Meinard Müller, Andreas Arzt, Stefan Balke,
Matthias Dorfer, and Gerhard Widmer

Cross-Modal Music Retrieval and Applications

An overview of key methodologies

©ISTOCKPHOTO.COM/TRAFFIC_ANALYZER

There has been a rapid growth of digitally available music data, including audio recordings, digitized images of sheet music, album covers and liner notes, and video clips. This huge amount of data calls for retrieval strategies that allow users to explore large music collections in a convenient way. More precisely, there is a need for cross-modal retrieval algorithms that, given a query in one modality (e.g., a short audio excerpt), find corresponding information and entities in other modalities (e.g., the name of the piece and the sheet music). This goes beyond exact audio identification and subsequent retrieval of metainformation as performed by commercial applications like Shazam [1].

In this article, we review several cross-modal retrieval scenarios, with a particular focus on sheet music (visual domain) and audio (acoustic domain). First, we discuss a traditional approach where the sheet music and audio representations are converted into common midlevel feature representations that capture musical properties related to pitches and harmony. The resulting feature sequences can then be compared using standard alignment algorithms [2], [3].

Second, we review an approach based on symbolic fingerprinting techniques. Originally, audio fingerprinting referred to a procedure that allows for a robust identification of exact replicas of audio recordings [4]. In our cross-modal scenario, however, we discuss tempo- and transposition-invariant symbolic fingerprinting methods based on note parameters extracted via audio transcription techniques [5], [6].

Third, employing deep-learning methods, we describe an end-to-end, cross-modal retrieval strategy that works without the need for manually crafted feature representations [7]. Given snippets of sheet music (in the form of pixel images) and corresponding audio excerpts (in spectrograms), a neural network learns a joint embedding space on which cross-modal retrieval can be performed using simple distance measures and nearest-neighbor search.

Using these three approaches as illustrative examples, the primary objective of this article is to discuss the principles and challenges encountered in general music processing, such as designing musically motivated features and similarity measures to cope with semantic data variability. Furthermore, to

Digital Object Identifier 10.1109/MSP.2018.2868887
Date of publication: 24 December 2018

 1053-5888/19©2019IEEE

illustrate the potential of cross-modal retrieval techniques, we describe some navigation and browsing applications, including a prototype system called the *Piano Music Companion*, while indicating future research directions.

Music representations

Before we delve into the various cross-modal retrieval approaches, we first introduce some basic notions, following [3, Ch. 1]. As indicated by Figure 1, music can be represented in many different ways and formats. For example, a composer may write a composition in the form of a musical score, where musical symbols are used to visually encode which notes are to be played and how. The printed form of a musical score is also referred to as *sheet music*. The original medium of this representation is paper, although it is now also accessible on computer screens in the form of digital images.

In electronic instruments and computers, music can be communicated by means of standard protocols—such as the widely used Musical Instrument Digital Interface protocol (https://www.midi.org/)—where event messages specify note pitches, note intensities (velocities), and other parameters to generate the intended sounds. Often, the term *symbolic* is used to refer to any data format that explicitly represents musical entities. The musical entities may range from timed note events, as is the case in MIDI files, to graphical shapes with attached musical meaning, as in music engraving systems. In contrast to such symbolic representations, the musical events are not given explicitly in audio representations, such as WAV or MP3 files. The latter formats encode acoustic waves that are generated when, e.g., playing an instrument and travel from the sound source to the human ear as air-pressure oscillations.

At this point, it is important to note that each of these representations reflects certain aspects of a musical entity but that no single representation encompasses all of the properties. For example, rather than giving strict specifications, a musical score serves only as a guide for performing a piece of music, leaving room for different interpretations. Reading the instructions in the score, a musician shapes the music by varying the tempo, dynamics, articulation, and other parameters, thus creating a personal interpretation of the piece. Furthermore, while sheet music visually encodes the musical notes, such information is hidden in an audio recording, which is basically a time series of samples. In summary, even if various formats refer to the same piece of music, there may be a significant gap—technically as well as semantically—between different representations, such as sheet music and audio.

The boundaries between the diverse music representations are not sharp. As illustrated by Figure 1, symbolic representations—depending on their specific format and intended application—may be closer to sheet music or audio representations. For example, symbolic representations such as MusicXML (https://www.musicxml.com/) are used for rendering sheet music, where the shape of the note objects and their arrangement on a page are determined. Optical music recognition (OMR) can be seen as the inverse process, with the objective to transform sheet music into a symbolic representation.

Furthermore, symbolic representations such as MIDI are used for synthesizing audio, where the note objects are transformed into musical tones and real sounds. The inverse process is known as *automatic music transcription* (*AMT*) and aims at extracting note events, key signature, time signature, instrumentation, and other score parameters from a given music recording [3]. Both transformations, OMR and AMT, are far from straightforward. For example, correctly recognizing and interpreting the meaning of all of the musical symbols in complex sheet music is easy for a trained human but hard for a computer. Even though current OMR software is reported to yield highly accurate results, manual postprocessing is necessary to obtain a high-quality symbolic representation [8]. Similarly, converting a music recording into a note-based representation is a largely unsolved problem—in particular, for multivoiced music involving different instruments [9].

For relating different types of data (e.g., sheet music and audio data) to each other, traditional methods are often based

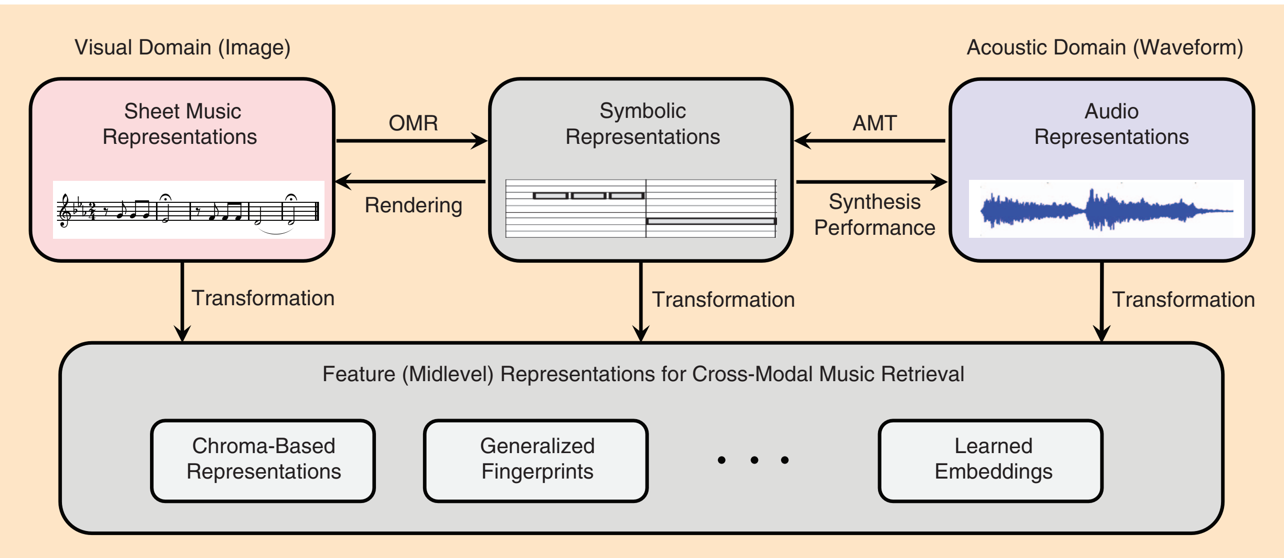

FIGURE 1. The different representations for music data and data transformations relevant for cross-modal music retrieval.

on midlevel representations that exploit specific domain knowledge. As an important example, we first consider midlevel representations that capture musical properties related to pitches and harmony. We then discuss symbolic fingerprints that are based on note-level descriptors. Both of these approaches require expert knowledge in the transformation process. As an alternative, we present an end-to-end learning approach based on deep neural networks, where the idea is to circumvent the explicit definition of a midlevel representation. In the following sections, we address the benefits and limitations of these conceptually different approaches in the context of cross-modal music retrieval.

Chroma-based approach

To make music data algorithmically accessible, traditional music processing tries to extract suitable features that capture relevant key aspects while suppressing irrelevant details. For music-related retrieval and analysis tasks, chroma features have turned out to be a powerful midlevel representation [3], [10].

Because of their central importance in music processing, we give a short introduction to the basics of chroma features, following [3, Ch. 1]. Recall that playing a note on an instrument results in a more or less periodic sound of a certain fundamental frequency. This frequency is closely related to what is called the *pitch* of a note. This notion allows us to order pitched sounds from lower to higher—similar to the keys of a piano keyboard ordered from left to right.

Two notes with fundamental frequencies in a ratio equal to any power of two (e.g., half, twice, or four times) are perceived as very similar or musically/harmonically equivalent, in some sense. This observation leads to the fundamental notion of an octave, which is defined as the interval between one musical note and another with half or double its fundamental frequency. In Western music, the space within one octave is generally subdivided into 12 scale steps with fundamental frequencies equally spaced on a logarithmic frequency axis, resulting in what is known as the *12-tone, equal-tempered scale*. In this scale, each pitch can be separated into two components, which are referred to as the *tone height* (or *octave number*) and the *chroma* (or *pitch spelling attribute*, denoted by $C, C^{\#}, D, \ldots, B$ in Western music notation).

Chroma features rely on this perception of octave equivalence and map absolute pitch into 12 octave-independent pitch classes, where a pitch class consists of all of the pitches that share the same chroma. Thus, a chroma feature is represented by a 12-dimensional vector $x = (x(1), \ldots, x(12))^{\top}$, where $x(1)$ corresponds to chroma C, $x(2)$ to $C^{\#}$, and so on. In the feature extraction step, a given audio signal is converted into a sequence of chroma vectors, also called *chromagrams*, where each vector expresses how the short-time energy of the signal is spread over the 12 chroma bands. A chromagram closely correlates to the melodic and harmonic progression of the music, while exhibiting a high degree of robustness to variations in instrumentation and dynamics.

There are many ways to compute chroma-based features from audio recordings, e.g., by using short-time Fourier transforms (STFTs) in combination with binning strategies [10] or employing suitable multirate filter banks [11]. Furthermore, the properties of chroma features can be significantly changed by introducing suitable pre- and postprocessing steps modifying spectral, temporal, and dynamical aspects. As an example, Figure 2(b) shows two different chromagram variants extracted from a piano audio recording. While the first is a traditional chromagram, the second version is enhanced, such that certain important frequencies that relate to melody notes, as specified by the upper staff, are emphasized—which can be important, e.g., for melody-based retrieval. When given a symbolic music representation, such as MIDI or MusicXML files, it is straightforward to derive chromagrams from the explicitly encoded note parameters (pitches, note onsets, and note durations).

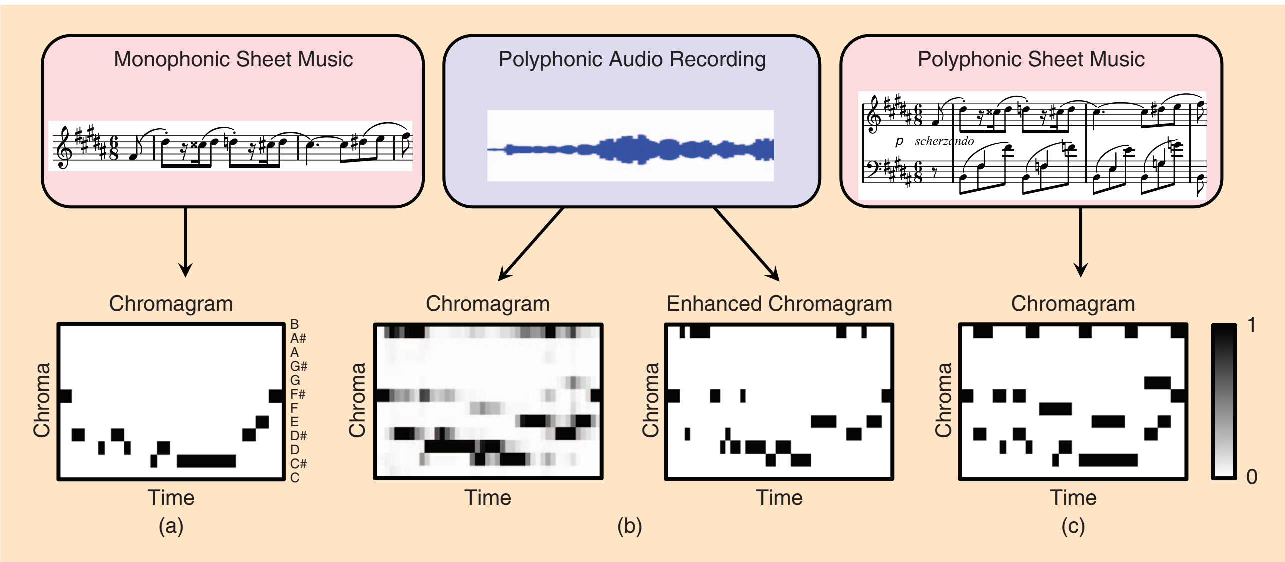

FIGURE 2. Some chromagrams obtained from (a) monophonic sheet music, (b) polyphonic audio representations, and (c) polyphonic sheet music for the beginning of Frédéric Chopin's Nocturne in B Major, op. 9, no. 3.

Figure 2 also shows a symbolic chromagram obtained from a monophonic [Figure 2(a)] and a polyphonic [Figure 2(c)] sheet music representation. While symbolic chromagrams are based on pure note information, audio-based chromagrams tend to be noisy, reflecting the full range of the signal's acoustic properties, including partials, transients, and room acoustics. Still, as also demonstrated by Figure 2, chroma features mainly capture melodic and harmonic properties and are suited to serve as a midlevel feature representation for comparing and relating acoustical and symbolic music.

To demonstrate the applicability and potential of chroma-based features, we consider a cross-modal retrieval scenario motivated by Barlow and Morgenstern's book *A Dictionary of Musical Themes*, published in 1949 [12]. This book contains about 10,000 musical themes of well-known instrumental pieces from the corpus of Western classical music. These monophonic themes (usually four bars long) are typically the most memorable parts of a piece of music. This motivates the retrieval scenario as considered in [13] and [14], where the objective is to retrieve all audio recordings from a music collection that contain a specified musical theme. More formally, let $\mathcal{Q}$ be the collection of musical themes where each element $Q \in \mathcal{Q}$ is regarded as a query. Furthermore, let $\mathcal{D}$ be a set of audio recordings, which we regard as a database collection consisting of documents $D \in \mathcal{D}$. Given a query $Q \in \mathcal{Q}$, the retrieval task is to identify the semantically corresponding documents $D \in \mathcal{D}$.

One approach, as illustrated by Figure 3(a), is to first transform a query Q (possibly using OMR as an intermediate step) and each of the documents D into chromagrams. Based on these midlevel representations, one computes a matching function Δ_D^Q by locally comparing the query chromagram to the audio chromagram using a subsequence variant of dynamic time warping (DTW) [11, Ch. 4]. For each position of the audio recording D, such a matching function indicates the local cost of aligning the query chromagram with a segment ending at that position of the audio chromagram. In other words, each local minimum of Δ_D^Q that is close to the value zero points to a location where the query (musical theme) is similar to a local segment of the document (audio recording). Thus, for a given query, the retrieval task can be solved by computing matching curves for all documents and screening for local minima that are below a certain threshold in these curves. The costs of the local minima yield a natural ranking of the retrieved documents and their relevant sections, which can then be presented in the form of a ranked list [Figure 3(b)].

As detailed in [13] and [14], there are various challenges that need to be addressed, including tempo deviations, OMR extraction errors, musical tunings, key transpositions, and differences in the degree of polyphony between the symbolic query and the audio recordings. For some of these challenges, there already exist reliable compensation strategies. For example, key transpositions are simulated by a cyclic shift of the query's chromagram, or local and global tempo deviations are compensated for by using sequence alignment techniques, such as DTW. Handling differences in the degree of polyphony is still subject to ongoing research. One strategy to bridge the polyphony gap is to first extract the predominant melody of the audio recording using harmonic summation [15] and source-filter models [16]. From the resulting salience representations, enhanced audio chromagrams that better match the monophonic theme may be derived [see Figure 2(b) for an illustration].

Obviously, computing matching curves for each database document results in a retrieval procedure that does not scale to large music collections. Indexing techniques based on short audio excerpts (so-called audio shingles) can help speed up the retrieval procedure [17], [18]. In the next section, we discuss an alternative approach that is based on symbolic fingerprints and permits extremely efficient retrieval.

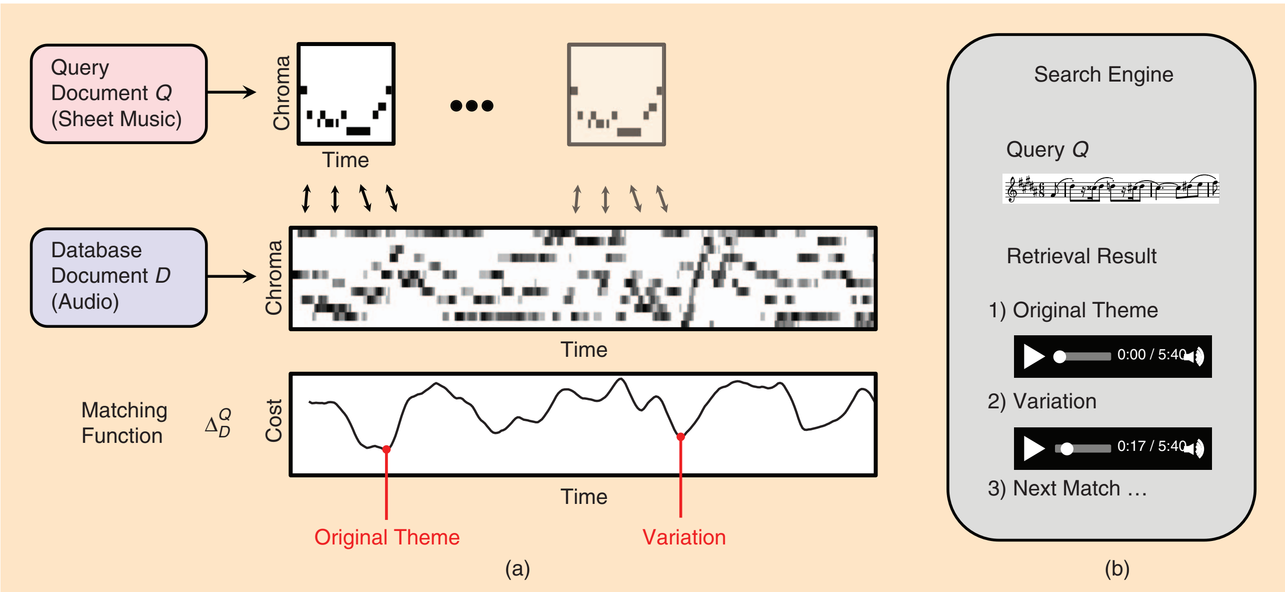

FIGURE 3. (a) An illustration of the matching procedure with chroma-based representations. (b) The costs of the local minima yield a natural ranking of the retrieved documents and their relevant sections, which are shown in the form of a ranked list.

Symbolic fingerprinting approach

We have seen that chroma features are a very convenient mid-level representation for comparing music data of different modalities. One main benefit is that both symbolic and audio data can be easily converted into chromagrams. Furthermore, capturing only the coarse harmonic/melodic progression, chromagrams are highly robust to musical and acoustic variations. However, the reduction onto the chroma level also leads to a loss of valuable information that may be contained in the input data, such as accurate timing and pitch parameters, as encoded by sheet music. As a consequence, chroma-based retrieval strategies often become problematic for short input sequences (e.g., covering only a couple of notes). Furthermore, reducing pitch information to the 12 chroma bands renders the comparison of monophonic and polyphonic versions difficult. An alternative to using chroma-based features is to exploit the high specificity of note parameters and of the resulting time–pitch patterns of occurring notes. To this end, both the visual and acoustic data need to be transformed into the symbolic music domain. In the following, we discuss such an approach, based on symbolic fingerprints, and highlight the resulting benefits and limitations.

Traditionally, in music processing, *audio fingerprinting* refers to methods for identifying exact replicas of audio recordings, which are possibly distorted in some way (e.g., compression artifacts or background noise). For this problem, also known as *audio identification*, powerful algorithms exist and are in everyday use in commercial applications (see, e.g., [1], [3], [4], and [19]). In the identification process, the audio material is compared by means of so-called audio fingerprints, which are compact and discriminative audio features. There are many different ways of designing and computing audio fingerprints, and the suitability of a specific type of fingerprint very much depends on the requirements imposed by the intended application. For example, in the pioneering work by Wang [1], a fingerprinting approach is described that operates on spectral peaks extracted from a time–frequency representation.

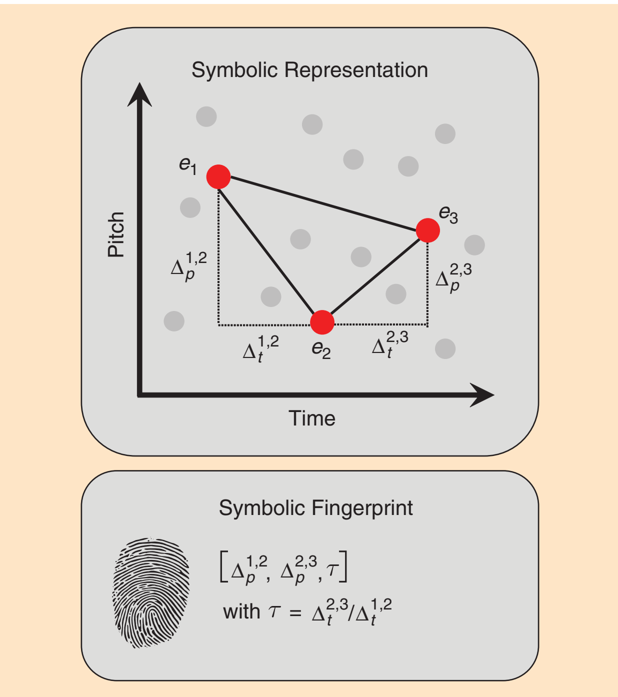

FIGURE 4. An illustration of symbolic fingerprints.

Recent work, such as [19] and [20], has focused on making fingerprinting algorithms more robust to transformation in the time scale (the playback speed of the audio) and the frequency scale (transpositions). Classical fingerprinting approaches, combined with indexing techniques, allow for an identification of audio material that is extremely efficient (scalable to huge fingerprint data sets) and effective (with high precision even for short queries). However, being based on audio-specific spectro-temporal patterns, these techniques are not suited for handling music-specific variations, as required for cross-modal music retrieval or related tasks, such as cover song retrieval [3], [21].

Inspired by classical fingerprinting techniques, Arzt et al. [5], [6] introduced a symbolic fingerprinting approach that allows not only for the identification of exact replicas of recordings but also for fast retrieval of different versions of the same piece of music, including differently performed audio recordings and score representations. In the following, we summarize the main idea of this approach. We start with a symbolic music representation where all note events are encoded explicitly. As illustrated by Figure 4, we assume that each note event $e = (t, p)$ is specified by an onset time t and a pitch p. To obtain fingerprints, we consider triples consisting of three events, $e_1 = (t_1, p_1)$, $e_2 = (t_2, p_2)$, and $e_3 = (t_3, p_3)$, with $t_1 < t_2 < t_3$. For each such triple, we define the time differences $\Delta_t^{1,2} := t_2 - t_1$ and $\Delta_t^{2,3} := t_3 - t_2$ as well as the pitch differences $\Delta_p^{1,2} := p_2 - p_1$ and $\Delta_p^{2,3} := p_3 - p_2$. Furthermore, we set $\tau := \Delta_t^{2,3} / \Delta_t^{1,2}$. Finally, a symbolic fingerprint is defined to be a list of the following numbers:

$$[\Delta_p^{1,2}, \Delta_p^{2,3}, \tau]. \tag{1}$$

Considering time and pitch relations in a relative fashion, each fingerprint is invariant with regard to musical transpositions (pitch shifts) and tempo changes. To obtain local descriptors, fingerprints are computed only from note events within a certain neighborhood, typically a few seconds. This not only facilitates short query lengths, but also reduces the number of fingerprints to be stored in the database. Also, observe that, since each individual fingerprint encodes relative timing information, we need to assume that the onset times of a triple are distinct. As a result, simultaneous note events (as occurring in a chord) may not be encoded by a single fingerprint. However, such cooccurring events can be captured by considering several fingerprints. In summary, being discriminative yet compact descriptors of fixed length, such fingerprints have turned out to be suitable for indexing symbolically encoded music data.

We now discuss how the symbolic fingerprints can be used for cross-modal music retrieval. As a challenging example scenario, we consider a combined sheet music identification and score-following application tailored to piano music [6]. Given a short excerpt of an audio recording (used as a query), the task is to identify the underlying sheet music document as well as the exact score position (see Figure 5). Accordingly, the database

$\mathcal{D}$ for this task consists of sheet music representations of all of the pieces to be potentially identified. In a preprocessing step, all sheet music documents $D \in \mathcal{D}$ are first transformed into a suitable symbolic format (e.g., by applying OMR or by extracting note parameters from a MusicXML file). From this encoding, symbolic fingerprints are extracted for each document by considering all of the possible triples of note events that obey certain constraints. For example, to avoid a combinatorial explosion, one typically imposes constraints in the form of minimum and maximum values for the time differences $\Delta_t^{1,2}$ and $\Delta_t^{2,3}$. The resulting fingerprints, along with links to suitable metadata (e.g., corresponding piece and sheet music positions), are stored in a fingerprint database that is equipped with efficient search structures based on indexing techniques.

Similarly, an incoming audio query is also transformed into a set of symbolic fingerprints. This, however, involves a nontrivial transcription step to convert the recording into a symbolic representation. In general, automatic music transcription is still an unsolved problem—in particular, for polyphonic music recordings with many different instruments (e.g., orchestral music) (see [9], [22], and [23]). In the case of single-instrument polyphonic music (such as piano music), state-of-the-art algorithms provide reasonable, albeit far from perfect, transcriptions. In our scenario, we employ a recent transcription algorithm based on a recurrent neural network [22]. The use of bidirectional hidden layers enables the system to better model the context of the notes, which exhibit a very characteristic envelope during their decay phase, particularly for piano music. The network was trained on a collection of several hundred piano pieces recorded on various pianos, virtual and real (see [22] for further details).

To make the transcriber also applicable in online scenarios, instead of preprocessing the whole piece of audio at one time, the signal is split into blocks that consist of several subsequent frames centered around the current frame. Using such blocks, each covering roughly 210 ms of audio, is a tradeoff between maintaining the system's ability to model the context of the notes and keeping the introduced delay to a minimum. The network outputs a transcription of the audio query consisting of a list of note onsets and pitches, which can be further transformed into a set of audio fingerprints. Finally, the score fingerprint database is searched for subsets that approximately fit the query's set of audio fingerprints. The best matching subset in the database yields the sheet music document, along with the score position (Figure 5).

In contrast to chroma-based midlevel representations, symbolic fingerprints are compact, possess a high discriminative power, and are well suited for indexing techniques. As a result, these techniques scale well to large amounts of data in terms of memory requirements, accuracy, and efficiency. However, there is also a price to be paid. The necessary transcription from audio signals into the symbolic domain is a hard problem that is solvable well enough only for certain classes of music (e.g., piano music recorded under reasonable acoustic conditions). Even though a small proportion of the fingerprints extracted from the query may suffice to identify the correct piece, symbolic fingerprinting may fail if the input representation becomes too noisy.

For general music recordings, including those with many instruments (e.g., an orchestra), there is still a long way to go. Here, one requires strategies that better adapt to the multitude of musical aspects, including harmony, melody, rhythm, dynamics, and instrumentation. In this context, recent advances in deep

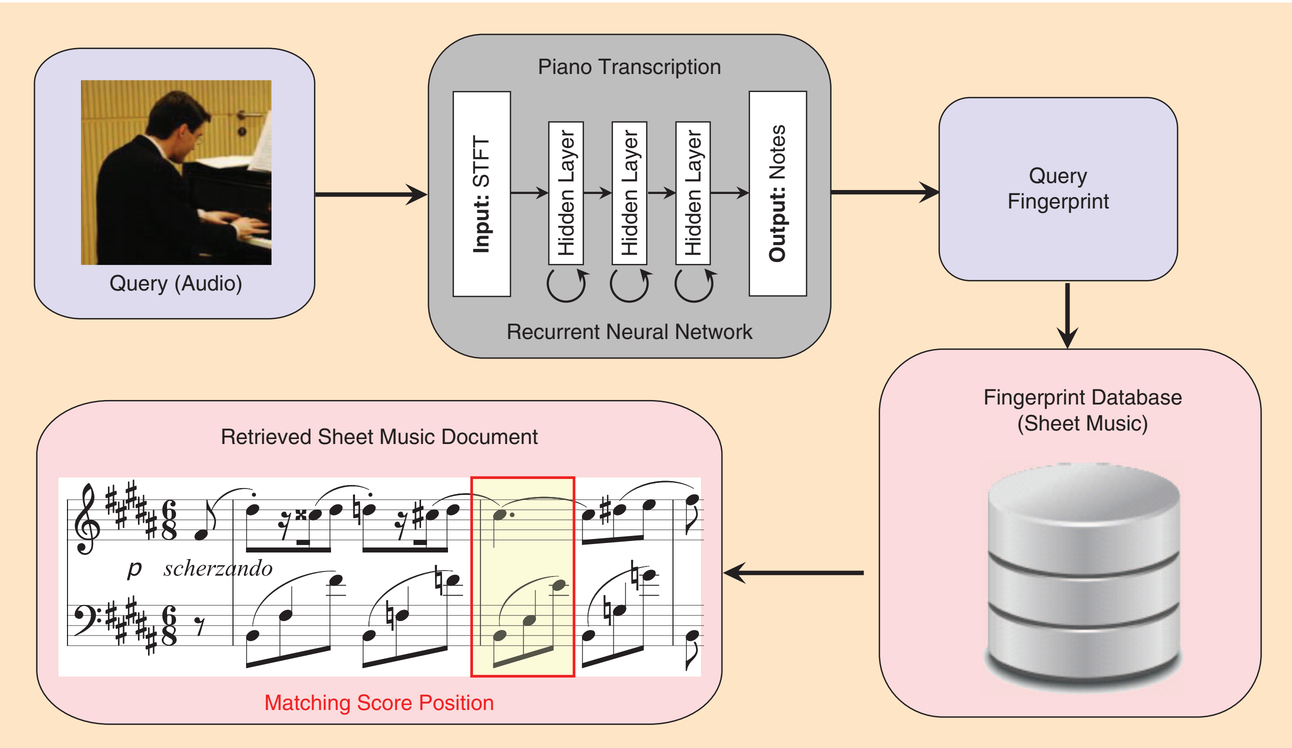

FIGURE 5. An illustration of cross-modal retrieval via piano transcription and symbolic fingerprinting. (Photo of Werner Goebl courtesy of Clemens Chmelar.)

learning may help to achieve further progress in this area. In the following section, we discuss such a deep-learning approach that tries to learn sheet music and audio correspondences directly from raw input representations, without the need for midlevel representations that explicitly exploit musical knowledge.

Deep-learning approach

In the previous sections, we have seen two more-traditional approaches for linking audio and sheet music data using musically informed midlevel representations—one using chroma features and one symbolic fingerprints. Such representations not only require expert knowledge at the design stage but are also problematic when relying on error-prone preprocessing steps, such as automatic music transcription on the audio side or optical music recognition on the sheet music side. As an alternative, we now present a methodology to directly learn correspondences between audio data and sheet music images from a set of training observations, thus circumventing the explicit definition of a midlevel representation.

This approach builds on the current success of artificial neural networks, today often referred to as *deep learning*, which have proven to be powerful tools for automatic feature learning [24]. Given snippets of sheet music images and corresponding audio excerpts, we introduce a cross-modal neural network that learns an embedding space in which both modalities are represented as low-dimensional vectors [7]. In this embedding space, cross-modal music retrieval can then be easily performed by using a simple similarity measure.

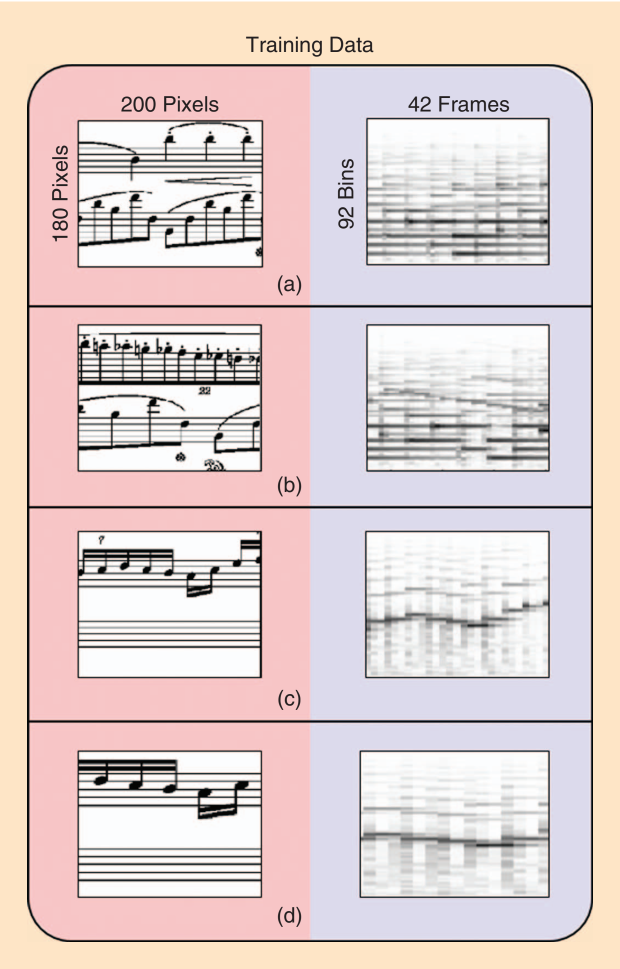

FIGURE 6. (a)–(d) Four training pairs, each consisting of a sheet music snippet and an audio excerpt. The pair in (d) was obtained from the pair in (c) by applying data augmentation techniques.

The general principle of supervised feature learning is to learn latent representations in an end-to-end fashion from a set of raw training observations. Such approaches are not only generally applicable but also have the advantage of automatically adapting the learned representations to the given problem. One limitation, however, is that supervised learning requires a sufficiently large set of training data to arrive at models that generalize well to unseen data.

In our scenario, we need training pairs that consist of sheet music snippets and corresponding audio excerpts. Typical examples as used in our system are shown in Figure 6(a)–(d). Note that, for creating such training pairs, we need to first establish correspondences between individual pixel locations of the note heads in a score and their respective counterparts (note onset events) in the corresponding audio recording. Establishing the correspondences can be done either in a manual annotation process or by relying on synthetic training data generated from digital sheet music formats, such as Musescore (https://musescore.com/) or Lilypond (http://lilypond.org/). Based on these relationships, one can generate corresponding snippets of sheet music images (in our case, 180×200 pixels) and short excerpts of audio (in our case, represented by log-frequency spectrograms with 92 bins $\times$ 42 frames). These are the pairs presented to the multimodal network for training.

To improve the generalization ability of the resulting network, one can further apply data augmentation techniques to synthetically increase the effective size of the training set and better account for relevant data variability. In this setting, different transformations for sheet music augmentation (e.g., image scaling and translation) and audio augmentation (e.g., using different sound fonts and tempo scaling) are applied. At this point, we emphasize that data augmentation is a crucial component for learning cross-modal representations that generalize to unseen music, especially when limited data are available.

In this process, augmenting the data set using data transformations is conceptually different from and more promising than automatically generating random scores. First, rendering (synthesizing) sheet music typically results in images with strong regularities (e.g., the same scale or perfectly centered staff lines). By applying image transformations, these regularities are disturbed, thus making the embedding networks robust to small distortions, like those that occur in realistic scenarios, e.g., images of printed sheet music scanned under different conditions and sheet music originating from different publishers using varying type settings. Second, note that music (and, hence, also sheet music) follows musical rules. Therefore, augmentation by adding randomly generated music may distort the inherent data distribution of realistic music, which may have a negative impact on embedding space learning.

Based on such training pairs, the retrieval task is formulated as an embedding problem, with the aim of learning a joint embedding space of the two different modalities [7]. This approach is inspired by a similar text-to-image retrieval problem, where a pairwise ranking loss is introduced as an optimization target [25]. In the following, let $(\mathbf{x}, \mathbf{y})$ denote a training pair consisting of a sheet image snippet $\mathbf{x}$ and an audio excerpt $\mathbf{y}$. As shown in Figure 7, the network consists of two separate pathways. One processes $\mathbf{x}$ and is represented by the function f_α, where α is the network parameters to be trained. The other pathway, which is represented by the function g_β, with parameters β, is responsible for $\mathbf{y}$. The two functions map $\mathbf{x}$ and $\mathbf{y}$, respectively, to a k-dimensional vector, where $k \in \mathbb{N}$ denotes the embedding dimension.

To define the loss function, we need a scoring function $s: \mathbb{R}^k \times \mathbb{R}^k \rightarrow \mathbb{R}$ to measure similarity in the embedding space. In our scenario, s is chosen to be the cosine measure, i.e., the cosine of the angle between two vectors. Furthermore, for each given training pair $(\mathbf{x}, \mathbf{y})$, we assume that there are $L \in \mathbb{N}$ additional contrasting examples $\mathbf{y}_\ell$ for $\ell \in \{1, 2, \ldots L\}$. Then, the pairwise ranking loss (also known as the *max-margin hinge loss* [25]) is defined as follows:

$$\mathcal{L}_{\text{rank}} = \sum_{(\mathbf{x},\mathbf{y})} \sum_{\ell=1}^{L} \max\{0, \gamma - s(f_\alpha(\mathbf{x}), g_\beta(\mathbf{y})) + s(f_\alpha(\mathbf{x}), g_\beta(\mathbf{y}_\ell))\}. \quad (2)$$

In this formula, the first sum is taken over a set of training pairs $(\mathbf{x}, \mathbf{y})$ called a *training batch*, where each such pair comes with a separate set of contrasting examples (in practice, all remaining audio samples of the current training batch). The purpose of this loss function is to encourage an embedding where the distance between matching samples $(\mathbf{x}, \mathbf{y})$ is lower than the distance between mismatching samples $(\mathbf{x}, \mathbf{y}_\ell)$. The parameter $\gamma \in \mathbb{R}_+$ is the margin parameter of the hinge loss and, in combination with the maximum function, imposes a penalty on poorly embedded training pairs. More precisely, if the elements of a matching pair $(\mathbf{x}, \mathbf{y})$ are already close in the learned embedding space and, in addition, the elements of the mismatching pairs $(\mathbf{x}, \mathbf{y}_\ell)$ are embedded far enough apart, the second term in the max operator goes below zero, and the respective pairs do not contribute to the overall loss. On the contrary, if the embedded elements of a matching pair are still far apart, the second term is usually above zero and will yield a substantial contribution to the overall loss.

In the training stage, the pairwise ranking loss in (2) is minimized via stochastic gradient descent with respect to the network parameters α and β. Once the networks represented by the functions f_α and f_β are learned, the elements of the matching pairs are close in the embedding space, while those of contrasting pairs are far apart (in the ideal case). For further details concerning the network topology and the training procedure, we refer to [7] and [26].

Given this learned embedding space, cross-modal retrieval can be performed based on a retrieval-by-embedding paradigm (Figure 7). It is important to note that, although the network pathways are trained simultaneously on pairs of sheet music snippets and audio excerpts, both modalities are required only at training time. At application time, the two network pathways operate independently of each other.

This has huge benefits in view of the cross-modal retrieval applications discussed in the previous sections. For example, in sheet music identification and score-following applications, one can first compute an embedding of an entire collection of sheet music snippets using the image embedding function f_α. The resulting k-dimensional embedding vectors can be further processed

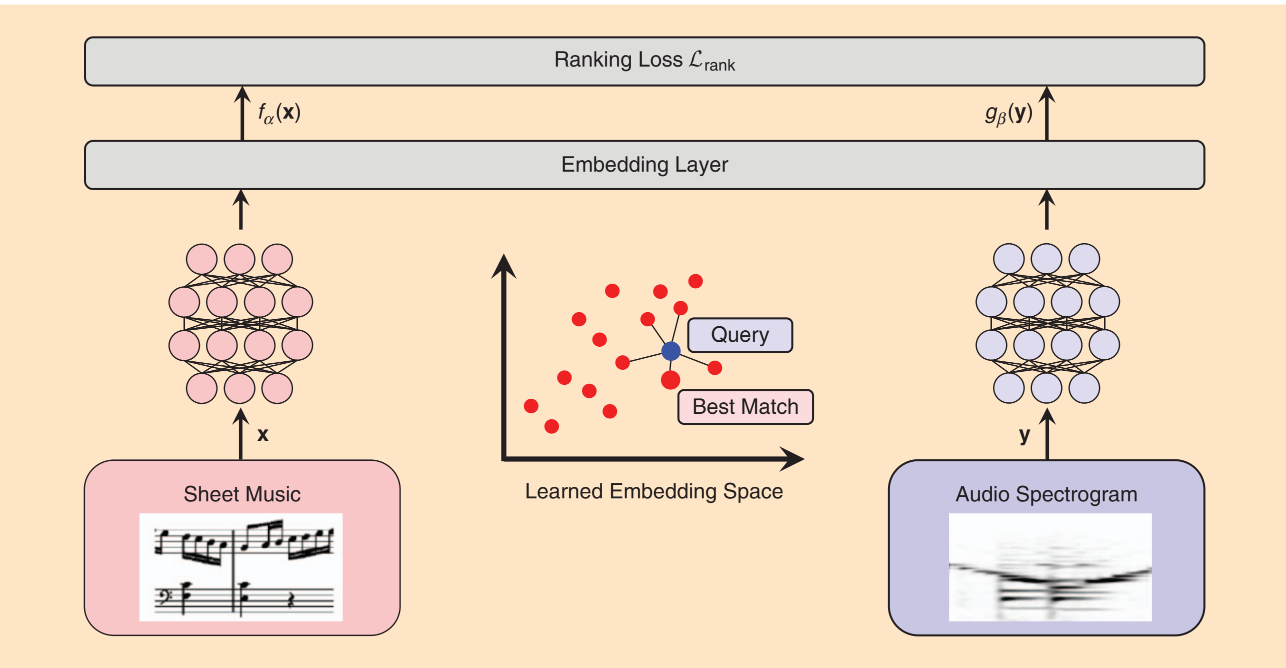

FIGURE 7. An illustration of the network used for learning a cross-modal embedding space. At application time, the learned functions f_α and g_β are used to project the sheet music snippets and audio excerpts, respectively, to the joint embedding space.

and stored using suitable index structures that allow for an efficient neighborhood search. Then, given an audio excerpt as a query, the search can be performed by first projecting the query into the joint embedding space using the audio embedding function g_β of the network and then performing a nearest-neighbor search.

The experiments reported in [7], which are based on 26 classical piano pieces (including the composers Bach, Haydn, Beethoven, and Chopin) and roughly 20,000 training pairs, demonstrate that the end-to-end learning approach yields reasonable retrieval results for sheet music of medium complexity (e.g., piano scores) and synthesized audio (used for evaluation to establish the ground truth). In particular, combining retrieval based on snippets/excerpts with a subsequent majority voting step, the approach is capable of correctly relating sheet music and audio recordings on the piece level with high accuracy. However, on the level of sheet music snippets (consisting of one or two bars) and audio excerpts (lasting a couple of seconds), the proposed system is not yet competitive with engineered approaches that exploit musical knowledge or are based on symbolic representations (see the approaches presented in the two previous sections).

At this stage, one may conclude that, even when comparatively scant training data are available, it is still possible to use deep-learning models by designing appropriate, task-specific data augmentation strategies. Initial experiments showed that, when trained on only one composer, the model started to generalize to unseen scores by other composers. Therefore, we may expect that the described model will develop its full potential when provided a comprehensive data set that consists of millions of training pairs comprising different editions and layouts of sheet music and different recorded performances.

Applications and future directions

In this article, we have introduced different approaches for cross-modal music retrieval aiming to bridge the gap between various music representations. Despite the remaining challenges, current technology enables a variety of music navigation and browsing applications that have educational and commercial relevance. For example, in the context of modern digital music libraries, cross-modal retrieval strategies have become an important component for content-based analysis, synchronization, indexing, and navigation in heterogeneous music collections [27]. Other cross-modal applications are often subsumed under the umbrella of score following, where the computer listens to a live performance and tries to read along in the sheet music. The output of a score-following algorithm can be used for highlighting the current measure in a digital score, automatic page turning (a page turner being a person who turns sheet music pages for a soloist during a performance), or automatic accompaniment (see, e.g., [28]).

In the following, we describe one specific example of a prototype system to give a concrete impression of what is already possible. The Piano Music Companion is a versatile system focused on piano music, intended to be useful for both pianists and music lovers [29]. The system is able to identify, follow, and synchronize live performances of classical piano music in real time. The Piano Music Companion is a permanent listener. Whenever the pianist starts playing (regardless of which piece or where within the piece), the companion identifies the piece, the position within the score, and continues to follow along. This allows triggering various actions synchronized to the performed music—for instance, the current position in the sheet music is highlighted.

While this is helpful for the performer and listener, further information about important themes, musical structures, and chords can be provided. In a concert setting, the system may also give hints to the listener about what to focus on at specific moments. The system may also give additional background information on the piece or composer, while telling the user where to acquire additional recordings of the current or related pieces.

Technically, the Piano Music Companion is based on two main components that run in parallel. The first is responsible for identifying the piece being played. To this end, symbolic fingerprinting, as described earlier, is used to continuously match the most recently detected notes of the live performance to a database of symbolically encoded sheet music (Figure 5). Currently, the database includes the complete solo piano works by Chopin and the complete Beethoven piano sonatas, and consists of roughly 1 million notes in total (about 330 pieces). Once the piece and the rough position within the sheet music representation have been identified, the actual score following is conducted using a separate chroma-based tracking procedure, which is realized as an online variant of the matching procedure shown in Figure 3. In this way, the system combines the strengths of the respective components. The fingerprinting component is flexible, it works globally across different pieces, and it scales over large data sets. However, since the fingerprinter's transcription step is in general faulty, the component often leads to outliers and local misalignments. This weakness is compensated for by the separate chroma-based tracking component, which is less efficient but introduces a high degree of robustness (due to the chroma features). This second component is applied only locally, for tracking the score once the piece and the rough position are known.

By combining these two components, the Piano Music Companion continuously reevaluates its hypothesis and tries to match the current input stream to the complete database. Thus, even if the musician suddenly jumps to a different score position or starts playing a completely different piece, the system is able to follow as long as the piece is part of the database. The Piano Music Companion is also highly tolerant of deviations from the notated score (due to performance errors, transcription errors, or intentional variations) and to tempo changes. A video demonstration of our system can be found at https://www.youtube.com/watch?v=SUBtND_MJZs.

Our vision is to extend this scenario toward a Complete Classical Music Companion. Such a system would be at one's fingertips anytime and anywhere, possibly as an application on a mobile device. Whatever source of music—be it a live concert, a digital video disc, a video stream, or a radio program—whatever piece of classical music, whatever instrumentation, and whoever the performers, the companion would detect what it is listening to and inform about the music, the historical context of the piece, famous interpretations, and so forth, thus guiding the user in the listening process.

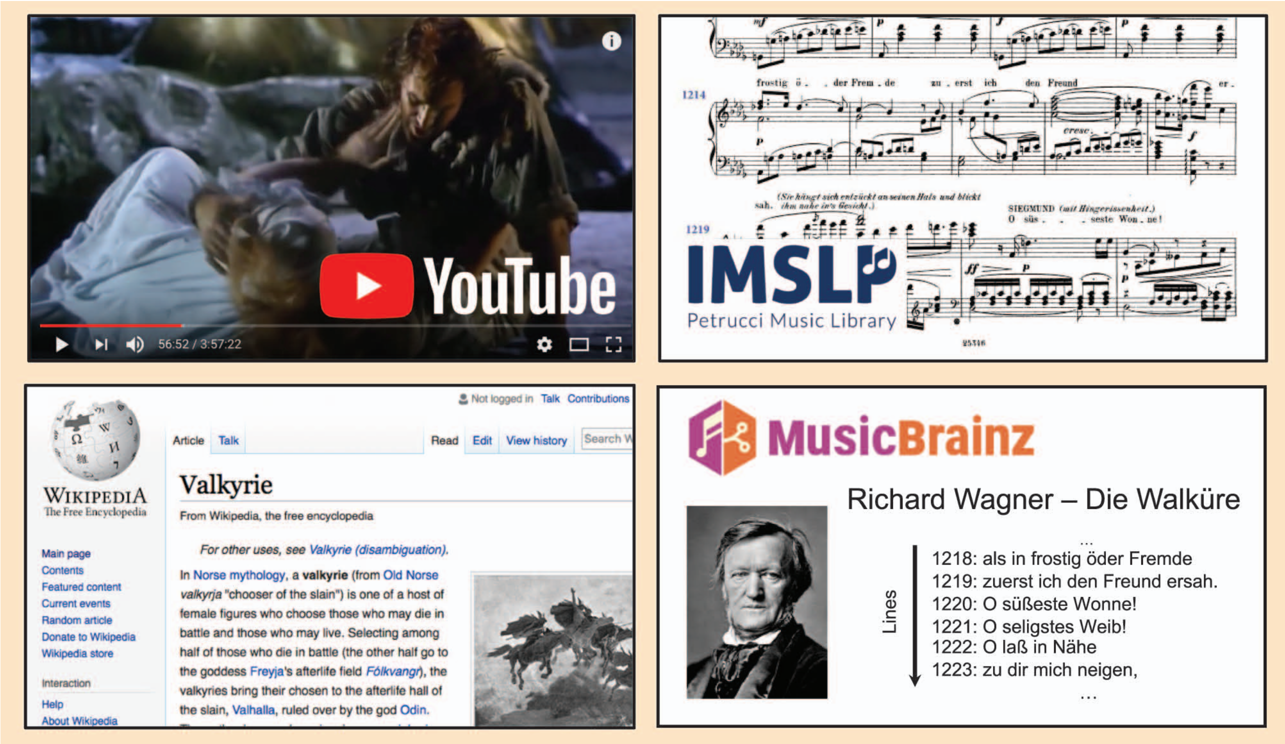

FIGURE 8. Some sources of freely accessible music data distributed via the Internet. (MusicBrainz image courtesy of the Warner Music Group.)

Beyond this specific music companion scenario, cross-modal music processing techniques are essential for organizing and searching information distributed via the Internet (Figure 8). For example, there are millions of digitized pages of sheet music publicly available on sites such as the International Music Score Library Project (IMSLP) Petrucci Music Library (http://imslp.org/). On the audio side, widely accessible music and video platforms, such as YouTube, offer a vast and rapidly growing corpus of music recordings. Furthermore, music-related websites, as available at *Wikipedia*, contain information of various types, including text, score, images, and audio. Finally, community-driven encyclopedias, such as MusicBrainz (https://musicbrainz.org/), collect and provide music-related metadata in a systematic fashion. For example, structured websites can be used to automatically derive text-, score-, and audio-based queries to look for other musically related documents on the Web [13], [30]. Furthermore, YouTube videos may be automatically enriched with manually or automatically generated musical annotations, as recently demonstrated in [31].

This rich application potential, demonstrated in concrete application scenarios, makes cross-modal music retrieval a very active research field that also drives research on other music processing tasks. For instance, one key challenge is to improve transformation techniques, such as OMR and AMT, which are a bottleneck in many of the current approaches. Also, deep neural networks that directly learn to relate different data modalities are a very promising alternative that is currently receiving a lot of attention. We hope that these prospects will serve as an inspiration for the signal processing community to pay even more attention to music as a promising (and beautiful) object of study.

Acknowledgments

The International Audio Laboratories Erlangen is a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg and the Fraunhofer-Institut für Integrierte Schaltungen IIS. Meinard Müller and Stefan Balke are supported by the German Research Foundation (DFG MU 2686/11-1). Matthias Dorfer acknowledges financial support for early versions of this work by the Austrian Ministries Federal Ministry of Transport, Innovation, and Technology and the Federal Ministry of Science, Research, and Economy and the Province of Upper Austria via the Competence Centers for Excellent Technologies, Software Competence Center Hagenberg. The work of Andreas Arzt and Gerhard Widmer was supported by the European Research Council (ERC) under the European Union's Horizon 2020 Framework Programme (H2020, 2014–2020)/ERC Advanced Grant Agreement 670035, project Con Espressione.

Authors

Meinard Müller (meinard.mueller@audiolabs-erlangen.de) received his diploma degree in mathematics and his Ph.D. degree in computer science from the University of Bonn, Germany, in 1997 and 2001, respectively. Since 2012, he has held a professorship in semantic audio signal processing at the International Audio Laboratories Erlangen, Germany. His recent research interests include music processing, music information retrieval, and audio signal processing. He has coauthored more than 100 peer-reviewed scientific papers and is the author of a monograph, *Information Retrieval for Music and Motion* (Springer, 2007), and textbook, *Fundamentals of Music Processing* (Springer, 2015).

Andreas Arzt (andreas.arzt@jku.at) received his B.S. degree in software and information engineering and his M.S. degree in computational intelligence from the Vienna University of Technology, Austria, in 2006 and 2008, respectively. In 2016, he completed his Ph.D. degree on the topic "Flexible and Robust Music Tracking" in the Department of Computational Perception, Johannes Kepler University, Linz, Austria, where he is currently an assistant professor. His research interests include real-time music tracking, music synchronization, and music identification.

Stefan Balke (stefan.balke@audiolabs-erlangen.de) received his diploma in electrical engineering from Leibniz Universität, Hanover, Germany, in 2013. In early 2018, he completed his Ph.D. degree in the Semantic Audio Signal Processing Group headed by Prof. Meinard Müller at the International Audio Laboratories Erlangen, Germany. Afterward, he joined the Institute of Computational Perception led by Gerhard Widmer at Johannes Kepler University, Linz, Austria. His research interests include music information retrieval, machine learning, and multimedia retrieval.

Matthias Dorfer (matthias.dorfer@jku.at) studied medical informatics at Vienna University of Technology, Austria, with a focus on computational image analysis and machine learning. After finishing his master's degree studies, he worked for two years as an industrial researcher in the field of medical image analysis. Since April 2015, he has been a Ph.D. student in the Department of Computational Perception, Johannes Kepler University, Linz, Austria, under the supervision of Prof. Gerhard Widmer. His research interests are artificial neural networks, especially multimodality deep learning, and audiovisual representation learning.

Gerhard Widmer (gerhard.widmer@jku.at) received his Ph.D. degree in computer science in 1989 from Technische Universität Wien, Austria. He is a professor and the head of the Department of Computational Perception, Johannes Kepler University, Linz, Austria, and head of the Intelligent Music Processing and Machine Learning Group, Austrian Research Institute for Artificial Intelligence, Vienna. His research interests include artificial intelligence, machine learning, and intelligent music processing. He is a fellow of the European Association for Artificial Intelligence, has been awarded Austria's highest research awards [the START Prize (1998) and the Wittgenstein Award (2009)], and currently holds a European Research Council Advanced Grant for research on computational models of expressivity in music.

References

[1] A. Wang, "An industrial strength audio search algorithm," in *Proc. Int. Society Music Information Retrieval Conf.*, 2003, pp. 7–13.

[2] F. Kurth, M. Müller, C. Fremerey, Y. Chang, and M. Clausen, "Automated synchronization of scanned sheet music with audio recordings," in *Proc. Int. Conf. Music Information Retrieval*, 2007, pp. 261–266.

[3] M. Müller, *Fundamentals of Music Processing*. Berlin: Springer-Verlag, 2015.

[4] P. Cano, E. Batlle, T. Kalker, and J. Haitsma. (2005). A review of audio fingerprinting. J. VLSI Signal Process. [Online]. 41(3), pp. 271–284. Available: http://dx.doi.org/10.1007/s11265-005-4151-3

[5] A. Arzt, S. Böck, and G. Widmer, "Fast identification of piece and score position via symbolic fingerprinting," in *Proc. Int. Society Music Information Retrieval Conf.*, 2012, pp. 433–438.

[6] A. Arzt, G. Widmer, and R. Sonnleitner, "Tempo- and transposition-invariant identification of piece and score position," in *Proc. Int. Society Music Information Retrieval Conf.*, 2014, pp. 549–554.

[7] M. Dorfer, A. Arzt, and G. Widmer, "Learning audio-sheet music correspondences for score identification and offline alignment," in *Proc. Int. Society Music Information Retrieval Conf.*, 2017, pp. 115–122.

[8] D. Byrd and J. G. Simonsen, "Towards a standard testbed for optical music recognition: Definitions, metrics, and page images," *J. New Music Res.*, vol. 44, no. 3, pp. 169–195, 2015.

[9] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. (2013). Automatic music transcription: challenges and future directions. *J. Intelligent Inform. Syst.* [Online]. *41(3)*, pp. 407–434. Available: http://dx.doi.org/10.1007/s10844-013-0258-3

[10] E. Gómez, "Tonal description of music audio signals," Ph.D. dissertation, Universitat Pompeu Fabra, Barcelona, Spain, 2006.

[11] M. Müller, *Information Retrieval for Music and Motion*. Berlin: Springer-Verlag, 2007.

[12] H. Barlow and S. Morgenstern, *A Dictionary of Musical Themes*, rev. 3rd ed. New York: Crown, 1975.

[13] S. Balke, S. P. Achankunju, and M. Müller, "Matching musical themes based on noisy OCR and OMR input," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2015, pp. 703–707.

[14] S. Balke, V. Arifi-Müller, L. Lamprecht, and M. Müller, "Retrieving audio recordings using musical themes," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2016, pp. 281–285.

[15] J. Salamon, J. Serrà, and E. Gómez. (2013). Tonal representations for music retrieval: From version identification to query-by-humming. Int. J. Multimedia Inform. Retrieval. [Online]. 2(1), pp. 45–58. Available: http://dx.doi.org/10.1007/s13735-012-0026-0

[16] J. S. Juan J. Bosch, R M. Bittner and E. Gómez, "A comparison of melody extraction methods based on source-filter modelling," in *Proc. Int. Conf. Music Information Retrieval*, 2016, pp. 571–577.

[17] M. A. Casey, C. Rhodes, and M. Slaney, "Analysis of minimum distances in high-dimensional musical spaces," *IEEE Trans. Audio, Speech, Language Process. (2006–2013)*, vol. 16, no. 5, pp. 1015–1028, 2008.

[18] P. Grosche and M. Müller, "Toward characteristic audio shingles for efficient cross-version music retrieval," in *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, 2012, pp. 473–476.

[19] R. Sonnleitner and G. Widmer, "Robust quad-based audio fingerprinting," *IEEE Trans. Audio, Speech, Language Process. (2006–2013)*, vol. 24, no. 3, pp. 409–421, 2016.

[20] J. Six and M. Leman, "Panako: A scalable acoustic fingerprinting system handling time-scale and pitch modification," in *Proc. Int. Conf. Music Information Retrieval*, 2014, pp. 259–264.

[21] J. Serrà, E. Gómez, and P. Herrera, "Audio cover song identification and similarity: Background, approaches, evaluation and beyond," in *Advances in Music Information Retrieval* (Studies in Computational Intelligence Series), Z. W. Ras and A. A. Wieczorkowska, Eds. Berlin: Springer-Verlag, 2010, vol. 274, pp. 307–332.

[22] S. Böck and M. Schedl, "Polyphonic piano note transcription with recurrent neural networks," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2012, pp. 121–124.

[23] S. Sigtia, E. Benetos, and S. Dixon, "An end-to-end neural network for polyphonic piano music transcription," *IEEE/ACM Trans. Audio, Speech, Language Process.*, vol. 24, no. 5, pp. 927–939, 2016.

[24] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2015.

[25] R. Kiros, R. Salakhutdinov, and R. S. Zemel. (2014). Unifying visual-semantic embeddings with multimodal neural language models. arXiv. [Online]. Available: https://arxiv.org/abs/411.2539

[26] M. Dorfer, J. Schlüter, A. Vall, F. Korzeniowski, and G. Widmer, "End-to-end cross-modality retrieval with CCA projections and pairwise ranking loss," *Int. J. Multimedia Inform. Retrieval*, vol. 7, no. 2, pp. 117–128, 2017.

[27] 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. Digital Libraries*, vol. 12, no. 2–3, pp. 53–71, 2012.

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

[29] A. Arzt, S. Böck, S. Flossmann, H. Frostel, M. Gasser, C. C. S. Liem, and G. Widmer, "The piano music companion," in *Proc. European Conf. Artificial Intelligence*, 2014, pp. 1221–1222.

[30] M. Gasser, A. Arzt, T. Gadermaier, M. Grachten, and G. Widmer, "Classical music on the web: User interfaces and data representations," in *Proc. Int. Conf. Music Information Retrieval*, 2015, pp. 571–577.

[31] S. Balke, C. Dittmar, J. Abeßer, K. Frieler, M. Pfleiderer, and M. Müller, "Bridging the gap: Enriching YouTube videos with jazz music annotations," *Front. Dig. Humanities*, vol. 5, Feb. 2018. doi: 10.3389/fdigh.2018.00001.

SP