Chapter 1
OngaCREST Project: Building a Similarity-Aware Information Environment for a Content-Symbiotic Society

Masataka Goto

Abstract The purpose of this project is to develop fundamental technologies for building a similarity-aware information environment in which people are able to know similarities among vast amounts of media content. This environment helps establish a "*content-symbiotic society*" in which media content such as music and video can be created and used in innovative, but ethical ways. Furthermore, by developing technologies for enhancing content appreciation and creation, we aim to promote a society in which people can actively engage in content appreciation and creation, and a content culture that respects past content and emphasizes experiencing emotion. We developed various types of technologies for supporting music appreciation and creation, such as *Songle*, *Songrium*, and *TextAlive* web services, and made those services open to the public.

Keywords Music information processing · Music-understanding technologies · Content-appreciation support technologies · Content-creation support technologies · Similarity and typicality

1.1 Introduction

The amount of digital content that can be accessed by people has been increasing and will continue to do so in the future. This is desirable since more people can enjoy more content. However, problems have arisen. For content creators, their own works become easily buried among huge amounts of past works. For listeners and viewers, it is becoming more difficult to find their favorite content. Furthermore, as content increases, the amount of similar content is also increasing. As a result, creators will be more concerned that their content might invite unwarranted suspicion of plagiarism. All kinds of works are influenced by existing content, and it is difficult to avoid the unconscious creation of content partly similar in some way to prior content.

M. Goto (✉)
National Institute of Advanced Industrial Science and Technology (AIST),
Tsukuba, Ibaraki, Japan
e-mail: m.goto@aist.go.jp

T. Nishida (ed.), *Human-Harmonized Information Technology, Volume 2*,
DOI 10.1007/978-4-431-56535-2_1

It is therefore desirable to clarify to what extent content can be similar but still acceptable. However, the ability of human beings to judge such similarity is limited. Judging the similarity between two things in front of one's eyes or ears is possible. But the speed of this judging is slow. Searching for similar content among a million items cannot be done for all practical purposes. Moreover, while humans are able to make accurate judgments based on past experiences, their ability is limited when it comes to judging "*typicality*" (commonness)—determining what will happen probabilistically from an overall phenomenon. For example, when an uncommon event is frequently observed, people tend to wrongly assume that it is likely to occur. And when an event that is by its nature frequent is not encountered, people tend to wrongly assume that it is rare. Because it is not quantitatively possible to view and listen to all accessible content, it is not humanly possible to carry out appropriate judgment that encompasses all content.

Consequently, if there is a high risk that one's work will be denounced as being similar to someone else's as a result of the monotonic increase in content, this could lead to a society in which it is difficult for people to create and share content with peace of mind. This could happen with the coming of an "age of billions of creators" in which anyone can enjoy creating and sharing works. Content is made up of a variety of elements. However, despite the existence of common elements (elements with high probability of appearing), there is the problem of fruitless suspicion of plagiarism due to misunderstanding of "the question of originality" that arises simply when these elements resemble other content. In the first place, creative activities build upon past content. Highly common elements and expressions should be appropriately recognized, shared, and used between creators and consumers as knowledge common to humankind.

In light of the above, we started the *OngaCREST Project*, a five-year research project to build a technological foundation that not only specialists and but also general users can use to answers the questions "What is similar here?" and "How often does this occur?" With the spread of such a technological foundation in the future, people could continue creating and sharing content with peace of mind. Furthermore, by developing *content-appreciation support technologies*, we want to allow people to actively encounter content and appreciate them. We also want to make it easy for even non-specialists to easily enjoy the content creation process by developing *content-creation support technologies* that enable "highly typical" elements (such as chord progressions and conventional genre-dependent practices) to be used as knowledge common to humankind.

The OngaCREST Project is officially entitled "*Building a Similarity-aware Information Environment for a Content-Symbiotic Society.*" It was carried out as a fiscal year 2011-selected research project (Research Director: Masataka Goto; Group Leaders: Masataka Goto, Shigeo Morishima, Satoshi Nakamura, and Kazuyoshi Yoshii) in the research area of "*Creation of Human-Harmonized information Technology for Convivial Society*" of the CREST (Core Research for Evolutional Science and Technology) funding program provided by the JST (Japan Science and Technology Agency). The main types of media content targeted in the project are music and music-related videos, such as music videos and dance videos, which are representative and important media content. It carried out basic and applied research related to music-understanding technologies and enabled end users to use research results as web services so harmonious interactions between humans and the information environment could be extracted. In this chapter, I introduce main research achievements of this project.

1.2 Project Overview

The overview of the OngaCREST Project is shown in Fig. 1.1. We call a society in which relationships between humans and content and between past content and future content are rich and capable of sustained development a "*content-symbiotic society.*" To realize such a society, this project aims to build a similarity-aware information environment that promotes an awareness of similarity—i.e., allows people to understand similarity—among a huge amount of media content. If a content-symbiotic society can be realized, media content can be richly and soundly created and used. In short, people will be able to continue to create and share content with peace of mind. Anyone will be able to actively encounter and appreciate content and, furthermore, enjoy creating content easily.

We hope to contribute to the creation of a culture that can co-exist and co-prosper with past content while paying appropriate respect to it. This will become possible by supporting a new music culture that enables creators to take delight in finding their content being reused in much the same way that academic researchers take delight in finding their articles being cited. We feel that the value of content cannot be measured by the extent to which it is not similar to other content—pursuing originality at all costs in content does not necessarily bring joy to people. Fundamentally, content has value by inducing an emotional and joyous response in people. We would like to make it a matter of common sense that content with emotional appeal and high-quality form has value. In addition, we would like to see conditions in which content

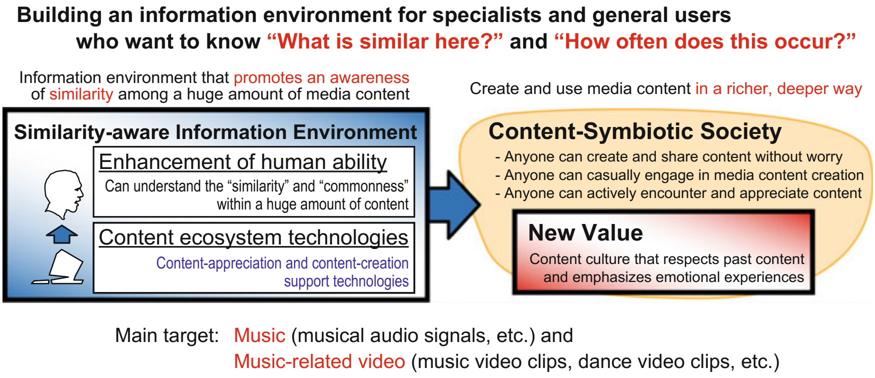

Building an information environment for specialists and general users
who want to know "What is similar here?" and "How often does this occur?"
Information environment that promotes an awareness of similarity among a huge amount of media content
Create and use media content in a richer, deeper way
Similarity-aware Information Environment
Enhancement of human ability
Can understand the "similarity" and "commonness" within a huge amount of content
Content ecosystem technologies
Content-appreciation and content-creation support technologies
Content-Symbiotic Society
- Anyone can create and share content without worry
- Anyone can casually engage in media content creation
- Anyone can actively encounter and appreciate content
New Value
Content culture that respects past content and emphasizes emotional experiences
Main target: Music (musical audio signals, etc.) and
Music-related video (music video clips, dance video clips, etc.)

Fig. 1.1 Overview of OngaCREST Project

that explicitly refers to many existing works and was build on the basis of those works can be considered valuable, similar to the situation with academic papers. We can anticipate that our similarity-aware information environment will form a new content culture that respects past content and gives greater importance to emotionally touching experiences.

In our similarity-aware information environment, not only technologies that can estimate “similarity” and “typicality” but also “*content ecosystem technologies*” will play a critical role in fostering a rich content ecosystem. We therefore also conducted research and development of technologies that comprehensively support appreciation and creation of music content as content ecosystem technologies. In a content ecosystem that is achieved by these technologies, the aim is to sustainably expand “*content circulation*” to foster new content from past content in collaboration with end users. To accomplish this, we developed “*content ecosystem web services,*” which mainly targeted music-related content available on the web, and field-tested them by releasing the services to the public. If a single web service has too different purposes and too diverse functions, it is difficult for users to understand. We therefore implemented several separate web services so that each service can have a different function of content appreciation and creation. These services are organically linked to form the entire content ecosystem web services.

In Sect. 1.3 below I first introduce research achievements related to content-appreciation support technologies that were first pursued for research of content ecosystem technologies. Next, in Sect. 1.4 I introduce research achievements related to content-creation support technologies realized by using content-appreciation support technologies. Then, in Sect. 1.5 I introduce our achievements in researching similarity and typicality estimation technologies, which we tackled in parallel with the above efforts. Finally, in Sect. 1.6 I discuss some related topics and in Sect. 1.7 I conclude by summarizing this chapter and discussing future directions.

1.3 Content-Appreciation Support Technologies

For appreciation support functions of the content ecosystem web services, support to appreciate both the internal aspect and external aspect of songs—in short, appreciation support focused on the content within a single song and appreciation support focused on the relationships between multiple songs—is important. To accomplish the former goal of providing appreciation support for the content within a song, we developed *Songle* (http://songle.jp), an active music listening service that uses music-understanding technologies to enrich ways of listening to music on the web. Furthermore, on the basis of Songle, we developed *Songle Widget* (http://widget.songle.jp), a web-based multimedia development framework for music-synchronized control. For the latter goal of providing appreciation support targeting the relationships between multiple songs, we developed *Songrium* (http://songrium.jp), a music browsing support service that visualizes relationships between content with web mining technologies and music-understanding technologies. We have released these services to the general public for field-testing, and have continued to research and develop functional extensions. In this section I describe Songle, Songle Widget, and Songrium in detail.

Besides these services, we developed a wide range of music-appreciation support technologies. These include:

- LyricListPlayer, a consecutive-query-by-playback interface for retrieving similar word sequences from different song lyrics [32]
- PlaylistPlayer, an interface using multiple criteria to change the playback order of a music playlist [33]
- LyricsRadar, a lyrics retrieval system based on latent topics of lyrics, which are analyzed and visualized by using Latent Dirichlet Allocation (LDA) [39]
- A music retrieval system based on vocal timbre analysis using Latent Dirichlet Allocation (LDA) and cross-gender vocal timbre similarity [35]
- An active music listening system that allows users to enjoy music by using timbre replacement of harmonic and drum components [30]

We also developed the following music-video-appreciation support technologies:

- SmartVideoRanking, a music video search system by mining emotions from time-synchronized comments on a video-sharing service [43]
- ExploratoryVideoSearch, a music video search system based on coordinate terms and diversification [42]

With these achievements, various types of user-led search, recommendation, browsing, and content appreciation by making use of music-understanding technologies and similarity became possible. Users could not only more freely appreciate music content to encounter favorite songs from a huge amount of music content, but also enjoy music content by changing it to reflect their personality.

1.3.1 Songle

Songle (http://songle.jp) [13] is a web service that allows users to enjoy music by using *active music listening interfaces* [12]. Active music listening is a way of listening to music through active interactions. In this context the word *active* does not mean that the listeners create new music but means that they take control of their own listening experience. For example, an active music listening interface called *SmartMusicKIOSK* [11] (Fig. 1.2) has a chorus-search function that enables a user to directly access his or her favorite part of a song (and to skip other parts) while viewing a visual representation of its music structure. This facilitates deeper understanding of the music structure and is useful for trial listening, but before we developed Songle, the general public has not had the chance to use such research-level active music listening interfaces in their daily lives.

Toward the goal of enriching music listening experiences, Songle uses automatic music-understanding technologies to estimate (analyze) musical elements (music

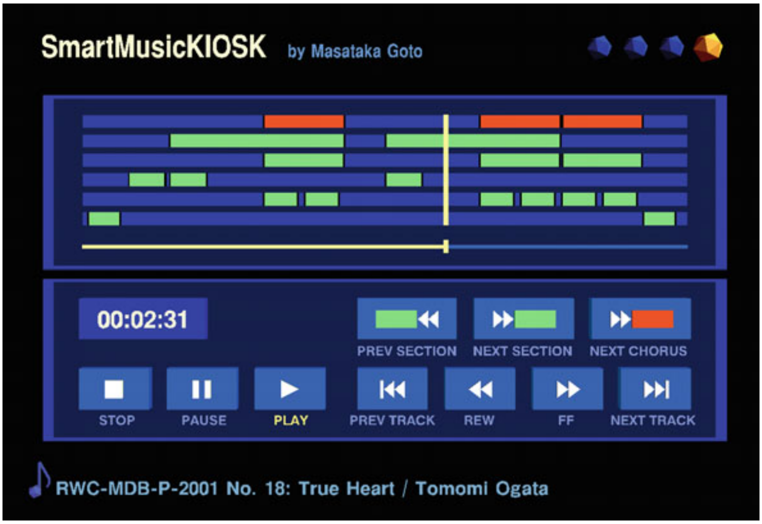


Fig. 1.2 SmartMusicKIOSK [11]: A user can actively listen to various parts of a song while moving back and forth as desired. The upper window shows a graphical representation of the entire song structure that is estimated by a chorus-section detection method [11] and consists of chorus sections (the *top row*) and repeated sections (the five *lower rows*). On each row, colored sections indicate similar (repeated) sections. Clicking directly on a colored section plays that section. A user can jump and listen to the chorus with just a push of the next-chorus button in the lower window

scene descriptions [10]) of songs (audio files) publicly available on the web. A Songle user can enjoy playing back a musical piece while seeing the visualization of the estimated descriptions. Four major types of descriptions are automatically estimated and visualized for content-based music browsing: music structure (chorus sections and repeated sections), beat structure (musical beats and bar lines), melody line (fundamental frequency (F0) of the vocal melody), and chords (root note and chord type). Songle implements all functions that the interface of SmartMusicKIOSK had and lets a user jump and listen to the chorus by just pushing the next-chorus button. Songle thus makes it easier for a user to find desired parts of a piece.

With a focus on popular songs with vocals, Songle has already analyzed more than 1,050,000 songs on music-sharing services such as *SoundCloud* (http://soundcloud.com) and *Piapro* (http://piapro.jp), video-sharing services such as *YouTube* (http://www.youtube.com) and *Niconico* (http://www.nicovideo.jp/video_top), and various web sites distributing MP3 files of music. The user's browser plays back music streamed directly from the original web site. Since Songle does not distribute any music, if a song is removed from the original web site, it is not possible to play back its song on Songle. In addition to contributing to the enrichment of music listening experiences, Songle serves as a showcase in which everybody can experience music-understanding technologies with a lot of songs and understand their nature: for example, what kinds of music or sound mixture are difficult for the technologies to handle. Because of the variety of music on the web and the complexity of sound mixtures, however, automatic music-understanding technologies cannot avoid making some errors.

Songle therefore provides a crowdsourcing error-correction interface that enables users to help improve its service by correcting music-understanding errors. As shown in Fig. 1.3, each user can see the music-understanding visualizations on a web browser, where a moving cursor indicates the audio playback position. A user who finds an error while listening can easily correct it by selecting from a list of candidate alternatives, or by providing an alternative description. The resulting corrections are then shared and used to immediately improve user experience with the corrected piece.

Songle supports three main functions: retrieving, browsing, and annotating songs. The retrieval and browsing functions facilitate deeper understanding of music, and the annotation (error correction) function allows users to contribute to the improvement of musical elements. The improved descriptions can lead to a better user experience of retrieving and browsing songs.

1.3.1.1 Retrieval Function

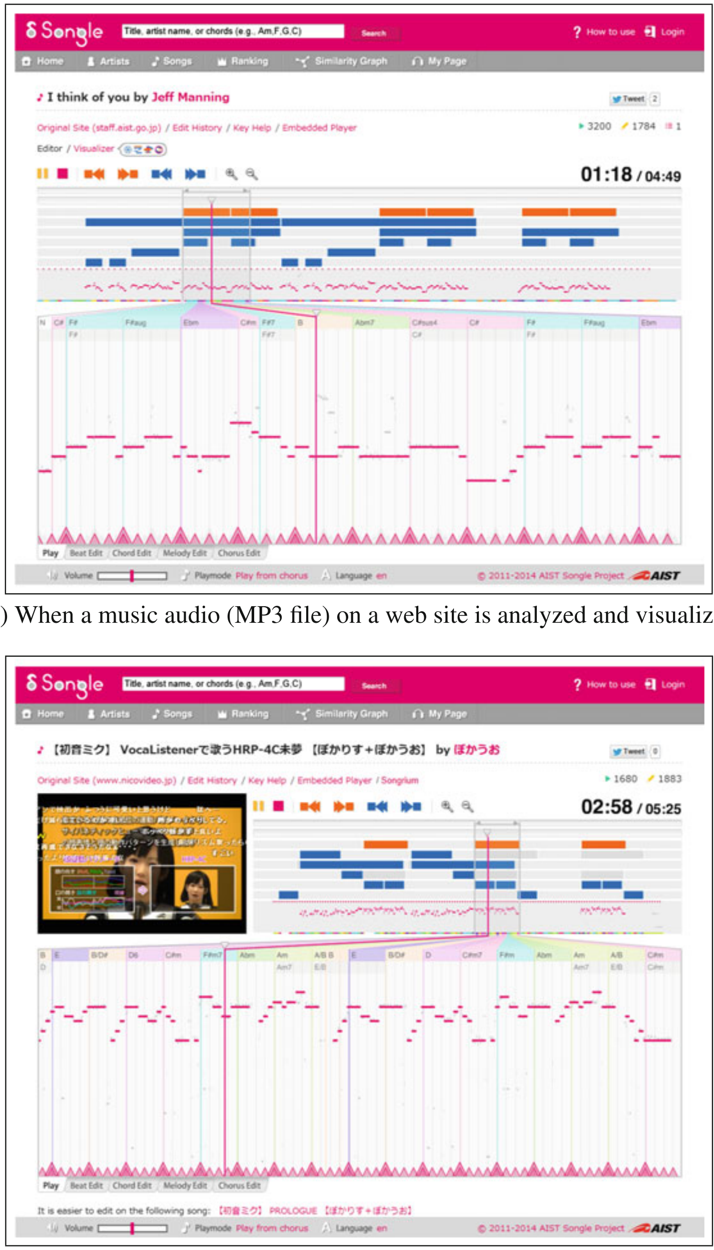
This function enables a user to retrieve a song by making a text search for the song title or artist name, by making a selection from a (recommended) list of songs or artists, or by making a chord-progression search in which a user can provide a favorite chord progression to find songs including its progression. Following the idea of an active music listening interface called *VocalFinder* [5], which finds songs with similar vocal timbres, Songle provides a similarity graph of songs so that a user can retrieve a song according to vocal timbre similarity. The graph is a radially connected network in which nodes (songs) of similar vocal timbre are connected to the center node (a user-specified song). By traversing a graph while listening to nodes, a user can find a song having the favorite vocal timbre.

By selecting a song, the user switches over to the within-song browsing function.

1.3.1.2 Within-Song Browsing Function

This function provides a content-based playback-control interface for within-song browsing by visualizing musical elements as a music map shown in Fig. 1.3 and the lower half of Fig. 1.7. In the music map, the upper window is the global view showing the entire song and the lower window is the local view magnifying the selected region. The music map consists of the following four types of musical elements:

1. *Music structure (chorus sections and repeated sections)*
   In the global view, the SmartMusicKIOSK interface [11] is shown below the playback controls including the buttons, time display, and playback slider. Just like SmartMusicKIOSK shown in Fig. 1.2, the music structure consists of chorus


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
I think of you by Jeff Manning
Tweet
Original Site (staff.aist.go.jp) / Edit History / Key Help / Embedded Player
3200
1784
1
Editor / Visualizer
01:18 / 04:49
Play
Beat Edit
Chord Edit
Melody Edit
Chorus Edit
Volume
Playmode Play from chorus
Language en
© 2011-2014 AIST Songle Project
AIST
【初音ミク】VocaListenerで歌うHRP-4C未夢【ぼかりす＋ぼかうお】 by ぼかうお
Original Site (www.nicovideo.jp) / Edit History / Key Help / Embedded Player / Songrium
1680
1883
02:58 / 05:25
It is easier to edit on the following song: 【初音ミク】PROLOGUE【ぼかりす＋ぼかうお】

(a) When a music audio (MP3 file) on a web site is analyzed and visualized

(b) When a music video on a video-sharing service (Niconico) is analyzed and visualized

Fig. 1.3 Screen snapshots of Songle's main interface for music playback. This interface visualizes the content of a song (musical elements) as a music map and allows the user to control playback position freely. The horizontal axis represents time. The global view in the upper portion of the screen presents the music structure of the entire song, and the local view in the lower portion shows an enlarged display of the song interval selected in the global view

sections (the top row) and repeated sections (the five lower rows). On each row, colored sections indicate similar sections. A user can jump and listen to a chorus section with just a push of its section or the next-chorus button.

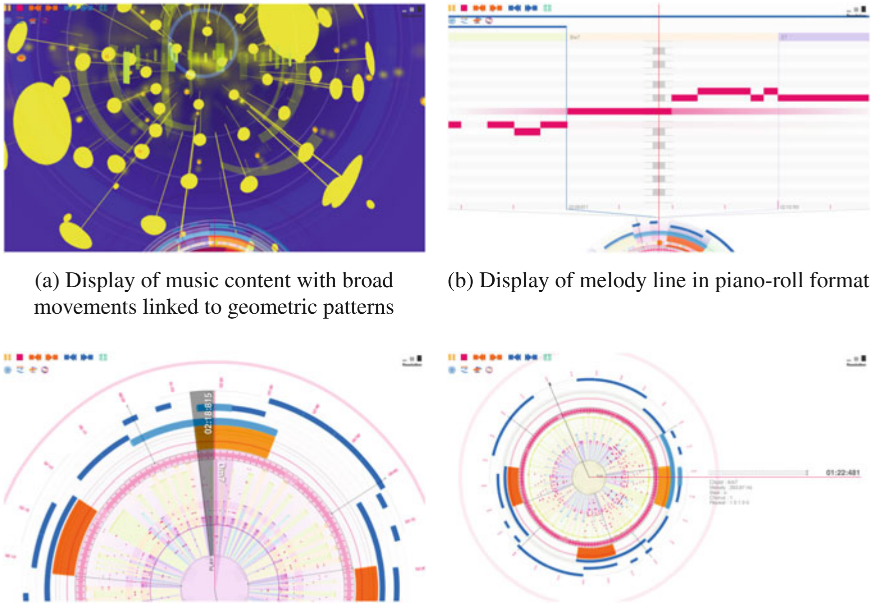
2. *Beat structure (musical beats and bar lines)*
   At the bottom of the local view, musical beats corresponding to quarter notes are visualized by using small triangles. The top of each triangle indicates its temporal position. Bar lines are marked by larger triangles.
3. *Melody line (F0 of the vocal melody)*
   The piano roll representation of the melody line is shown above the beat structure in the local view. It is also shown in the lower half of the global view. For simplicity, the fundamental frequency (F0) can be visualized after being quantized to the closest semitone.
4. *Chords (root note and chord type)*
   Chord names are written in the text at the top of the local view. Twelve different colors are used to represent twelve different root notes so that a user can notice the repetition of chord progressions without having to read chord names.

The music map makes it easy for a user without musical expertise to learn about the existence of the musical elements, the relationships between them, and their respective roles in the song. When a music audio (MP3 file) on a web site is played back, Songle can also visualize the music map in the four more attractive interactive display modes shown in Fig. 1.4.

1.3.1.3 Annotation Function (Crowdsourcing Error-correction Interface)

This function allows users to add annotations to correct any estimation errors. Here, annotation means describing the contents of a song, either by modifying the estimated descriptions or by selecting the correct candidate if it is available. In the local view, a user can switch between editors for four types of musical elements.

1. *Music structure* (Fig. 1.5a)
   The beginning and end points of every chorus or repeated section can be adjusted. It is also possible to add, move, and delete sections.
2. *Beat structure* (Fig. 1.5b)
   Several alternative candidates for the beat structure can be selected at the bottom of the local view. If none of the candidates are appropriate, each beat position or bar line can be changed directly. For fine adjustment, the audio can be played back with click tones at beats and bar lines.
3. *Melody line* (Fig. 1.5c)
   Note-level correction is possible on the piano-roll representation of the melody line. A user can specify a temporal region of a note and then adjust its F0 with semitone resolution. The synthesized melody line can be played back along with the original song to make it easier to check its correctness.



(a) Display of music content with broad movements linked to geometric patterns

(b) Display of melody line in piano-roll format

(c) Display of musical elements in semicircular format

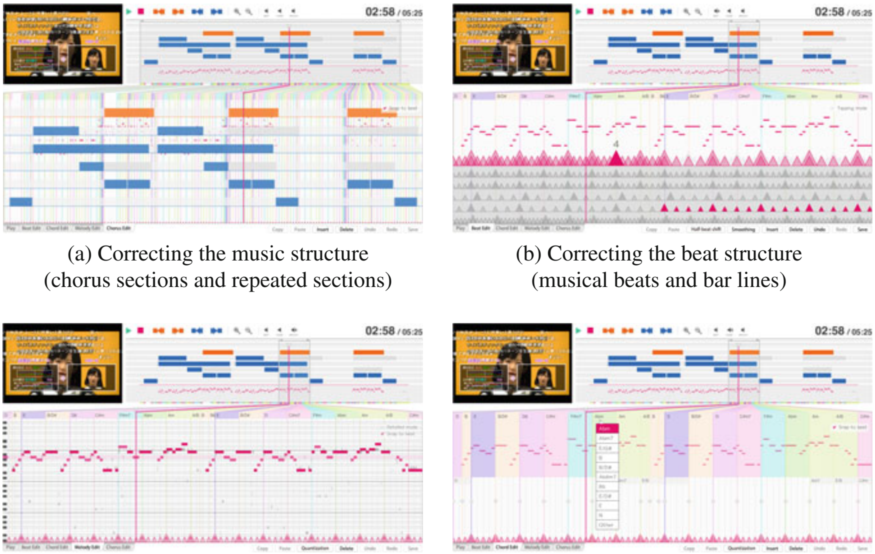
(d) Disk-shaped “all-encompassing” display of musical elements

Fig. 1.4 Four interactive display modes of Songle’s music visualizer for animating musical elements

4. *Chords* (Fig. 1.5d)
   A user can correct chord names by choosing from candidates or by typing in chord names, and each chord boundary can be adjusted. As with beats and melody lines, it is possible to synthesize chords to be played along with the original song.

When the music-understanding results are corrected by users, the original automatically annotated values are visualized as trails with a different color (gray in Fig. 1.6) that can be distinguished by users. These trails are important to prevent excessive evaluation of the automatic music-understanding performance after the user corrections. Songle also makes a complete history of changes (corrections), and in this regard, functions are provided to enable any user to compare the musical elements before and after changes and to return to any point in the past. In this sense, we provided the “annotation version control” system. This has served as an effective deterrent to vandalism because even if some users should make inappropriate changes deliberately, annotations before the vandalism can easily be recovered.

Note that users can simply enjoy active music listening without correcting errors. We understand that it is too difficult for some users to correct the above descriptions (especially, chords). Moreover, users are not expected to correct all errors, only some according to each user’s interests.



(a) Correcting the music structure (chorus sections and repeated sections)

(b) Correcting the beat structure (musical beats and bar lines)

(c) Correcting melody line (F0 of the vocal melody)

(d) Correcting chords (root note and chord type)

Fig. 1.5 Screen snapshots of Songle's annotation function (crowdsourcing error-correction interface) for correcting musical elements. This is an efficient, web-based annotation interface (editor) that allows the user to make corrections by candidate selection or direct editing

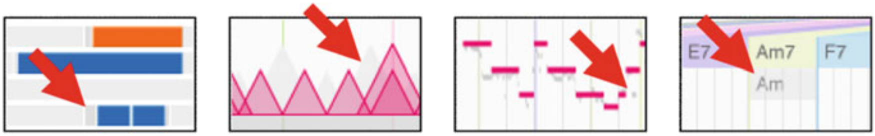


Fig. 1.6 Trails of original music-understanding results (colored in *gray*) remain visible after error corrections on Songle

1.3.1.4 Implementation of Music-Understanding Technologies for Songle

The four types of musical elements are estimated as follows.

1. *Music structure*
   This is estimated by using the chorus-section detection method *RefraiD* [11] which focuses on popular music. By analyzing relationships between various repeated sections, the RefraiD method tries to detect all the chorus sections in a song and estimate both ends of each section. It also has an advantage to detect modulated chorus sections.

2. *Beat structure*
   The beats are estimated using a hidden Markov model (HMM) with 43 tempo states, each having 18 to 60 sub-states corresponding to the beat phase of different tempi. In each tempo a beat is modeled as a left-to-right HMM in which only some states have non-deterministic transition probabilities to allow for tempo fluctuations or tempo changes. The emission probability of a sub-state is calculated via the cosine similarity between a comb filter and an onset detection function.
   The bar lines are estimated using harmonic cues. First, tatum-synchronous bass and treble chromagrams are extracted using NNLS Chroma [26]. Second, a chord detection model based on the chromagrams calculates posterior probabilities of chord changes. Using a sliding window, we compute the cosine similarity between the chord change probabilities and several different bar patterns that cover the 3/4, 4/4 and 6/8 meters and all possible bar phases. Then, the cosine similarities at each frame are normalized and used as emissions in another HMM similar to the beat-tracking model.
3. *Melody line*
   This is estimated by using the F0 estimation method for the vocal melody [7], which is implemented by extending the predominant-F0 estimation method *PreFEst* [10]. This method focuses on the vocal melody by evaluating a GMM-based vocal probability for each F0 candidate estimated by PreFEst. Moreover, vocal activity detection was implemented by using a method described in [6].
4. *Chords*
   Songle transcribes chords using 14 chord types: major, major 6th, major 7th, dominant 7th, minor, minor 7th, half-diminished, diminished, augmented, and five variants of major chords with different bass notes: /2, /3, /5, /b7, and /7. The resulting 14 types × 12 root notes = 168 chords and one ‘no chord’ label are estimated using an HMM approach on the same tatum-synchronous chromagram used for the bar-line estimation. Chord changes are allowed to happen only on beats.
   We also include knowledge from the bar-line estimation and a key estimate. We model the key in a simple separate HMM with three different key scales: major, natural minor, and harmonic minor. Every key state has observation probabilities for all different chords, based on an expert function [27]. The posterior probability obtained from the HMM is then used to weight the chord probabilities for the chord HMM. During Viterbi decoding we use the bar-line estimates for dynamic transition probability weighting in order to encourage chord changes at bar lines.

1.3.2 Songle Widget

Songle Widget [14] is a web-based multimedia development framework that makes it possible to control computer-graphic animation and physical devices such as lighting devices and robots in synchronization with music publicly available on the web. As shown in Fig. 1.7, Songle Widget is implemented by using Songle

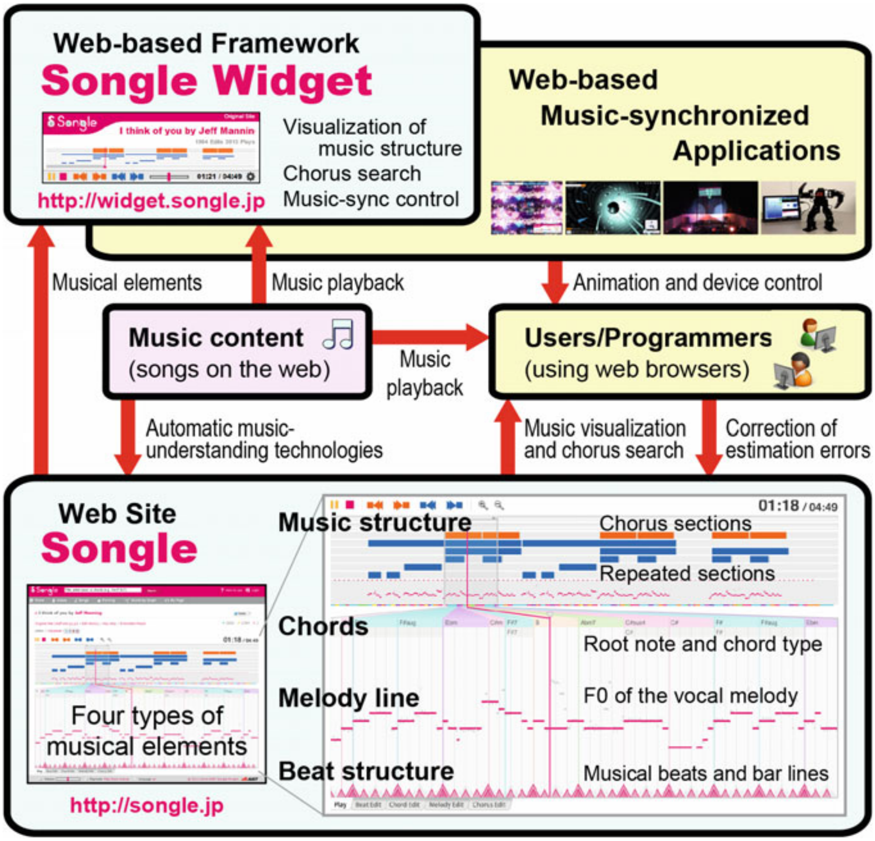

Web-based Framework
Songle Widget
Visualization of music structure
Chorus search
Music-sync control
http://widget.songle.jp
Web-based Music-synchronized Applications
Musical elements
Music playback
Animation and device control
Music content (songs on the web)
Music playback
Users/Programmers (using web browsers)
Automatic music-understanding technologies
Music visualization and chorus search
Correction of estimation errors
Web Site
Songle
Four types of musical elements
http://songle.jp
Music structure
Chorus sections
Repeated sections
Chords
Root note and chord type
Melody line
F0 of the vocal melody
Beat structure
Musical beats and bar lines
01:18 / 04:49

Fig. 1.7 Overview of Songle Widget framework (http://widget.songle.jp) using Songle web service (http://songle.jp)

(http://songle.jp) [13] described in Sect. 1.3.1. Songle Widget can use the four important types of musical elements (music structure, beat structure, melody line, and chords) to trigger changes in the motion and color of animation, or in the control of physical devices.

Since music can be easily and effectively combined with various types of content, it is often used when showing images, movies, and stories, and controlling physical devices such as robots and lighting devices. Synchronization is crucial when music is combined with other content, and to rigidly synchronize animation or physical devices with music, people usually have to annotate temporal positions of target events (musical elements related to the synchronization) in the music. This manual annotation is time-consuming, however.

Automatic music analysis is therefore useful for rigid synchronization. While there are music visualizers built into existing media players that can show music-synchronized (music-sync) animation of geometric patterns, their music analysis is usually based on the amplitude or spectrogram of audio signals. Such analysis is too basic to reflect various musical elements such as musical beats, bar lines (downbeats), chorus sections, and chord changes.

Songle Widget therefore makes it easy to develop web-based applications with such rigid music synchronization. Since Songle has already annotated the four types of musical elements for more than 1,050,000 songs on music- or video-sharing services, those songs can readily be used by music-synchronized applications. Since each musical element is represented as a series of time-stamped events (e.g., beat positions), Songle Widget can compare the time stamps with the current playback position to trigger a user program while playing back a music audio or video. To use a music video clip on video-sharing services for music-sync applications, Songle Widget uses the official embedded YouTube/Niconico player and its API to get the current playback position (elapsed time) as Songle does.

While playing back a music video clip on YouTube, for example, music-sync applications enable humanoid robot devices or animated human-like computer-graphic characters to dance to music, they control multiple lighting devices projecting various types of music-sync lighting patterns onto a stage-like space, and they show graphical objects whose changes in motion, size, and color are synchronized with changes in music.

For high-quality synchronization, we can take full advantage of the crowdsourcing error-correction interface on Songle. Any error corrections made to the musical elements can be instantly reflected on all applications using Songle Widget. This is effective when applications require error-free annotation.

1.3.2.1 Songle Widget Interface

Figure 1.8 shows the user interface of Songle Widget, which allows a compact dedicated player to be embedded in any web page for music-sync web applications. The outstanding features here are that it enables music-sync applications to instantly access the musical elements for more than 1,050,000 songs, which was hitherto difficult to achieve without music-understanding technologies.

To facilitate the development based on Songle Widget, we provide a template to write the JavaScript source code using the Songle Widget API so that programmers can simply add and modify codes for each event. For example, if a user code for showing a visual effect is written for the event corresponding to the bar line, its effect is automatically shown at the beginning of each bar. Programmers could also write an additional code to change all visual effects into more intense ones during chorus sections. Programmers without knowledge of music-related programming can thus achieve music-sync control quite easily.

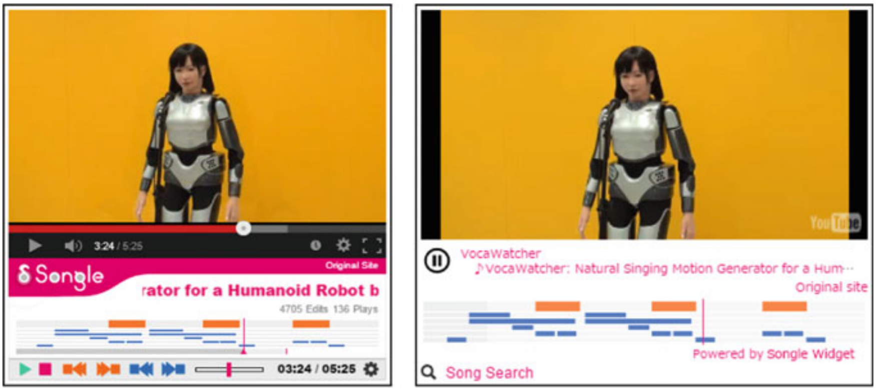


Fig. 1.8 Screenshots of Songle Widget user interface with different appearances, which should be embedded in a web page for music synchronization

1.3.2.2 Music-Synchronized Applications

Since we have made the Songle Widget framework open to the public, various music-sync applications using Songle Widget have been developed by us and by third parties. Seven of them are chosen to represent example applications.

1. **Two-dimensional Music-sync Animation**
   The Songle Widget framework has been used since August 2012 to draw music-sync animation in the background of various web pages including personal home-pages and blogs. We provide sample source codes for music-sync background animation in which each bar line (downbeat) generates a new expanding and disappearing pattern of geometric shapes (circles, triangles, or squares), the beginning of a chorus or repeated section generates several simultaneous expanding and disappearing patterns with different colors, and each chord change changes the background color of the embedding web page.
   In August 2012 Songle Widget was used by Crypton Future Media, Inc. to let visitors of their site watch a two-dimensional animated character dancing to music.
2. **Three-dimensional Animated Dancing Characters**
   In December 2012 Crypton Future Media, Inc. used Songle Widget to let visitors of their site watch three-dimensional computer-graphic characters dancing to music through WebGL rendering.
   This music-sync application also has a touchpad-like display with buttons labeled with the names of chords. These buttons light up in synchronization with the chord information from Songle Widget, and users can push any of these buttons to hear synthesized voices singing that chord.
3. **Music-sync Lighting**
   In 2013 we started using Songle Widget to link real-world physical devices—lighting devices—to music. This music-sync application supports the control of various lighting devices compatible with the DMX512 standard. It enables

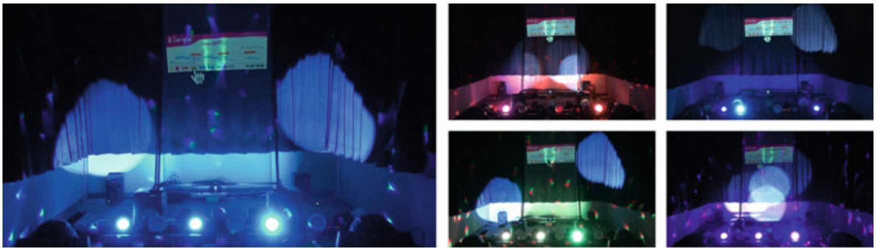


Fig. 1.9 Photographs showing stage lighting linked to music by using Songle Widget

physical lighting devices to be linked to the musical elements of any song being played on Songle Widget and to be controlled accordingly. As shown in Fig. 1.9, various types of lighting linked to music were projected in a stage-like setting.

4. **Melvie: VJ Service for Coloring Music with Videos**
   A researcher *Makoto Nakajima* used Songle Widget in collaboration with us to develop the video jockey (VJ) web service *Melvie* (http://melvie.songle.jp/) that was opened to the public in June 2014. Melvie renders different sources of short video clips without audio after mixing them and applying various special effects such as overlaying, zooming, tiling, and color change in synchronization with the music playback.
5. **V-Sido x Songle: Real-time Control of Music-sync Robot Dancing**
   A roboticist *Wataru Yoshizaki* used Songle Widget in collaboration with us to develop a music-sync robot control system, called *V-Sido x Songle*, that automatically switches several different predefined dance motions according to the music structure and the beat structure of any song registered to Songle. By using a joystick or tablet, people can change motions on the fly while the robot is dancing. Such flexible music-sync robot control had not been achieved before. In January 2015 Asratec Corp. and AIST showed that this system can make three different types of robots dance in unison as shown in Fig. 1.10.
6. **Songrium3D: Music Visualizer Featuring Three-dimensional Music-sync Animation**
   Songle Widget has been used in an advanced music visualizer, called *Songrium3D* (http://songrium.jp/map/3d/) [17], that features three-dimensional music-sync animation through WebGL rendering. In this animation, visual effects and the

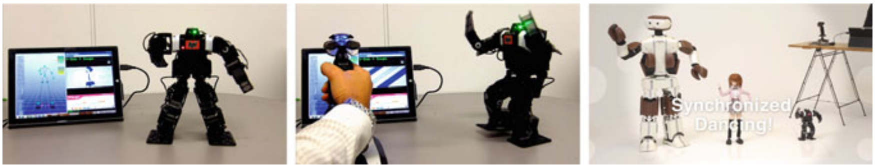


Fig. 1.10 Photographs of music-sync robot dancing controlled by V-Sido x Songle

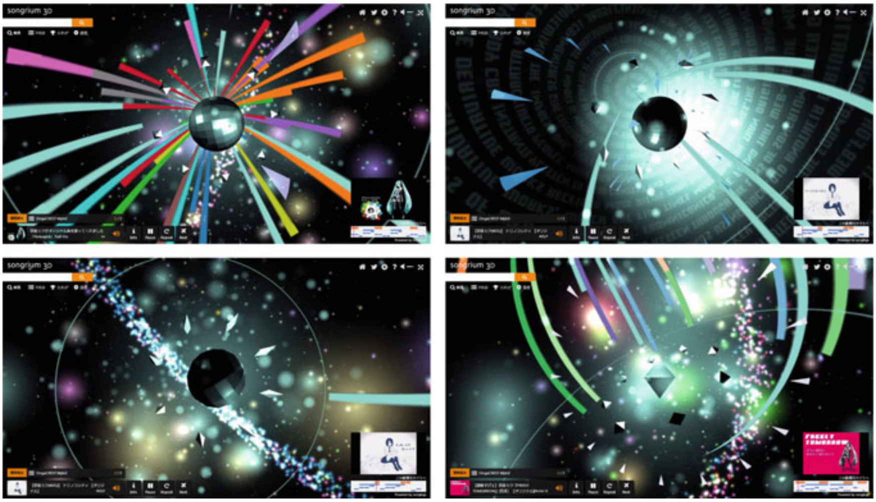


Fig. 1.11 Screenshots of Songrium3D, the music visualizer for generating three-dimensional music-sync animation in real time

motions of various objects are triggered by events in the music structure and the beat structure of a music video clip as shown in Fig. 1.11. Songrium3D is a function of Songrium web service described in Sect. 1.3.3.

7. **Photo x Songle: Music-sync Photo Slideshow**
   In December 2014 Songle Widget was used to develop a web service *Photo x Songle* (http://photo.songle.jp) that enables users to generate photo slideshows. A keyword or phrase entered by a user is used to retrieve relevant photos on the web. Those photos are then shown as animated slideshows during the playback of any song registered to Songle. A new photo usually appears at the beginning of each bar, but during chorus sections the new photos appear at every beat and with more vivid motion effects.

1.3.2.3 Implementation of Songle Widget

The implementation overview is shown in Fig. 1.12. Songle Widget was carefully designed and implemented so that it can be embedded into any web-based music-sync application without harmful side effects. We therefore chose an IFrame-based sandbox implementation that can execute our JavaScript code in a controlled environment isolated from the user application. For the sandbox implementation in JavaScript, we had to implement all necessary functions such as input sanitizer and DOM (Document Object Model) manipulation by ourselves without using external libraries such as jQuery and underscore.js.

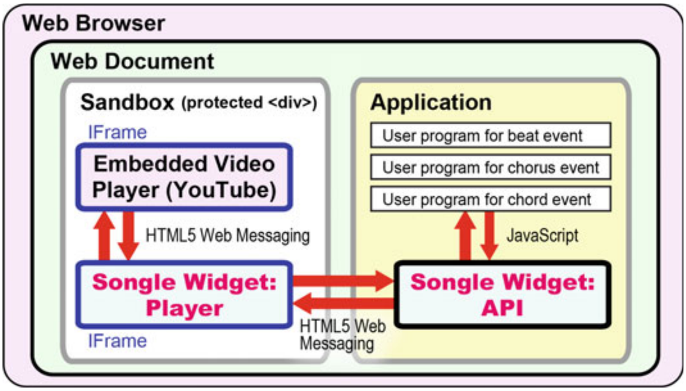

Web Browser
Web Document
Sandbox (protected <div>)
IFrame
Embedded Video Player (YouTube)
HTML5 Web Messaging
Songle Widget: Player
IFrame
Application
User program for beat event
User program for chorus event
User program for chord event
JavaScript
Songle Widget: API
HTML5 Web Messaging

Fig. 1.12 Overview of Songle Widget implementation (in the case of using beat, chorus, and chord events with the YouTube embedded video player)

As shown in Fig. 1.12, Songle Widget consists of two components: player and API. The Songle Widget player generates the user interface shown in Fig. 1.8, manages user interactions on a web browser, and provides an encapsulation wrapper of different embedded video players and music players. On the other hand, the Songle Widget API serves as the programming interface for user applications and handles all events to trigger user programs. Those user programs should be registered and bound to events (musical elements) in advance by using this API. To achieve interdomain communication over the sandbox and IFrame, we use the *HTML5 Web Messaging* mechanism.

1.3.3 Songrium

Songrium (http://songrium.jp) [15–17] is a music browsing assistance service that enables visualization and exploration of large amounts of music content with the aim of enhancing user experiences in enjoying music. The main target content of Songrium is music video clips of original songs using a singing synthesis technology called *VOCALOID* [25] and their derivative works on the most popular Japanese video-sharing service, *Niconico* (http://www.nicovideo.jp/video_top). Songrium has analyzed more than 750,000 music video clips and revealed that over 610,000 derivative works such as covers and dance arrangements have been created from more than 140,000 original songs.

Figure 1.13 shows an overview of Songrium. Songrium allows people to understand various relations of music. It was difficult for people listening to original songs to notice that there exist various derivative works related to them, such as cover versions, singing or dancing video clips, and music video clips with 3D animations. By providing people with easy, intuitive access to those derivative works, Songrium

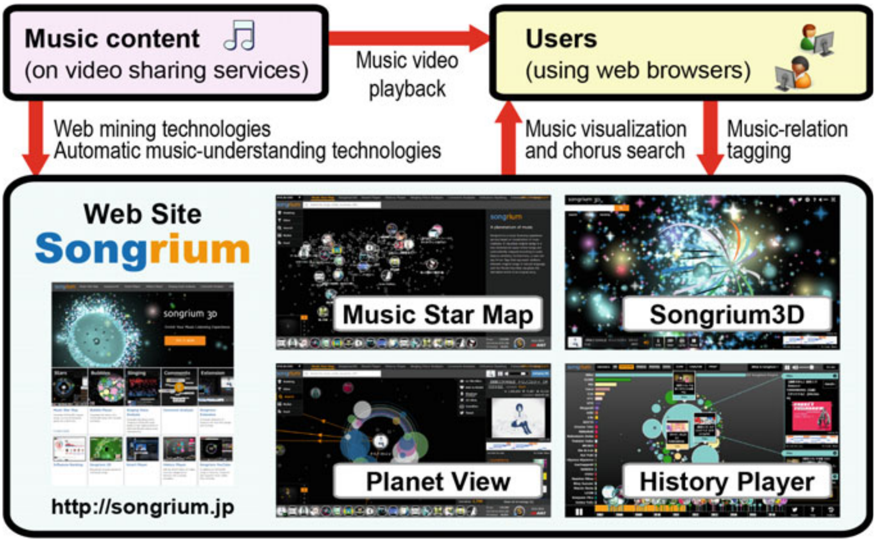


Fig. 1.13 Overview of Songrium web service (http://songrium.jp)

enables them not only to find interesting music video clips, but also to know and respect the creators of music and video clips. To visualize such relations, Songrium automatically gathers information related to original songs and their derivative works, which are expanding day-by-day. It then classifies them and estimates the relations between original songs and derivative works.

Songrium uses web mining and music-understanding technologies together with advanced data visualization techniques to achieve unique functions, such as a *Music Star Map* (Sect. 1.3.3.1), *Planet View* (Sect. 1.3.3.1), *Songrium3D* (Sect. 1.3.3.2), and *History Player* (Sect. 1.3.3.3), as shown in Fig. 1.13. These functions provide various bird's eye viewpoints about a huge amount of media content.

1.3.3.1 Music Star Map and Planet View

Music Star Map is a function that visualizes original songs. Figure 1.14a shows a screenshot of the Music Star Map function. The position of each original song is determined in a two-dimensional space of the Music Star Map so that similar songs can be closer to each other on the basis of audio-based music similarities among original songs.

When a user clicks an original song on the Music Star Map, its derivative works appear as colorful circular icons and orbit the selected original song. We designate this view as *Planet View*. Figure 1.14b shows a Planet View screenshot. Each circular icon denotes a derivative work with attributes represented by the icon orbit, size, color, and velocity. Table 1.1 shows how attributes of each derivative work are visually

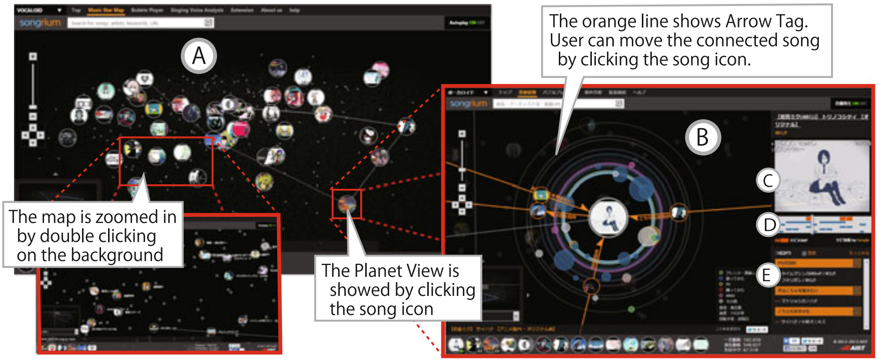


Fig. 1.14 Screenshots of the (**a**) *Music Star Map* and (**b**) *Planet View* interfaces of Songrium: the former visualizes original songs; the latter visualizes their derivative works. **a** All original songs are arranged in a two-dimensional space with similar songs positioned in proximity. Any area can be expanded for viewing by double clicking. It can then be scrolled to by dragging. **b** After selecting an original song on the Music Star Map, users can view its derivative works rotating around the selected song in the center. **c** Embedded video player of Niconico for video playback. **d** SmartMusicKIOSK interface provided by Songle Widget. **e** A list of *Arrow Tags* that represent annotated relations to and from this song instance

Table 1.1 Attributes of a derivative work represented by a circular icon on the Planet View of Songrium

Visual features	Attribues of derivative work
Radius of orbit	Publishing date
Icon size	Number of page views
Velocity	Proportion of “favorites” to page views
Color	Derivative category

represented by a circular icon. The official embedded video player of the Niconico service, shown at the upper-right corner, can play back a video clip of the selected song (Fig. 1.14c). Since this music-video playback interface is implemented by using the Songle Widget framework described in Sect. 1.3.2, Songrium also has all functions of the SmartMusicKIOSK interface, which is shown below the embedded video player (Fig. 1.14d). Songrium has an original social tagging framework called *Arrow Tag* that allows users to annotate various relations between music content. Figure 1.14e shows a list of Arrow Tags.

1.3.3.2 Songrium3D

Songrium3D is a visualization function based on the *Music Star Map* of Songrium. While the Music Star Map visualizes original songs and their derivative works in a two-dimensional space, Songrium3D visualizes them in a three-dimensional space.

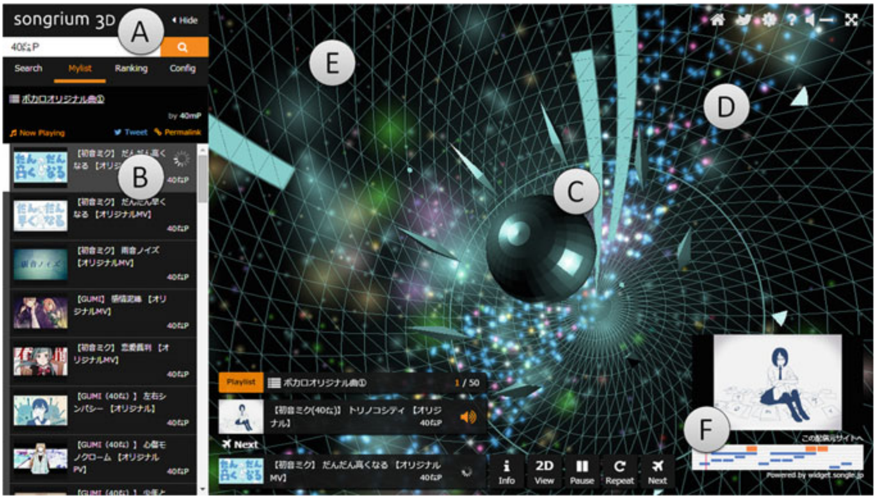


Fig. 1.15 Screenshot of the *Songrium3D* function. **a** A user can search songs using keywords. Similarly, a user can search playlists in Niconico using keywords or a URL. When the user chooses a playlist, it starts automatic playback of the playlist. **b** The lower-left panel shows a playlist. **c** The spherical object in the center represents an original song. Some objects and ribbons near the song are visual effects that are synchronized to a song. **d** A song is encompassed with many colorful particles that represent its derivative works. **e** Other original songs are apparent way out there. **f** The embedded video player of Niconico for video playback and the SmartMusicKIOSK interface provided by Songle Widget

Using three-dimensional visualization, it visualizes many original songs and derivative works together with their music structure simultaneously and seamlessly.

Figure 1.15 as well as Fig. 1.11 show screenshots of Songrium3D. The spherical object represents an original song in the center of the screen. When it plays a song, this object and peripheral objects move rhythmically in synchronization with the song. Just like the Music Star Map, the three-dimensional (x–y–z) position of a song is also determined so that similar songs can be closer to each other on the basis of audio-based music similarities among original songs. Many colorful circumjacent objects indicate derivative works of an original song. Color corresponds to the category of a derivative work in the same manner as the Planet View (Fig. 1.14b).

Three-dimensional animation of Songrium3D was used as a back-screen movie on the live stage of a virtual singer, *Hatsune Miku* [2], in the “SNOW MIKU 2015 LIVE!” concerts held four times in February 7–8, 2015. It was hosted by Crypton Future Media, Inc. for a total number of audiences of about 7,000. We generated a prerecorded animation of Songrium3D to avert problems deriving from internet connections or real-time rendering. Screenshots of Songrium3D are automatically

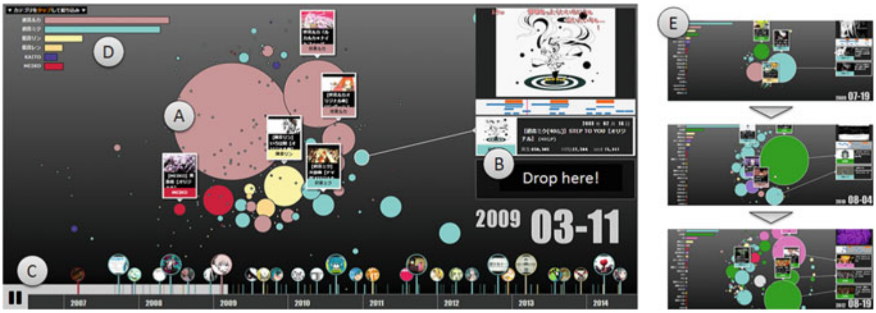


Fig. 1.16 Screenshot of the *History Player* function of Songrium. It visualizes the history of VOCALOID songs. **a** Each bubble represents a music video clip. Its size denotes the play counts; its color shows the VOCALOID character. When a user clicks a bubble, its thumbnail and metadata are shown. **b** If a user drags and drops a bubble, its song is added to the playlist. **c** The timeline at the bottom displays the current playback time and popular video clips in each period of the timeline. When a user clicks on the timeline, it jumps to the clicked period. **d** The bar chart shows the temporal popularity of VOCALOID characters (the summation of play counts of bubbles with the same color). **e** Along the playback timeline, new bubbles automatically appear in chronological order when those songs were published (uploaded)

generated and combined to produce a single movie. A movie of the live performance[1] and the animation of Songrium3D for the live[2] are available on the web.

1.3.3.3 History Player

History Player visualizes the history of VOCALOID songs on Niconico. Figure 1.16 portrays screenshots of the History Player. By automatically displaying songs in chronological order, it gives a user a full perspective on the trends and transitions of popular VOCALOID songs over time. Each song is represented as a “bubble” (a colored circle). New song bubbles appear in accordance with their respective published dates and congregate in an animation (Fig. 1.16a). The color of each bubble corresponds to the VOCALOID character, whereas the sizes of the bubbles indicate play counts on Niconico. On the left side of the screen, the bar chart presents a summation of play counts of bubbles for each VOCALOID character.

The interface exhibits growth in the music content creation community, arranged by published date, in an animated display. It automatically plays back music video clips of songs for which the play count is high in the period. Consequently, this feature enables a user to experience various popular songs in one continuous movie, providing a clear, intuitive picture of how trends on the Niconico video-sharing service change.

[1] https://youtu.be/GOano9x9cBY.

[2] https://youtu.be/71o8Jit1c4I.

A user can play back songs of interest with drag-and-drop operation if the user becomes curious about some displayed songs (Fig. 1.16b). In addition, the user can click on the timeline shown in Fig. 1.16c to jump from the current playback time to the clicked period (we called this function "time warp"), or click the bar corresponding to a VOCALOID character of interest on the left side shown in Fig. 1.16d to listen to songs of its character only. Furthermore, Songrium enables a user to easily play back the chorus section by using the SmartMusicKIOSK interface. This interactive function thus provides a bird's eye viewpoint about the history of a huge amount of media content; such historical viewpoint cannot be achieved on video sharing services in usual.

Furthermore, for users who particularly like singing or dancing derivative works, the History Player has two different versions: "Singing derivative works" version and "Dancing derivative works" version. They display specified derivative works with the same interface. The bubble colors correspond to their original songs and a bar chart shows a trend of original songs for derivative works.

1.3.3.4 Implementation of Web Mining Technologies for Songrium

Every music video clip on Songrium is automatically classified as an original song or a derivative work by using social tags of video clips and analyzing various related web sites. Derivative works can be identified when the description text of the video clip includes a hyperlink to the original video clip from which it was derived. These hyperlinks almost always exist on Niconico because users prefer to acknowledge the original video clip.

When a derivative work is incorporated, its relation to the original song is estimated automatically. The derivative works are classified into predefined categories. We defined six categories of derivative works: (a) Singing a song, (b) Dancing to a song, (c) Performing a song on musical instruments, (d) Featuring 3D characters in music video, (e) Creating a music video for a song, and (f) Others. The first three categories are derived from official categories used by Niconico; the other two categories are derived from our previous work [18, 19]. "Others" includes, for example, videos which review or rank existing videos, or which is karaoke version, or which use VOCALOID songs as the background music of other video clips. It also includes videos that are not classifiable. With the exception of category *Others*, all have their own unique social tags on Niconico. Using these tags, Songrium can produce a reliable classification of derivative works.

Moreover, Songrium provides a crowdsourcing error-correction interface that enables users to easily report an error in any of the above classification of video clips, extraction of links, or estimation of relations to improve the user experience further.

1.4 Content-Creation Support Technologies

For creation support functions of the content ecosystem web services, it is important to realize technologies that support and complement human abilities of content creation by using automatic generation of music content as basic technologies. As an example of such technologies, we developed *TextAlive* (http://textalive.jp), a lyrics animation production support service that automatically synchronizes music and lyrics to allow users to easily create music-synchronized lyrics animation. We have released this service to the general public for field-testing, and have continued to research and develop functional extensions. While this service is a content-creation support technology, it is also a content-appreciation support technology that allows users to enjoy creation results. It can also be used as a new method for discovering (recommending) music through lyrics animation. In this section I describe TextAlive in detail.

Besides TextAlive, we developed a wide range of content-creation support technologies applicable to songs. These include:

- Songmash (http://songmash.jp), a mash-up music production service that enables users to enjoy creating multi-song music mashups [3]
- AutoGuitarTab, a computer-aided composition system for rhythm and lead guitar parts in the tablature space [28]
- VocaRefiner, an interactive singing recording system with integration of multiple singing recordings [31]
- Song2Quartet: a system for generating string quartet cover songs from polyphonic audio of popular music [38]
- CrossSong Puzzle, a music game interface for generating and unscrambling music mashups with real-time interactivity [41]
- An interface that can edit the vocal F0 in existing songs by using F0 estimation and singing voice separation [21]
- A music performance assistance system based on vocal, harmonic, and percussive source separation and content visualization for music audio signals [4]

We also developed the following content-creation support technologies related to music video:

- Songroid (http://songroid.jp), a dance animation creation/appreciation support service based on an automated choreography synthesis technology using Gaussian Process leveraging dance motion examples [8, 9]
- Dancing Snap Shot, a dancing character animation generation system that converts the user's face photo input into a three-dimensional computer graphics character model having a face similar to the user
- VRMixer, a music video mixing system that enables users to visually appear in existing music videos by using a video segmentation technology [20]
- A soundtrack generation system to synchronize the climax of a video clip with music [40]

Simple creation like easy customization by amateurs as well as advanced creation by professionals are important in the “age of billions of creators.” With these achievements, users could enjoy casual content creation in a variety of forms even if expert knowledge is lacking.

1.4.1 TextAlive

TextAlive (http://textalive.jp) [22–24] is a lyrics animation production support service that enables users to animate lyrics in time to music. Due to the spread of music- and video-sharing services, many videos matched to music have been published (uploaded). Lyrics animations (kinetic typography videos) in particular can express the lyrics of music attractively, but production requires enormous effort, from installing and learning how to use video production tools to adjusting the timing of character movement.

Since TextAlive is based on music-understanding technologies and programming environment technologies, users can easily produce and share lyrics animations (Fig. 1.17). By choosing a song and specifying the video’s style, novice users without expertise can quickly produce lyrics animations with various effects. In addition, users can choose “templates” of visual effects for each phrase, word, or character with an intuitive interface to express intended effects. Furthermore, advanced users (programmers) can program templates and their parameter tuning interfaces, then share the templates with other users on TextAlive. Using TextAlive, thus, allows people with diverse background to demonstrate their creativity and enjoy lyrics animation production without an enormous effort.

TextAlive uses a song and its lyrics on the web to create, edit, and share lyrics animations. It operates by the following mechanisms (Fig. 1.18).

(1) When a user registers the URL of a song and the URL of its lyrics, TextAlive analyzes the content of the song and estimates the timing of each word and character in music by using our lyrics synchronization technology.
(2) When a user selects music registered to TextAlive, the music streamed directly from the original web site is played back, and lyrics animation is rendered in synchrony, so the music and its lyrics can be enjoyed visually and acoustically.

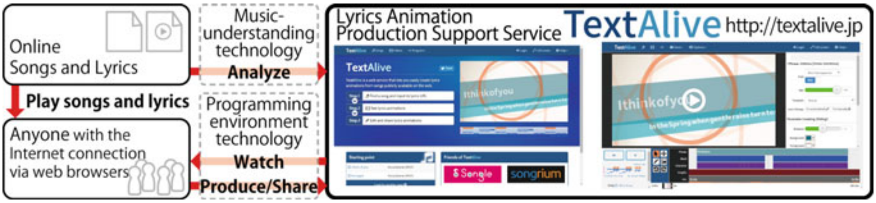

Online Songs and Lyrics
Play songs and lyrics
Anyone with the Internet connection via web browsers
Music-understanding technology
Analyze
Programming environment technology
Watch
Produce/Share
Lyrics Animation Production Support Service
TextAlive
http://textalive.jp

Fig. 1.17 Overview of TextAlive web service (http://textalive.jp)

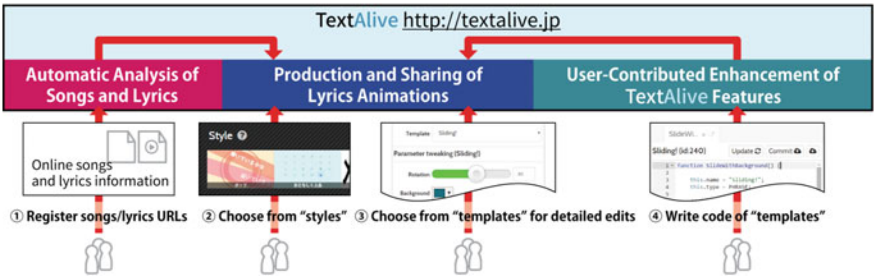

TextAlive http://textalive.jp
Automatic Analysis of Songs and Lyrics
Production and Sharing of Lyrics Animations
User-Contributed Enhancement of TextAlive Features
Online songs and lyrics information
Style
① Register songs/lyrics URLs
② Choose from "styles"
③ Choose from "templates" for detailed edits
④ Write code of "templates"

Fig. 1.18 TextAlive enables users to produce and share lyrics animation easily

Users can choose visual arrangements for the entire video from various "styles" and change instantly, easily producing lyrics animations which they prefer.

(3) Users can choose detailed visual effects from "templates" for each phrase, word, or character, arrange them as they like, and share lyrics animations online. Other users can add further edits to produce derivative works of shared lyrics animations.

(4) Users who have programming skills can edit and produce new "templates" and "parameter tuning interfaces for templates." Because these are shared on TextAlive, this enhances the expressivity of all users.

1.4.1.1 Three Features of TextAlive

TextAlive has the following three features:

1. Enables easy production of lyrics animations based on online music and lyrics information
   Users can view lyrics animations that change in time to music by entering the URL of a song (audio file in MP3 format or music content on a music- and video-sharing service) and the URL of its lyrics (text file) into TextAlive, or by selecting a pair of a song and its lyrics which has already been registered. The lyrics animation is synthesized automatically on the user's web browser, based on lyric utterance timing information (Fig. 1.19).
   Users can choose a style on the web browser to instantly change effects of the entire video, and produce lyrics animations to their preference. They can also choose effects from templates and adjust them for each phrase, word, and character, to perform even more detailed effects. This is achieved with programming environment technologies, which can execute and update programs that synthesize lyrics animations from lyric utterance timing (styles), and programs that control the shape changes and movement of characters (templates) on the web browser.
   Thus, even users who have never produced a lyrics animation can enjoy producing music-synchronized lyrics animations with effects of their preference.

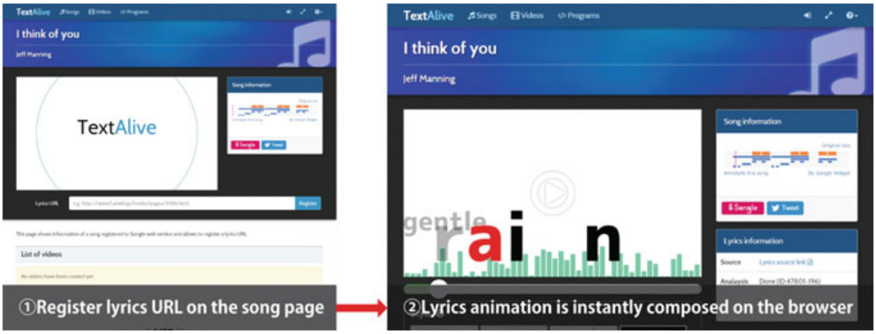
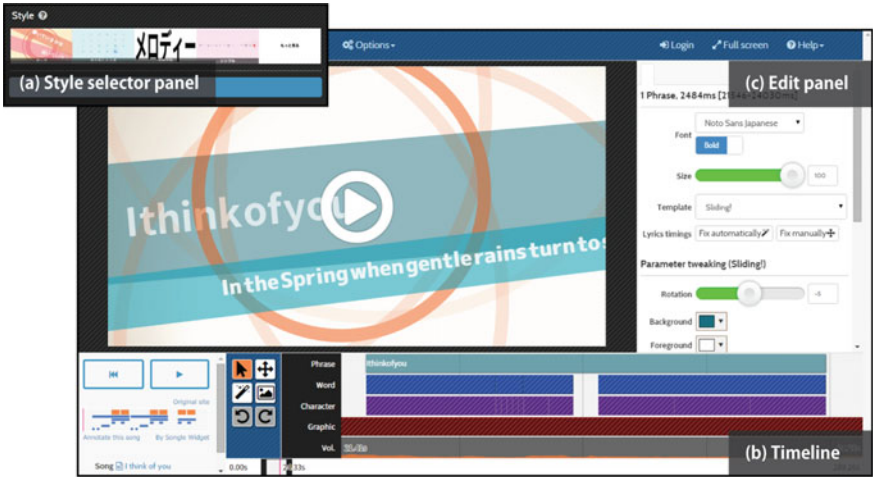


Fig. 1.19 Procedure to create lyrics animations on TextAlive



1 Phrase, 2484ms

Font: Noto Sans Japanese, Bold

Size: 100

Template: Sliding!

Lyrics timings: Fix automatically | Fix manually

Parameter tweaking (Sliding!)

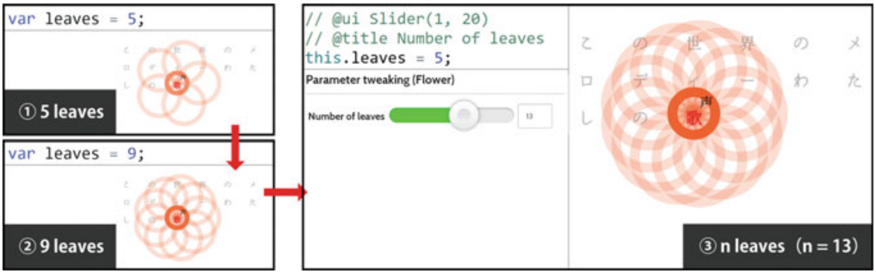
Rotation: -6

Background

Foreground

Fig. 1.20 Screenshot of TextAlive's editing interface. Because lyrics animations are saved in the TextAlive web site, edited animations and programs can be shared with other users. Users not only can view lyrics animations, but can also enjoy producing derivative works to their preference

2. Supports people's creativity with intuitive interfaces and mechanisms that simplify production of derivative work
   TextAlive provides three editing interfaces with which users can easily produce lyrics animations. Because users can edit using lyrics animations produced automatically based on the chosen style, manual effort is greatly reduced even for advanced users with experience in production (Fig. 1.20).
   (a) Style selector panel: Using the simple interface for choosing styles, users can change effects for the entire lyrics animation instantly, and produce lyrics animations to their preference.
   (b) Timeline: Users can easily search for lyrics in any part they want to see by moving the timeline cursor left and right with the mouse. In addition, automati-



```
var leaves = 5;
```

```
var leaves = 9;
```

```
// @ui Slider(1, 20)
// @title Number of leaves
this.leaves = 5;
```

Fig. 1.21 Example of using the programming panel on TextAlive. Changing the default value of the leaf number of the template program from 5 to 9 changes the number of red circles displayed. The final screen shown has a slider for adjusting the number of leaves added to the editing panel. Thus, by programmers preparing interfaces, even users who do not program can set the desired number of leaves, and reflect their intended effect

cally analyzed lyric utterance timing is displayed in colored sections by phrase, word, and character, so that by stretching sections and moving sections left and right, users can edit lyric utterance timing for any group of characters or single character, and correct errors in automatic synchronization results. This is yet another crowdsourcing error-correction interface.

(c) Edit panel: This panel is displayed when the user clicks the edit button while viewing a lyrics animation. Users can choose the font, size, motion and transformation of phrases, words, and characters chosen on the timeline from a lot of templates. Visual effects of templates can be easily customized with the intuitive parameter tuning interface, such as sliders.

3. Enables enhancement of effects and editing features by web-based programming
   In TextAlive, when users are not satisfied with the templates prepared in advance, they can program their own new templates. The service realizes a new type of live programming, in which users can revise algorithms that determine the movement of characters while still playing a lyrics animation, without closing the editing interface. Since each edited program is immediately executed to update the lyrics animation, it is easier for novice users, who are new to the study of programming, to understand the program.
   The parameter tuning interface on the edit panel can be easily extended by adding text comments to a template program with a simple notation method. With the added interfaces, users can customize the visual effect of a template, (Fig. 1.21). Furthermore, TextAlive provides an web API that allows users to display lyrics animations on their own web pages.

1.4.1.2 Implementation of TextAlive

TextAlive generates lyrics animation in real time without using video clips that are rendered in advance: programs such as styles and templates written in JavaScript are safely executed on the user's web browser to display lyrics animations. On the other hand, all video-sharing services just distribute video clips that are displayed by rapidly changing multiple still images prepared in advance. Because of this, it was difficult to change the content of video clips after production. Since TextAlive does not rely on such prepared video clips, it is easy for TextAlive users to apply effects for existing videos to videos of other music, or edit just part of a video. This advantage could underpin a culture in which various users can create many derivative works from a single original work.

Each user-generated lyric animation on TextAlive consists of the following information: 1) URLs to the song and lyrics, 2) the structure of the lyrics, which is a tree of the text units (phrases, words, and characters), and 3) a list of assigned templates and their parameter values for each text unit. These set of information is represented by a JSON object and is version-controlled on the TextAlive web site. In fact, not only the created animation but also the estimated timing information, correction history, and template definitions are all version-controlled, enabling to fork derivative work and to rollback flawed edits.

Just like Songrium, TextAlive is also implemented by using Songle Widget that streams music directly from the original web site. Furthermore, all music-understanding technologies including the lyrics synchronization technology are internally executed on Songle, and TextAlive retrieves vocalized timing of lyric text and other music-analysis information from Songle.

1.5 Musical Similarity and Typicality Estimation Technologies

Musical similarity between songs refers to the degree to which a song is similar to a baseline song. We proposed to estimate the similarity by calculating the generative probability of musical elements in a song from a probabilistic generative model of the baseline song. Meanwhile, musical typicality refers to the degree to which an individual song is typical (common) in a baseline set of songs. We therefore proposed to estimate the typicality by calculating the generative probability of musical elements of an individual song from a probabilistic generative model of a set of songs. The probabilistic generative model here refers to a model that can calculate the probability (generative probability) of each musical element (musical feature, lyrics, chord progression, etc.) in terms of how likely it is for the element to appear. A key advantage of this approach is that a common unified framework based on probabilistic generative models can be used for calculating both similarity and typicality [34, 36, 37].

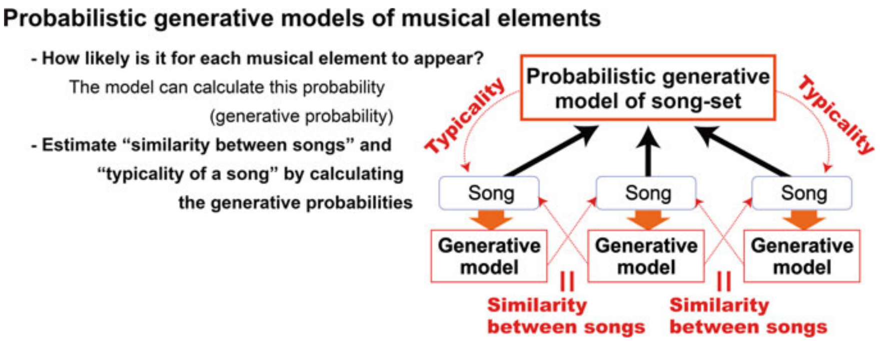

Probabilistic generative models of musical elements
- How likely is it for each musical element to appear?
The model can calculate this probability
(generative probability)
- Estimate "similarity between songs" and
"typicality of a song" by calculating
the generative probabilities
Probabilistic generative model of song-set
Typicality
Typicality
Song
Song
Song
Generative model
Generative model
Generative model
||
||
Similarity between songs
Similarity between songs

Fig. 1.22 Estimation of musical similarity and typicality based on probabilistic generative models

1.5.1 Musical Similarity and Typicality Estimation Based on Probabilistic Generative Models

Since music contains multifaceted aspects, such different aspects should be taken into account when calculating musical similarity and typicality. We therefore developed a framework that can calculate the similarity and typicality for various aspects (musical elements) of music [34, 36, 37]. Our framework uses probabilistic generative models of five musical elements (vocal timbre, musical timbre, rhythm, lyrics, and chord progression) to estimate "similarity between songs" and "typicality of a song" by calculating the generative probabilities from these models. The five musical elements are estimated from music audio signals containing sounds of singing voices and various instruments.

As shown in Fig. 1.22, for each aspect of music (each musical element), estimation of the "similarity" between songs was made possible by building a probabilistic generative model of each song and then calculating the generative probability of the target musical element within a different song. Furthermore, the "typicality" of each song in a set of songs was estimated by building a probabilistic generative model of the entire set of songs and then calculating the generative probability of the target musical element within an individual song.

Of the five musical elements, the audio features representing vocal timbre, musical timbre, and rhythm were estimated for every frame (time-series unit). They were then discretized with vector quantization. Lyrics were broken down into symbols of Japanese morphemes and English words. Those discrete symbols of vocal timbre, musical timbre, rhythm, and lyrics were modeled using Latent Dirichlet Allocation (LDA) as shown in Fig. 1.23. With LDA analysis, the topic distribution in each song (distribution of topic mixture ratio) and symbol distribution in each topic (distribution of unigram probability of symbols) were obtained. These distributions can be estimated as posterior distributions by Bayesian estimation assuming Dirichlet distributions. The parameter of the posterior Dirichlet distribution can be interpreted as the number of observations of the corresponding topic or symbol. Using these

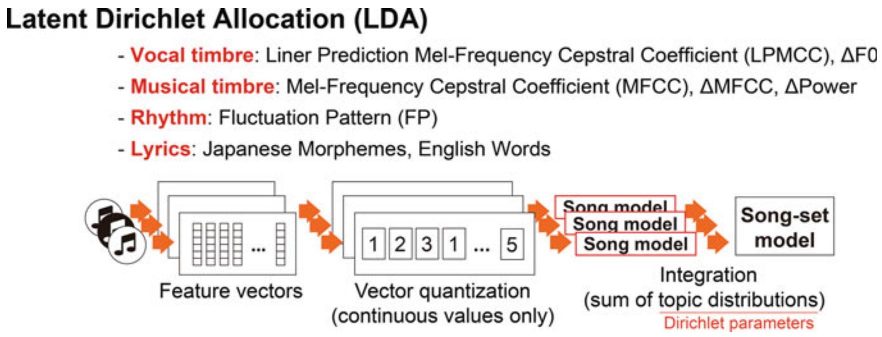
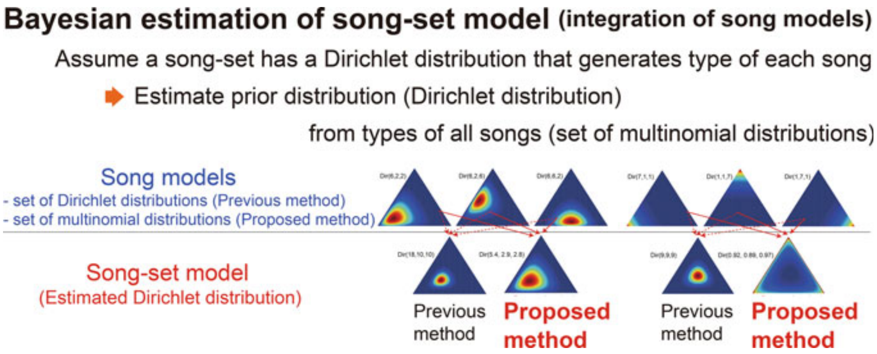

Latent Dirichlet Allocation (LDA)
- Vocal timbre: Liner Prediction Mel-Frequency Cepstral Coefficient (LPMCC), ΔF0
- Musical timbre: Mel-Frequency Cepstral Coefficient (MFCC), ΔMFCC, ΔPower
- Rhythm: Fluctuation Pattern (FP)
- Lyrics: Japanese Morphemes, English Words
...
1 2 3 1 ... 5
Song model
Song model
Song model
Song-set model
Feature vectors
Vector quantization (continuous values only)
Integration (sum of topic distributions)
Dirichlet parameters

Fig. 1.23 Song modeling and song-set modeling with Latent Dirichlet Allocation (LDA)

parameters, the topic distribution of the song-set was estimated from the topic distributions of the songs in the song-set. Similarity and typicality were estimated by calculating the generative probability of a song from the model of each song and the model of the song-set obtained in this way.

Meanwhile, as for the chord progression, we used the chord estimation results from Songle described in Sect. 1.3.1, and modeled their chord progression by using a variable-order Markov process (up to a theoretically infinite order) called the variable-order Pitman-Yor language model (VPYLM) [29, 44]. By infinitizing n (length of chord progression) in the n-gram model, posterior distribution of the appropriate n for each chord could be estimated. The probabilistic generative model of the song-set was trained from chord progressions of all songs by using VPYLM. However, because the number of chords in each song is small, suitable training for the probabilistic generative model of each song could not be carried out with just a Bayesian model. We therefore dealt with this problem by first using maximum likelihood estimation to obtain a trigram (n = 3) model for each song, and then integrating it with VPYLM trained from the song-set. The perplexity of each song was calculated from the model trained on each song and the model trained on the song-set. The average generative probability of each chord in the song was calculated as the inverse of the perplexity to estimate similarity and typicality [36, 37].

However, when scrutinizing the above estimation results, we discovered the problem that the probability of unintended sequences became high with our initially-proposed generative probability (likelihood)-based calculation methods. For example, suppose for the sake of simplicity a music model that generates 0 with the probability of 60% and 1 with the probability of 40%. Under undesirable conditions in which 0 is 100% (e.g., "0 0 0 0 0" in a song), the problem of the generative probability becoming maximized occurs, regardless of the original importance of each proportion in the song. To resolve this issue, we introduced *type theory* from information theory [1], and defined the generative probability as the probability of a sequence sharing the *type* (unigram or multinomial distribution). We then calculated this probability by an exponential function of Kullback-Leibler divergence of the multinomial distribution that expresses the type of a song from the multinomial


Bayesian estimation of song-set model (integration of song models)

Assume a song-set has a Dirichlet distribution that generates type of each song

➡ Estimate prior distribution (Dirichlet distribution)

from types of all songs (set of multinomial distributions)

Song models

- set of Dirichlet distributions (Previous method)
- set of multinomial distributions (Proposed method)

Song-set model

(Estimated Dirichlet distribution)

Fig. 1.24 Bayesian estimation of song-set model from song models

distribution generated from the generative model (Latent Dirichlet Allocation). In this way, using the above example the generative probability under conditions that include probabilities 60% for 0 and 40% for 1 (e.g., “0 1 0 0 1” in a song) could be maximized [34].

Furthermore, with our initially proposed method, the song-set model is determined by summing the Dirichlet distribution parameters of individual LDA song models. However, as shown in Fig. 1.24, the problem that these individual song models (set of Dirichlet distributions) are not appropriately reflected in the song-set model arises. To resolve this problem, each song is expressed with multinomial distribution (topic distribution). By assuming that multinomial distributions of the songs in the song-set are generated from the Dirichlet distribution of the song-set, we used Bayesian estimation to estimate the generative model of the song-set [34].

The above methods are not limited to audio or music. We have already estimated the typicality of text data and developed an interface that uses its results. Specifically, to estimate the typicality of artists, we obtained categories on the artists’ Wikipedia pages and important words (words found on the Wikipedia pages), and created an LDA-based artist-set model. Because an artist’s history is written in Wikipedia, text-based relations that are different from audio-based relations obtained from musical features of songs could be found [33].

1.5.2 Evaluation of Typicality Estimation Results

We investigated how the results of estimating typicality should be evaluated and proposed original evaluation methods. In general, evaluation of similarity between songs is based on correlation with human judgment. However, because typicality is defined by the relations among songs in a huge song-set, which are difficult for human beings to grasp in the first place, typicality cannot be evaluated on the basis of human judgment criteria as in the case of similarity. Thus we proposed evaluation based on *central tendency* (the number of similar songs), one of the definitions of typicality in cognitive psychology. Specifically, we assume that typicality can be appropriately estimated if the correlation with "the number of highly similar songs" is high [36, 37].

However, it turned out that using only the above method was insufficient for showing the effectiveness of the methods for estimating typicality. This is because the absolute value of similarity and the estimation accuracy differ with different methods, so standards of "high similarity" also differ. We therefore tested the effectiveness of the proposed methods of calculating typicality by using the correlation between the ratio of songs making up the song-set and the estimated typicality, using song elements whose characteristics could be known in advance. For example, we focused on vocal timbre, namely the gender of the vocalist (male or female), and estimated the typicality provided by several of our proposed methods while changing the ratio of male and female vocals. This made it possible to conduct evaluation testing by comparing the correlations. With these steps, our contributions of calculating similarity and typicality and settling on overall criteria applicable to research on typicality in general are not limited to the domain of music, but useful for any domain in the future [34].

1.6 Discussion

In the future, everything can be digitally traceable and computable. This chapter has described various research achievements under this future vision. In the past, it was difficult to assume a huge amount of media content for research and provide various bird's eye viewpoints. Our project has explored the frontiers of dealing with such a huge amount of music content on the web and developed Songle for analyzing it, Songrium for providing such bird's eye viewpoints, and TextAlive for creating new values based on the accumulation of such content. In Sect. 1.6.1 below I discuss "*web-native music*" that is a word coined by us to emphasize such a future vision. Next, in Sect. 1.6.2, I introduce that Songrium started providing a new type of influence ranking verifiable by third parties. Then, in Sect. 1.6.3, I summarize how the above mentioned web services we developed form the entire content ecosystem web services.

1.6.1 Web-Native Music

By advancing research on music content available on the web, we investigated what is different about music on the web, which will become mainstream in the future, compared with previous music. We then realized that while the integration of music and the web gradually began with web-streaming of music previously distributed as packaged media, what is being born is "music content native to the web." This type of music could not have arisen without the web.

We call music that fulfills the following three conditions "*web-native music*" [16]:

(1) Original music people naturally think as being first released on the web
(2) Music whose derivative works can be created and released on the web without hesitation
(3) Music whose creators can publicly receive feedback that promotes further creation

With web-native music it is possible to identify works, release dates, and reactions of viewers and listeners (reviews, comments, number of playbacks, number of users who bookmarked, etc.) because it is released and played on the web. Web-native music also makes it possible to track original songs and their derivative works. As a result, phenomena that cannot previously be observed can be observed. Because everyone can refer to the original song by its permalink URL, every derivative work can reliably refer to the URL of the original song, and every introduction and acknowledgment can refer to the URL of the original song, attention to (the URLs of) original songs on the web encourages a chain of creation. Such music does not exist in the past.

An example of web-native music is VOCALOID music that uses the VOCALOID singing synthesis technology and is posted on the Niconico video-sharing service. By taking advantage of web-native music that can be digitally traceable and computable, we were able to develop a series of music web services. Such music is continuing to grow steadily. For example, the musical genre of Electronic Dance Music (EDM), which has surged in popularity in recent years in Europe and North America, has qualities similar to web-native music. We therefore started adding EDM music on SoundCloud to our services of Songle and Songrium.

In this way, web-native music will further spread in a variety of forms. Furthermore, I believe that every media content will eventually become *web-native content* in the future.

1.6.2 Influence Ranking Verifiable by Third Parties

The number of derivative works shows the power of influence of original songs. There is great value to songs from which many derivative works are created—in short, songs that are liked so much that many people create and release derivative works. Songrium made it possible to visualize that different songs could have different trends of derivation. For example, there are songs that people want to dance to and songs that they want to sing. When we first began our research of Songrium, this could not be realized at all. With Songrium, we are able to analyze and realize qualities and quantities about the derivation of content for the first time.

We therefore developed and released a new function of Songrium "*Influence Ranking*" to the general public as the world's first ranking based on the act of creating and releasing derivative works by users. This ranking represents a song's power to promote content generation (we call it "*content-generating power*"). With this ranking, songs for which many derivative works are created can be discovered for each type of derivation.

To improve the reliability of rankings with this function, we discovered that it was important to provide a list of Niconico URLs of derivative works on the web so third-party verification becomes possible. Actually, in widely used popularity rankings based on the number of playbacks, acts of individual playbacks do not have transparency due to the lack of third-party verification. Therefore, one must trust the ranking aggregator. If for some reason an error or improper operation occurs, there is no way of knowing. In contrast, the Influence Ranking system we developed has the unprecedented feature of enabling third parties to verify an original work's popularity in the form of individual derivative works on the web. So it is a highly transparent third-party verifiable ranking system. We believe that this philosophy will become essential and critical for media content of the future.

1.6.3 Content Ecosystem Web Services

Even if automatic analysis based on similarity calculation and music-understanding technologies becomes possible, and content-appreciation support technologies and content-creation support technologies are realized, they are insufficient as a similarity-aware information environment for a content-symbiotic society if they are not in conditions to be used by people. We therefore made it possible for people to directly use "content ecosystem web services" such as Songle, Songrium, and TextAlive as an information environment that accumulates automatic analysis results of music content and allows people to use those results for appreciation and creation.

In terms of an information environment, Songle allows people to access music-understanding results for more than one million pieces of music content, and similarity between content (similarity of vocal timbre and musical timbre). By leveraging Songle, Songle Widget serves as an information environment that promotes collaboration with external web services and applications, and contributes to delivering a music-synchronized world to people. Next, Songrium allows people to see user annotations on song relationships (Arrow Tags), relationships between original songs and derivative works, audio-based similarity between songs, the history of media content (History Player), and content-generating power (Influence Ranking). Those functions make possible user-led search, recommendation, and browsing and help people find their favorite content. Furthermore, TextAlive makes possible the content circulation that fosters new content from past content, and appropriate reference (citation) that respects past content.

When this OngaCREST Project initially began, we could not assume that use of these services could grow to their present state and synergistic effects between services and use cases could be obtained. However, already a variety of collaborations and applications have taken place with this series of web services. In particular, use of Songle has begun inside and outside the lab as a common foundational platform. Songle's music-understanding technologies are being used directly or indirectly by several content-appreciation support technologies and content-creation support technologies. In this way, provision of "content ecosystem web services" to end users and developers has already begun and they are being used.

1.7 Conclusion

This chapter introduces research achievements of the OngaCREST Project and describes content-appreciation support technologies that provide interfaces and services for actively enjoying music, content-creation support technologies that contribute to an "age of billions of creators," and musical similarity and typicality estimation that is important to build a similarity-aware information environment for the content-symbiotic society. In the digital content society, future content would be apt to be overwhelmed by a huge amount of ever-increasing past content but without being forgotten. The OngaCREST Project took up the challenge of creating the content-symbiotic society capable of rich, sustainable development in a "cannot-be-forgotten society" brought about through digitization. I hope that our research achievements could contribute to enable people to feel the symbiosis between past and future content and to create a society in which people can enjoy a huge amount of content through a symbiosis between people and content.

In the future, *uncopyable experiences* will become more important. A content industry has developed by advancing methods of copying experiences—-e.g., by reproducing appreciation of music performances with music CDs and DVDs. As the distribution cost of digital content will further approach zero, there is concern that the industrial value from copyable, passive experiences will be gradually lost. On the other hand, active experiences unique to each person have value because they are uncopyable experiences. For example, the activity of creating something is an uncopyable experience. Even if one looks at a creative work and imagines its process of creation, one cannot have the same experience and emotions as the work's creator.

Our technologies described in this chapter can thus contribute to create such uncopyable experiences. Even if we seek to create an "age of billions of creators," it is not easy to enable anyone to easily create high-quality music content from scratch by using content-creation support technologies. However, from the standpoint of creating "uncopyable experiences," no matter how simple a customization may be, this is an active experience that differs for each person. It has the possibility of providing sufficient value if it is uncopyable. Furthermore, content-appreciation support technologies for active music listening are also critical for creating uncopyable, active experiences. The experiences of deepening understanding through visualization during playback, customizing content through casual modification, and encountering content through interactive music search and browsing are the first steps toward "uncopyable experiences."

Although the digitization of content has progressed, the latent true value of the digital content has not yet been fully extracted. In the past, changes in content appreciation have centered on *quantitative changes* (changes in the number of content) in the huge amount of content that can be accessed passively. The next stage of changes is *qualitative changes* (changes in the quality of experiences) that bring about active "uncopyable experiences." I believe that this is the essence of digitization. We therefore seek to continue to contribute academically, industrially, socially, and culturally by researching and developing diverse content-appreciation and content-creation support technologies of achieving uncopyable, active experiences.

Acknowledgements This research was supported in part by CREST, JST. I would like to thank all those involved in the OngaCREST Project, especially co-researchers Shigeo Morishima, Satoshi Nakamura, and Kazuyoshi Yoshii, and the following who devoted their energies to developing the web services: Masahiro Hamasaki, Tomoyasu Nakano, Satoru Fukayama, Jun Kato, Kosetsu Tsukuda, Yuta Kawasaki, Keisuke Ishida, and Takahiro Inoue. I also would like to extend my appreciation to the Research Supervisor, Professor Toyoaki Nishida and Late Professor Yoh'ichi Tohkura, and Advisors of the research area of "*Creation of Human-Harmonized information Technology for Convivial Society*" of CREST, JST.

References

1. M.T. Cover, J.A. Thomas, *Elements of Information Theory* (Wiley, 2006)
2. Crypton Future Media. What is the HATSUNE MIKU movement? http://www.crypton.co.jp/download/pdf/info_miku_e.pdf, 2008
3. M.E.P. Davies, P. Hamel, K. Yoshii, M. Goto, AutoMashUpper: Automatic creation of multi-song music mashups. IEEE/ACM Trans. Audio Speech Lang. Process. **22**(12), 1726–1737 (2014)
4. A. Dobashi, Y. Ikemiya, K. Itoyama, K. Yoshii, A music performance assistance system based on vocal, harmonic, and percussive source separation and content visualization for music audio signals, in *Proceedings of SMC*, pp. 99–104, 2015
5. H. Fujihara, M. Goto, T. Kitahara, H.G. Okuno, A modeling of singing voice robust to accompaniment sounds and its application to singer identification and vocal-timbre-similarity-based music information retrieval. IEEE Trans. Audio Speech Lang. Process. **18**(3), 638–648 (2010)
6. H. Fujihara, M. Goto, J. Ogata, K. Komatani, T. Ogata, H.G. Okuno, Automatic synchronization between lyrics and music CD recordings based on Viterbi alignment of segregated vocal signals, in *Proceedings of ISM*, pp. 257–264, 2006
7. H. Fujihara, T. Kitahara, M. Goto, K. Komatani, T. Ogata, H.G. Okuno, F0 estimation method for singing voice in polyphonic audio signal based on statistical vocal model and Viterbi search, in *Proceedings of ICASSP 2006*, pp. V–253–256, 2006
8. S. Fukayama, M. Goto, Automated choreography synthesis using a gaussian process leveraging consumer-generated dance motions, in *Proceedings of ACE, 2014*
9. S. Fukayama, M. Goto, Music content driven automated choreography with beat-wise motion connectivity constraints, in *Proceedings of SMC*, pp. 177–183, 2015
10. M. Goto, A real-time music scene description system: predominant-F0 estimation for detecting melody and bass lines in real-world audio signals. Speech Commun. **43**(4), 311–329 (2004)
11. M. Goto, A chorus-section detection method for musical audio signals and its application to a music listening station. IEEE Trans. Audio Speech Lang. Process. **14**(5), 1783–1794 (2006)
12. M. Goto, Active music listening interfaces based on signal processing, in *Proceedings of ICASSP*, 2007

13. M. Goto, K. Yoshii, H. Fujihara, M. Mauch, T. Nakano, Songle: A web service for active music listening improved by user contributions, in *Proceedings of ISMIR*, pp. 311–316, 2011
14. M. Goto, K. Yoshii, T. Nakano, S. Widget, Making animation and physical devices synchronized with music videos on the web, in *Proceedings of IEEE ISM*, pp. 85–88, 2015
15. M. Hamasaki, M. Goto, Songrium: A music browsing assistance service based on visualization of massive open collaboration within music content creation community, in *Proc. of the 9th International Symposium on Open Collaboration (WikiSym + OpenSym 2013)*, pp. 1–10, 2013
16. M. Hamasaki, M. Goto, T. Nakano, Songrium: A music browsing assistance service with interactive visualization and exploration of a web of music, in *Proceedings of the 23rd International World Wide Web Conference (WWW 2014)*, pp. 523–528, 2014
17. M. Hamasaki, M. Goto, T. Nakano, Songrium: browsing and listening environment for music content creation community, in *Proceedings of SMC*, pp. 23–30, 2015
18. M. Hamasaki, H. Takeda, T. Hope, T. Nishimura, Network analysis of an emergent massively collaborative creation community: How can people create videos collaboratively without collaboration?, in *Proceedings of ICWSM*, pp. 222–225, 2009
19. M. Hamasaki, H. Takeda, T. Nishimura, Network analysis of massively collaborative creation of multimedia contents—case study of Hatsune Miku videos on Nico Nico Douga, in *Proceedings of uxTV*, pp. 165–168, 2008
20. T. Hirai, S. Nakamura, T. Yumura, S. Morishima, VRMixer: Mixing video and real world with video segmentation, in Proceedings of ACE, 2014
21. Y. Ikemiya, K. Yoshii, K. Itoyama, Singing voice analysis and editing based on mutually dependent f0 estimation and source separation, in *Proceedings of IEEE ICASSP*, pp. 574–578, 2015
22. J. Kato, T. Igarashi, M. Goto, Programming with examples to develop data-intensive user interfaces. IEEE Comput. **49**(7), 34–42 (2016)
23. J. Kato, T. Nakano, M. Goto, TextAlive: Integrated design environment for kinetic typography, in *Proceedings of ACM CHI*, pp. 3403–3412, 2015
24. J. Kato, T. Nakano, M. Goto, TextAlive Online: Live programming of kinetic typography videos with online music, in *Proceedings of ICLC*, pp. 199–205, 2015
25. H. Kenmochi, H. Ohshita, Vocaloid—commercial singing synthesizer based on sample concatenation, in *Proceedings of Interspeech*, pp. 4010–4011, 2007
26. M. Mauch, S. Dixon, Approximate note transcription for the improved identification of difficult chords, in *Proceedings of ISMIR*, pp. 135–140, 2010
27. M. Mauch, S. Dixon, Simultaneous estimation of chords and musical context from audio. IEEE Trans. ASLP **18**(6), 1280–1289 (2010)
28. M. McVicar, S. Fukayama, M. Goto, AutoGuitarTab: computer-aided composition of rhythm and lead guitar parts in the tablature space. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(7), 1105–1117 (2015)
29. D. Mochihashi, E. Sumita, The infinite Markov model, in *Proceedings of Advances in Neural Information Processing Systems 20 (NIPS 2007)*, pp. 1017–1024, 2007
30. T. Nakamura, H. Kameoka, K. Yoshii, M. Goto, Timbre replacement of harmonic and drum components for music audio signals, in *Proceedings of IEEE ICASSP*, pp. 7520–7524, 2014
31. T. Nakano, M. Goto, VocaRefiner: an interactive singing recording system with integration of multiple singing recordings, in *Proceedings of SMC*, pp. 115–122, 2013
32. T. Nakano, M. Goto, LyricListPlayer: a consecutive-query-by-playback interface for retrieving similar word sequences from different song lyrics, in *Proceedings of SMC*, pp. 344–349, 2016
33. T. Nakano, J. Kato, M. Hamasaki, M. Goto, PlaylistPlayer: An interface using multiple criteria to change the playback order of a music playlist, in *Proceedings of ACM IUI*, pp. 186–190, 2016
34. T. Nakano, D. Mochihashi, K. Yoshii, M. Goto, Musical typicality: how many similar songs exist?, in *Proceedings of ISMIR*, pp. 695–701, 2016
35. T. Nakano, K. Yoshii, M. Goto, Vocal timbre analysis using latent dirichlet allocation and cross-gender vocal timbre similarity, in *Proceedings of IEEE ICASSP*, pp. 5239–5343, 2014

36. T. Nakano, K. Yoshii, M. Goto, Musical similarity and commonness estimation based on probabilistic generative models, in *Proceedings of IEEE ISM*, pp. 197–204, 2015
37. T. Nakano, K. Yoshii, M. Goto, Musical similarity and commonness estimation based on probabilistic generative models of musical elements. Int. J. Semant. Comput. (IJSC) **10**(1), 27–52 (2016)
38. G. Percival, S. Fukayama, M. Goto, Song2Quartet: a system for generating string quartet cover songs from polyphonic audio of popular music, in *Proceedings of ISMIR*, pp. 114–120, 2015
39. S. Sasaki, K. Yoshii, T. Nakano, M. Goto, S. Morisihima, LyricsRadar: a lyrics retrieval system based on latent topics of lyrics, *Proceedings of ISMIR*, pp. 585–590, 2014
40. H. Sato, T. Hirai, T. Nakano, M. Goto, S. Morishima, A soundtrack generation system to synchronize the climax of a video clip with music, in *Proceedings of IEEE ICME*, 2016
41. J.B.L. Smith, G. Percival, J. Kato, M. Goto, S. Fukayama, Generating and unscrambling music mashups with real-time interactivity. CrossSong Puzzle, in *Proceedings of SMC*, pp. 61–67, 2015
42. K. Tsukuda, M. Goto, ExploratoryVideoSearch: a music video search system based on coordinate terms and diversification, in *Proceedings of IEEE ISM*, pp. 221–224, 2015
43. K. Tsukuda, M. Hamasaki, M. Goto, SmartVideoRanking: video search by mining emotions from time-synchronized comments, in *Proceedings of IEEE ICDMW*, 2016
44. K. Yoshii, M. Goto, A vocabulary-free infinity-gram model for nonparametric bayesian chord progression analysis, in *Proceedings of ISMIR*, pp. 645–650, 2011