performances and allows comparisons by ear, a web-based interface [14] facilitates a more objective comparison of features of loudness (using dynagrams) and tempo (using tempograms) in two performances. Music performances can also be shown as two-dimensional (2-D) tempo-loudness trajectories called *performance worms*. The alignment between the waveform displays of two performances is visualized as line patterns connecting the corresponding bar lines. This visualization also includes an interactive musical-score display based on automatic alignment.

LyricSynchronizer [15], another interface that synchronizes symbolic text displays with music playback, is lyrics oriented and displays scrolling time-aligned lyrics by using an automatic lyrics-to-audio synchronization method. Because lyrics are automatically highlighted, a user can easily follow the current playback position. Additionally, the user can click on a word that interests them and listen to a song from that word forward.

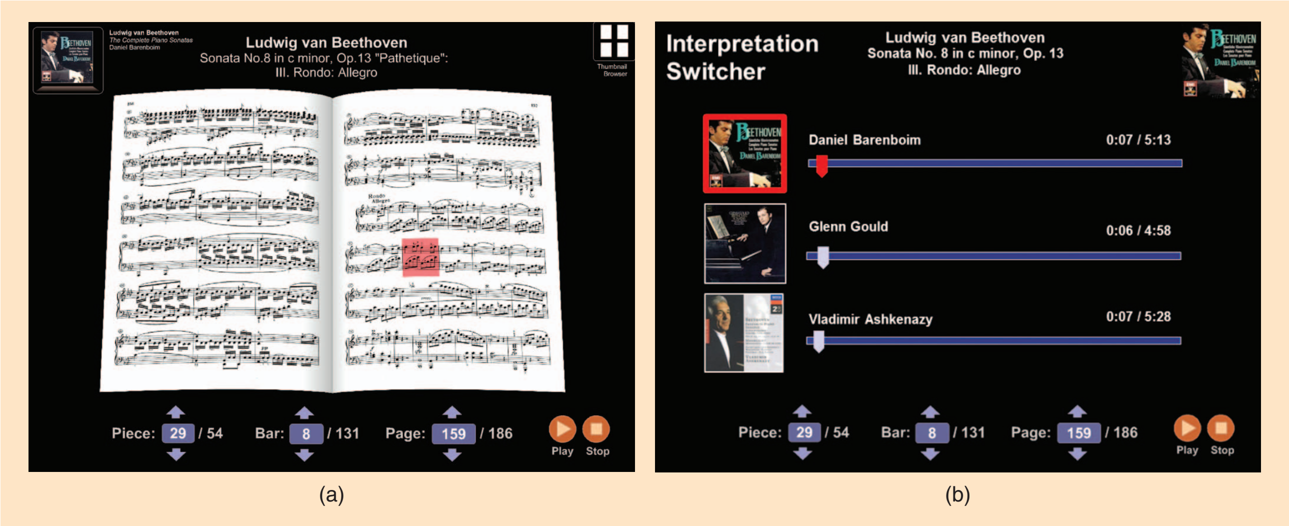
Music interfaces for customization and personalization

Traditional music players often include graphic equalizers or tone controls for bass and treble. Listeners can therefore customize/personalize music playback in a simple way by adjusting the overall frequency response. However, listeners cannot change the volume or timbre of each individual instrument in existing recordings unless individual tracks, called *stems* (separate recordings before mixing, corresponding to different instruments), are provided.

Sound source separation of musical audio signals can overcome this limitation and enable new types of music interfaces that allow a listener to customize music by changing the volume or timbre of instrument sounds in existing music recordings or by altering notes and styles. These kinds of creative customization represent music personalization for a user.

Music-customization interfaces based on sound source separation

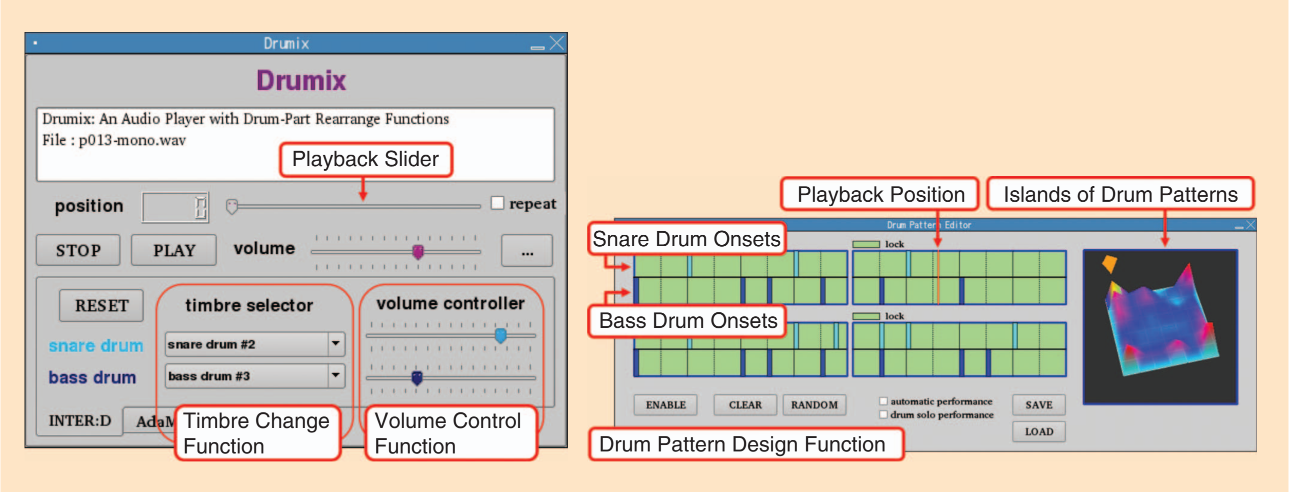
Drumix [16] is an early example of a music-customization interface that allows a user to edit the drum part of an existing recording during music playback the same as if another drummer was performing different drum patterns. With this interface, a user can change the volume or timbre of the sounds of



(a)

(b)

FIGURE 3. (a) The Score Viewer and (b) Interpretation Switcher interfaces. The Score Viewer displays interactive scanned sheet music synchronized with music playback. The Interpretation Switcher enables a user to seamlessly switch to different recordings of the same piece of music [11].



Drumix: An Audio Player with Drum-Part Rearrange Functions
File : p013-mono.wav

FIGURE 4. The Drumix interface. A user can actively change the volume or timbre of drum sounds and rearrange drum patterns during music playback [16].

bass and snare drums, as shown in the left window of Figure 4. In the right window, the user can also rearrange drum patterns of bass and snare drums by dragging a pattern from "islands of drum patterns" in which each island represents a different cluster of similar drum patterns. The larger an island is, the more popular its drum patterns are in a corpus of drum patterns. A user who is usually unaware of the drum pattern or timbre of drum sounds can use Drumix to edit them, which helps a user develop an appreciation of the musical choices of performers and producers. Drumix thus enhances the ability of the user to understand the role of drums in songs.

The onset times and spectrograms of drum sounds are automatically estimated by AdaMast, a drum-sound-recognition method [16]. It first prepares a seed template that is the spectrogram of a typical bass or snare drum sound and then detects onset times of drum-sound candidates in an existing music recording (polyphonic sound mixture) by using a template-matching technique. Since the seed template is different from the actual drum sound in the recording, AdaMast uses the median of the detected spectrograms to update its template. It then uses the updated template to repeat this iterative-matching and adaptation process. After several iterations, AdaMast obtains the template (i.e., spectrogram) of the actual drum sound, which can then be used to separate, change, remove, and add drum sounds. To deal with drum patterns in units of bar (measure), Drumix also uses a beat-tracking method.

The concept of Drumix can be used not only with other drum-sound recognition methods [3] but also with any instrument or voice if sound source separation for them can be achieved. Given polyphonic sound mixtures of popular music, however, it is well known to be extremely difficult to decompose them into all of the original stems because musical audio signals often combine more than ten simultaneous sounds with overlapping frequency, content, and reverberation. An ongoing, unsolved challenge for signal processing researchers is to achieve better source separation [17] and enable higher-quality audio manipulation of arbitrary music mixtures [11], [16].

Despite these challenges, this concept has been further investigated by different research groups. For example, a music-manipulation method [18] can change the timbre and phrases of a pitched instrument part. Because it is difficult to segregate an arbitrary instrument part from polyphonic sound mixtures, this method is based on score-informed source separation [19] that leverages a musical score of the target part to help isolate its sound and change its timbre. This method also changes the original phrase into a phrase specified by another score provided by the user.

By decomposing an existing recording of the input song into the vocal track (singing voice) and the karaoke track (the rest of the input sound mixture), a vocal-editing interface was proposed in [20]. This interface allows a user to manipulate vocal F0 by adding vocal expressions (e.g., vibrato and glissando) and changing the melodic contour (i.e., the pitches of musical notes).

Even if users are not musicians, music signal processing enables easy-to-use customization of existing music that allows for enjoying music in active ways and facilitates a deeper understanding of music. The interfaces discussed in this section are considered good examples of active music-listening interfaces [5] and augmented music-understanding interfaces [6].

Music interfaces for production and performance

Music analysis presents new capabilities for computer-assisted music creation and performance. In the "Music-Production Interfaces Based on Score-to-Audio Alignment" section, we examine how music analysis enables audio-editing software to "adjust" music recordings automatically based on models of pitch and rhythm, perhaps with guidance from machine-readable music notation, envisioned as early as 1982 [21]. In the "Real-Time Signal Analysis in Interactive Music Performance" section, we see examples of how new modes of music performance are enabled by real-time machine listening.

Music-production interfaces based on score-to-audio alignment

Audio editors typically use visual representations of waveforms and spectrograms, but these are difficult to comprehend and navigate. As an alternative, music-editing software can display symbolic representations of music alongside waveforms, as shown in the Figure 5 mockup. With this style of interface, the music audio is actively labeled with a human-readable notation, which facilitates search and navigation. Constructing such an interface requires some form of symbolic notation and a method to align audio to it.

Obtaining symbolic notation directly from audio requires automatic music transcription [17]. Since this is extremely difficult to achieve (especially for recordings of multiple instruments), full, automatic transcription from arbitrary music signals is not likely to be practical for building notation-based interfaces in the near future. On the other hand, since many composers already use music-notation software, their music exists in a machine-readable form. Rather than transcribing audio, interfaces can align existing notation to music audio. For example, an experimental version of the Audacity (https://www.audacityteam.org/) audio editor can display MIDI files in a piano-roll view that is automatically aligned to a corresponding audio track.

Score-to-audio alignment (synchronization) generally works by converting music to feature sequences, such as the chromagram described in the "Music-Listening Interfaces Based on Automatic Music Structure Analysis" section, and using DTW or hidden Markov models to align them [22]. In this audio-editing system, DTW was used, as in SyncPlayer, described in the "Music-Listening Interfaces Based on Automatic Music Synchronization" section.

Navigating a digital audio track is a common facet of audio editing. Digital-editing software allows recording engineers and music producers to apply advanced signal processing techniques to make timing, pitch, and loudness adjustments on a note-by-note basis. Sophisticated interfaces have evolved to support this work, but actual edits are nearly always specified manually. One exception is the Antares Audio Technologies Auto-Tune product (https://www.antarestech.com/), which has become a standard tool in music production for correcting off-key pitch. Auto-Tune works mainly by shifting pitch to the nearest musical scale degree as specified by the user, so it can automatically calculate a target pitch and apply pitch corrections. Apple's Flex Time (https://support.apple.com/kb/PH13083) processing interface enables automatic timing adjustments in one track to be guided by audio transients in another track, which is much easier than manually performing "microsurgery" to achieve the same result.

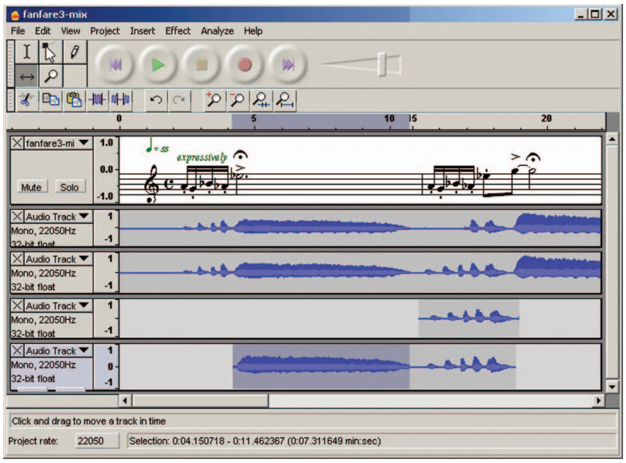
Rather than simply quantizing to pitches or beats, score-to-audio alignment provides an audio editor the ability to automatically determine the intended timing, pitch, and loudness of every note by reading the score, compare that to every performed note based on the score-to-audio alignment, and then use signal processing techniques to adjust audio recordings [23]. In this article, multitrack audio is assumed, and each instrument is recorded on a separate track. Each track is then aligned separately with music notation for a specific instrument. Since monophonic instruments are assumed, alignment is based on DTW to match pitch sequences obtained from onset-detection-based note segmentation and F0 estimation. Next, the interface produces a list of edits, applying small pitch adjustments (through resampling) and timing adjustments (by cutting, splicing and cross-fading) on a note-by-note basis. Finally, tracks are mixed to balance the average root mean square. This article shows that "intelligent" editors can automate and simplify many routine edits made in music production.

Of course, forcing audio to meet precise specifications can remove important musical nuance. Rather than "fixing" everything, an audio editor might present an interface to act as a spell checker, in which the human engineer decides to accept or reject each of the computer's suggested changes. We see this as a promising direction for future audio-editing interfaces and a logical extension of some of the automated tools and interfaces that exist for commercial editors today.

Beyond editing to correct mistakes or polish recorded performances, music producers use equalization, gain control, reverberation, stereo placement, and many other techniques to creatively enhance their work. There is growing interest in computational music production, and there are many automated mastering services online, already claiming millions of mastering sessions in total. As signal processing becomes more complex [24], interfaces are needed to operate at higher levels of abstraction. A machine-learning approach [25], for instance, was proposed to describe filter-transfer functions with user-oriented terms such as *warm* or *bright*.

Real-time signal analysis in interactive music performance

Some of the earliest work in music audio analysis was motivated by composers and performers exploring real-time sensing and computation to create interactive musical works. These works often used F0 estimation to obtain pitches from monophonic instruments because the simple hardware needed for this purpose was readily available. For example, Voyager [26] is a pioneering interactive system that uses note-level analysis. Monophonic audio input is analyzed for pitch (F0) and dynamic (signal-amplitude) information, which is processed to form pitch histograms, note density, and other features. These, in turn, influence music-generation algorithms that control a music synthesizer, thereby producing something akin to a collectively improvising ensemble. In [26], Lewis describes his system in terms of improvisation: "Improvisation must be



Click and drag to move a track in time

Project rate: 22050 Selection: 0:04.150718 - 0:11.462367 (0:07.311649 min:sec)

FIGURE 5. The concept design for an intelligent editor that stretches and aligns music notations and audio, which enables users to quickly navigate to, select, and splice together the best "takes" from a recording session [23].

open—i.e., open to input, open to contingency—a real-time and (often enough) a real-world mode of production." In this sense, music analysis is critical to his work.

Live electronic compositions are not as well known as commercial popular music and Western classical music, but there are many international festivals that feature interactive computer music, and music analysis is playing an increasingly important role by enabling composers to incorporate more sophisticated "listening" into their works. These works illustrate and explore the possibilities of nongraphical user interfaces. One example is CataRT [27], which slices input sound into short grains and organizes those grains according to composer-/performer-selected features. These features can have high dimension, drawing from spectral, perceptual, and harmonic descriptors; or, when applied another way, grains are projected onto a 2-D display that can be navigated with mouse or trackpad input. The performer then uses this control space to create sonic textures by summing from a few to thousands of grains per second. The navigation can also be controlled by features obtained from a musician's audio in real time, offering a sort of analysis/resynthesis system in which the representation is a highly abstract feature space.

Another interesting development is that of Wekinator [28] (http://www.wekinator.org/), a machine-learning software package developed especially for musicians and interactive music performances. Wekinator uses a visual interface to simplify the capture of input-to-output examples that are used to train the response of interactive systems. A typical application is mapping audio features or even a simple fast Fourier transform to the multidimensional control space of a music synthesizer or music-composition algorithm. A variety of classifiers are then used for supervised training of the input-to-output mapping.

Yet another class of interactive music performance systems is occasioned by computer accompaniment, which models the familiar scenario of a soloist and accompanist, such as a flute accompanied by piano, except that the accompanist's part is played by a computer system that "listens" to the soloist, follows along in a machine-readable score, and synchronizes the accompaniment part to the live soloist [22], [29]. In this model, both the solo and accompaniment parts are composed and played note for note; therefore, the task performed by the computer is primarily that of synchronization. Computer accompaniment systems use various algorithms for "score following," including DTW and hidden/semihidden Markov models. The signal processing challenges associated with these systems include dealing with the presence of accompaniment audio in a live performance (even with a microphone in close to the soloist). These systems also implement various strategies for musically adjusting tempo to maintain synchronization. In addition to performance, score-following technology allows for rich-performance interfaces that feature automatic music page turning and automatically generated feedback to student musicians.

Discussion and future directions

Music signal processing continues to encourage exciting new ways of working with music. From the listener's perspective, we have seen how music interfaces can help to visualize music information and assist users in music playback, navigation, and multifaceted browsing. For creative amateurs, interfaces can harness sophisticated signal processing for customization or personalization of music, while for professionals, applications automate high-level editing and production tasks, allowing composers and performers to use music analysis to creatively control music generation and create new "instruments."

In the future, advances in automatic music analysis will inspire and provide more advanced music interfaces. Conversely, the invention of novel music interfaces will require more advanced signal processing for music analysis. This interdependency drives the development and improvement of both novel music interfaces and state-of-the-art signal processing methods. Although significant research progress has been made in the past 30 years, music-analysis technologies have not yet matured and remain far from human levels of music understanding and analysis. Further progress is important for advanced music interfaces. For example, active music-listening and augmented music-understanding interfaces may benefit from advances in automatic music analysis. As discussed in the "Music Interfaces for Customization and Personalization" section, music-customization interfaces would benefit from better source separation and audio-manipulation techniques; and music-production interfaces would benefit from automatic music transcription of arbitrary polyphonic sound mixtures.

In addition to the efforts of advancing music signal processing, another important future direction is to research and develop a variety of music interfaces that involve human intelligence (i.e., human in the loop). For example, the Songle service discussed in the "Music-Listening Interfaces Based on Automatic Music Structure Analysis" section features an error-correction interface. Tarsos [30], a system used for pitch analysis in Western and non-Western music, employs F0-estimation algorithms such as the standard YIN and McLeod Pitch Method (https://github.com/JorenSix/TarsosDSP) but offers a graphical interface to guide the analysis. Similarly, the Interactive Source Separation Editor (ISSE) [31] (http://isse.sourceforge.net/) uses a sophisticated interface for source separation based on probabilistic latent-component analysis, including machine learning from manual corrections. Interfaces that integrate human control and knowledge with automatic music analysis are advancing rapidly, and we expect to see increasingly sophisticated interaction in future intelligent systems used for music editing and production.

Another theme in emerging research is a consistent drive toward more active listening. If computers bring interactive and "smart" capabilities, and if music is now mediated by computers, it seems only natural to pursue greater interactivity and intelligence in interfaces for music. We see this trend in many experimental interfaces for music listening, and there are hundreds of interactive music games, tablet-based electronic instruments, and composing programs. More active music-listening interfaces such as SmartMusicKIOSK, Songle, and Drumix have the potential to blur the boundaries between music listening, music creation, games, and entertainment. Perhaps the extreme form of active listening is music performance, where interactive software such as SmartMusic (https://www.smartmusic.com/) provides always-available music instruction and accompaniment.

We have seen a revolution in music storage, processing, and distribution brought about by digital signal processing. The digitization of music has progressed from an initial, quantitative phase in which costs came down and the number of recordings in music collections went up. Today, we are in a second, qualitative phase that is changing the nature of musical experiences. We believe this phase will reveal the true value of digitization. The key to change is automatic music analysis, which enables music interfaces to move from just storing music to offering high-level musical interactions. Music interfaces based on music analysis produce qualitative changes in music experiences for professional and casual listeners alike.

Authors

Masataka Goto (m.goto@aist.go.jp) received his B.E. and Ph.D. degrees in engineering from Waseda University, Tokyo, Japan, in 1993 and 1998, respectively. He is currently a prime senior researcher at the National Institute of Advanced Industrial Science and Technology (AIST). In 1992, he worked on automatic music understanding based on signal processing and has since contributed to the research and development of music technologies and music interfaces based on those technologies. He has published more than 300 papers in refereed journals and international conferences and has received 47 awards, including several best paper awards, best presentation awards, the Tenth Japan Academy Medal, and the Tenth Japan Society for the Promotion of Science Prize.

Roger B. Dannenberg (rbd@cs.cmu.edu) received his B.S. degree in electrical engineering from Rice University in 1977 and his Ph.D. degree in computer science in 1982 from Carnegie Mellon University, Pittsburgh, Pennsylvania, where he is currently a professor of computer science, art, and music. His pioneering work in computer accompaniment led to the awarding of three patents and the advent of the SmartMusic system now used by tens of thousands of music students. He is the chief science officer for Music Prodigy and a cocreator of Audacity, an open-source audio editor that has been downloaded more than 300 million times. As a trumpet player, he is active in performing jazz, classical music, and new works. His compositions include many interactive computer works, and, in 2017, he premiered La Mare dels Peixos, an opera cocomposed with Jorge Sastre.

References

[1] R. Dannenberg and M. Goto, “Music structure analysis from acoustic signals,” in *Handbook of Signal Processing in Acoustics*, D. Havelock, S. Kuwano, and M. Vorländer, Eds. Berlin: Springer-Verlag, 2009, pp. 305–331.

[2] J. Paulus, M. Müller, and A. Klapuri, “State of the art report: Audio-based music structure analysis,” in *Proc. 11th Int. Society Music Information Retrieval Conf. (ISMIR)*, 2010, pp. 625–636.

[3] C. W. Wu, C. Dittmar, C. Southall, R. Vogl, G. Widmer, J. Hockman, M. Müller, and A. Lerch, “A review of automatic drum transcription,” *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 26, no. 9, pp. 1457–1483, 2018.

[4] M. Goto, “A chorus-section detection method for musical audio signals and its application to a music listening station,” *IEEE Trans. Audio, Speech, Language Process.*, vol. 14, no. 5, pp. 1783–1794, 2006.

[5] M. Goto, “Active music listening interfaces based on signal processing,” in *Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2007, pp. 1441–1444.

[6] M. Goto, “Frontiers of music information research based on signal processing,” in *Proc. 12th IEEE Int. Conf. Signal Processing*, 2014, pp. 7–14.

[7] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano, “Songle: A web service for active music listening improved by user contributions,” in *Proc. 12th Int. Society Music Information Retrieval Conf. (ISMIR)*, 2011, pp. 311–316.

[8] J. Kato, M. Ogata, T. Inoue, and M. Goto, “Songle Sync: A large-scale web-based platform for controlling various devices in synchronization with music,” in *Proc. ACM Multimedia*, 2018, pp.1697–1705.

[9] A. Porter, M. Sordo, and X. Serra, “Dunya: A system to browse audio music collections exploiting cultural context,” in *Proc. 14th Int. Society Music Information Retrieval Conf. (ISMIR)*, 2013, pp. 101–106.

[10] F. Kurth, M. Müller, D. Damm, C. Fremerey, A. Ribbrock, and M. Clausen, “SyncPlayer—an advanced system for multimodal music access,” in *Proc. 6th Int. Society Music Information Retrieval Conf. (ISMIR)*, 2005, pp. 381–388.

[11] M. Müller, “*Fundamentals of Music Processing—Audio, Analysis, Algorithms, Applications*,” Berlin: Springer-Verlag, 2015.

[12] D. Damm, C. Fremerey, V. Thomas, M. Clausen, F. Kurth, and M. Müller, “A digital library framework for heterogeneous music collections: from document acquisition to cross-modal interaction,” *Int. J. Digit. Libraries*, vol. 12, no. 12, pp. 1726–1737, 2014.

[13] A. Arzt, H. Frostel, T. Gadermaier, M. Gasser, M. Grachten, and G. Widmer, “Artificial intelligence in the concertgebouw,” in *Proc. 24th Int. Joint Conf. Artificial Intelligence (IJCAI)*, 2015, pp. 2424–2430.

[14] M. Gasser, A. Arzt, T. Gadermaier, M. Grachten, and G. Widmer, “Classical music on the web—user interfaces and data representations,” in *Proc. 16th Int. Society Music Information Retrieval Conf. (ISMIR)*, 2015, pp. 571–577.

[15] H. Fujihara, M. Goto, J. Ogata, and H. G. Okuno, “LyricSynchronizer: Automatic synchronization system between musical audio signals and lyrics,” *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 6, pp. 1252–1261, 2011.

[16] K. Yoshii, M. Goto, K. Komatani, T. Ogata, and H. Okuno, “Drumix: An audio player with real-time drum-part rearrangement functions for active music listening,” *Info. Proc. Soc. Japan (IPSJ) Journal*, vol. 48, no. 3, pp. 1229–1239, 2007.

[17] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri, “Automatic music transcription: Challenges and future directions,” *J. Intell. Inform. Syst.*, vol. 41, no. 3, pp. 407–434, 2013.

[18] N. Yasuraoka, T. Abe, K. Itoyama, T. Takahashi, T. Ogata, and H. G. Okuno, “Changing timbre and phrase in existing musical performances as you like: Manipulations of single part using harmonic and inharmonic models,” in *Proc. ACM Multimedia*, 2009, pp. 203–212.

[19] S. Ewert, B. Pardo, M. Müller, and M. D. Plumbley, “Score-informed source separation for musical audio recordings: An overview,” *IEEE Signal Process. Mag.*, vol. 31, no. 3, pp. 116–124, 2014.

[20] Y. Ikemiya, K. Yoshii, and K. Itoyama, “Singing voice analysis and editing based on mutually dependent F0 estimation and source separation,” in *Proc. IEEE Int. Conf. Acoustics Speech and Signal Processing (ICASSP)*, 2015, pp. 574–578.

[21] C. Chafe, B. Mont-Reynaud, and L. Rush, “Toward an intelligent editor of digital audio: Recognition of musical constructs,” *Comput. Music J.*, vol. 6, no. 1, pp. 30–41, 1982.

[22] R. Dannenberg and C. Raphael, “Music score alignment and computer accompaniment,” *Commun. ACM*, vol. 49, no. 8, pp. 38–43, 2006.

[23] R. Dannenberg and N. Hu. “Polyphonic audio matching for score following and intelligent audio editors,” in *Proc. Int. Computer Music Conf. (ICMC)*, 2003, pp. 27–34.

[24] E. P. Gonzalez and J. D. Reiss, “Automatic equalization of multi-channel audio using cross-adaptive methods,” in *Proc. 127th Audio Engineering Society Conv. (AES)*, 2009.

[25] B. Pardo, D. Little, and D. Gergle, “Building a personalized audio equalizer interface with transfer learning and active learning,” in *Proc. 2nd Int. ACM Workshop Music Information Retrieval with User-Centered and Multimodal Strategies (MIRUM)*, 2012, pp. 13–18.

[26] G. Lewis, “Too many notes: Computers, complexity and culture in Voyager,” *Leonardo Music J.*, vol. 10, pp. 33–39, Dec. 2000.

[27] D. Schwarz, “Corpus-based concatenative synthesis: Assembling sounds by content-based selection of units from large sound databases,” *IEEE Signal Process. Mag.*, vol. 24, no. 2, pp. 92–104, 2007.

[28] R. Fiebrink, “Real-time human interaction with supervised learning algorithms for music composition and performance,” Ph.D. dissertation, Comput. Sci. Dept. Princeton Univ., New Jersey, 2011.

[29] A. Maezawa and K. Yamamoto, “MuEns: A multimodal human-machine music ensemble for live concert performance,” in *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, 2017, pp. 4290–4301.

[30] J. Six, O. Cornelis, and M. Leman, “Tarsos, a modular platform for precise pitch analysis of western and non-western music,” *J. New Music Res.*, vol. 42, no. 2, pp. 113–129, 2013.

[31] N. Bryan, G. Mysore, and G. Wang, “ISSE: An interactive source separation editor,” in *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, 2014, pp. 257–266.