Songle Widget: Making Animation and Physical Devices Synchronized with Music Videos on the Web

Masataka Goto
National Institute of Advanced Industrial Science and Technology (AIST), Japan
m.goto@aist.go.jp

Kazuyoshi Yoshii
Kyoto University, Japan
yoshii@kuis.kyoto-u.ac.jp

Tomoyasu Nakano
National Institute of Advanced Industrial Science and Technology (AIST), Japan
t.nakano@aist.go.jp

***Abstract*—This paper describes a web-based multimedia development framework, *Songle Widget*, 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. To avoid the difficulty of time-consuming manual annotation, Songle Widget makes it easy to develop web-based applications with rigid music synchronization by leveraging music-understanding technologies. Four types of musical elements (music structure, hierarchical beat structure, melody line, and chords) have been automatically annotated for more than 920,000 songs on music- or video-sharing services and can readily be used by music-synchronized applications. Since errors are inevitable when elements are annotated automatically, Songle Widget takes advantage of a user-friendly crowdsourcing interface that enables users to correct them. This is effective when applications require error-free annotation. We made Songle Widget open to the public, and its capabilities and usefulness have been demonstrated in seven music-synchronized applications.**

***Keywords*-music synchronization; music understanding; web service; multimedia control**

I. Introduction

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 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.

The goal of our research is to provide a web-based open framework that enables people to control virtual or physical objects in synchronization with the playback of songs or music video clips on music- or video-sharing services on the web. Although it has become popular for people to enjoy music by watching a music video clip on the web, the full potential of the ever-increasing number of songs on the web

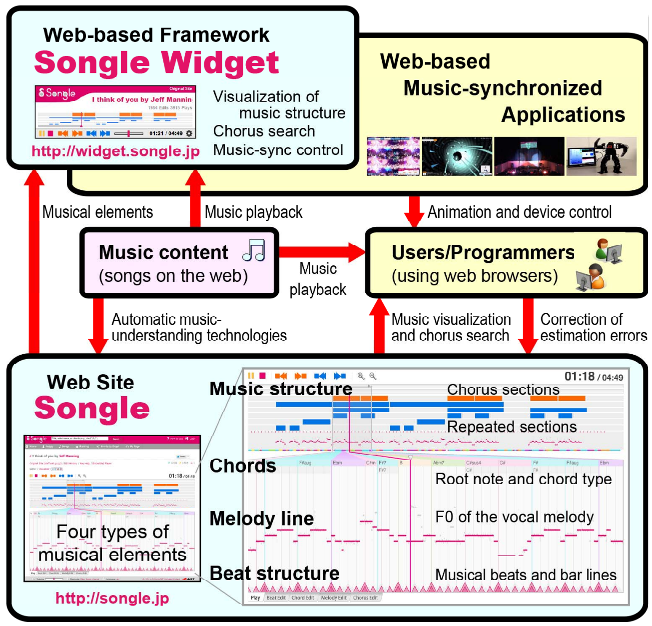

Figure 1. Overview of Songle Widget framework (http://widget.songle.jp) using Songle web site (http://songle.jp).

has yet to be realized. We want to make it possible to music-synchronize almost anything. While playing back a song on the *YouTube* video-sharing service, for example, we want people to enjoy various types of computer-graphic animation in which the motion, size, and color of various graphical objects on a web browser change in ways coordinated with the music being played back. We also want to control physical devices such as lighting devices and humanoid robots through such web-driven music synchronization.

To enable easy development of these kinds of music-sync web applications, we propose a new web-based framework called *Songle Widget* (Figure 1). To harness the power of music on the web, Songle Widget is implemented by using our existing web service called *Songle* (http://songle.jp) [1] that leverages music-understanding technologies to automatically analyze songs publicly available on the web and annotate four important types of musical elements (music structure, hierarchical beat structure, melody line, and chords) as shown in Figure 1. Songle Widget can use those annotation results to trigger changes in the motion and

978-1-5090-0379-2/15 $31.00 © 2015 IEEE
DOI 10.1109/ISM.2015.64

color of animation, or in the control of physical devices.

Given a wide variety of music, a drawback of music-understanding technologies is that errors are inevitable. Some users and music-sync applications require error-free annotations for high-quality music synchronization in public performances and presentations. To overcome this drawback, Songle Widget takes advantage of a Songle's crowdsourcing interface that enables users to correct errors in the annotation results. Songle Widget is thus considered a dynamic evolving framework that provides immediate access to up-to-date musical elements improved by users. This feature is original to our framework and has not been achieved by state-of-the-art music-understanding technologies [2]–[4], music-analysis web APIs such as the Echo Nest API (http://developer.echonest.com) and the AcousticBrainz project (http://acousticbrainz.org), or other systems for generating music-sync display, animation, or videos [5]–[7].

We have already made the Songle Widget framework open to the public (http://widget.songle.jp). To demonstrate its capabilities and potential, in this paper we present seven music-sync applications. While playing back a music video clip on YouTube, for example, those web 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.

II. Issues in Web-based Music Synchronization

Music synchronization on the web is difficult. When a user plays back a music video clip on popular video-sharing services, such as *YouTube* (http://www.youtube.com) and *Niconico* (http://www.nicovideo.jp/video_top), or uses their official embedded player, a user program on a web browser cannot receive and analyze audio signals of the clip. Real-time on-the-fly audio analysis for music synchronization is therefore not feasible and there is no web framework dedicated to it.

In this paper we propose a new approach for music synchronization, one that executes music-understanding technologies on music on the web in advance to obtain four important types of musical elements (Figure 1):

(i) Music structure (chorus sections and repeated sections)
(ii) Hierarchical beat structure (musical beats and bar lines)
(iii) Melody line (fundamental frequency (F0) of the vocal melody)
(iv) Chords (root note and chord type)

We represent each musical element as a series of time-stamped events (e.g., beat positions) so that Songle Widget can compare the time stamps with the current playback position to trigger a user program. Songle Widget uses the official embedded YouTube/Niconico player and its API to get the current playback position (elapsed time) while playing back a video clip. Music-sync applications can thus synchronize animation and physical devices with music videos on the web.

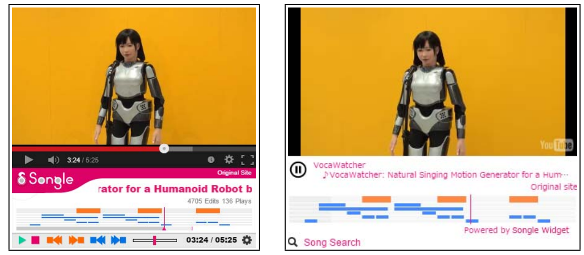

Figure 2. Screenshots of Songle Widget user interface with different appearances. This is equipped with the SmartMusicKIOSK interface proposed by Goto [8], which shows the music structure consisting of chorus sections (the top row) and repeated sections (the five lower rows). The colored sections in each row indicate similar (repeated) sections.

Since using the embedded YouTube/Niconico player to develop music-sync applications is not easy and requires advanced engineering skills and careful efforts, Songle Widget provides the event-driven architecture in which programmers can just write a *JavaScript* code for each event, such as each musical beat in a bar, the bar line (downbeat), and the chord change. The beginning or end of each chorus or repeated section can also be used as an event. In controlling physical devices such as robots (or even computer graphics), however, latency is problematic because the actual robot motion is usually delayed for tens or hundreds of milliseconds after being triggered by a motor-control signal. Songle Widget therefore supports a timing-offset mechanism in which a code for each event can be triggered before the actual event.

For high-quality synchronization, we can take full advantage of the crowdsourcing error-correction interface on Songle. With this interface, users can make corrections by candidate selection or direct editing. Any error corrections made to the musical elements can be instantly reflected on all applications using Songle Widget.

III. Songle Widget: Web-based Development Framework for Music-sync Applications

Figure 2 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 920,000 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.

The implementation overview is shown in Figure 3. Songle Widget was carefully designed and implemented so that it can be embedded into any web-based music-sync

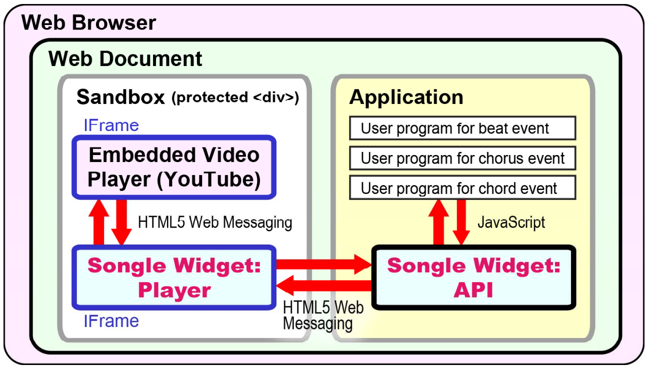

Figure 3. Overview of Songle Widget implementation (in the case of using beat, chorus, and chord events with the YouTube embedded video player).

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.

As shown in Figure 3, Songle Widget consists of two components: player and API. The Songle Widget player generates the user interface shown in Figure 2, 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.

IV. Case Study: Music-sync Applications

Since we have already 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.

A. 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 homepages 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.

Figure 4. Photographs showing stage lighting linked to music by using Songle Widget.

Figure 5. Photographs of music-sync robot dancing controlled by V-Sido x Songle.

B. 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.

C. 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 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 Figure 4, various types of lighting linked to music were projected in a stage-like setting.

D. 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.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.

E. 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 hierarchical 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

Figure 6. Screenshots of Songrium3D, the music visualizer for generating three-dimensional music-sync animation in real time.

flexible music-sync robot control had not been achieved before. In January 2015 Asratec Corp. and AIST published a press release and showed that this system can make three different types of robots dance in unison as shown in Figure 5.

F. 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/) [9], that features three-dimensional music-sync animation in which visual effects and the motions of various objects are triggered by events in the music structure and hierarchical beat structure of a music video clip as shown in Figure 6.

G. 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 by using Google Images. 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.

V. Discussion

In these seven applications the use of musical elements made possible advanced music synchronization unattainable by typical music visualizers that for music synchronization simply use the amplitude or spectrogram of audio signals. Each application has a different way of using the elements to control animation or physical devices (Table I), but the hierarchical beat structure (musical beats and bar lines) is always used to enable people to feel the music synchronization. The musical beats are usually used to show regular changes or motions. The bar lines (downbeats) are also useful for adding accents at a higher level. This basic synchronization is important but is insufficient because it often seems monotonous. The music structure (chorus sections and repeated sections) is therefore important to change the way of the music-sync control dynamically and temporally within a song. For example, most applications control the target in the most intense and appealing way during chorus sections. Some applications also make the target's motion more interesting by linking different combinations of repeated sections to different motions or changes. The chords (root note and chord type) are also used to change the mood of the control.

Table I
Events (musical elements) used in music-sync applications.

Application	Beat	Structure (Chorus)	Chord
Two-dimensional animation	V	V	V
Three-dimensional characters	V	V	V
Lighting devices	V	V	V
Melvie	V	V	
V-Sido x Songle	V	V	
Songrium3D	V	V	
Photo x Songle	V	V	

VI. Conclusions

We have described the Songle Widget framework that makes it easy for people to develop and enjoy music-sync applications such as music-sync animation, music-sync dancing of robots or animated characters, and music-sync lighting. Since more than 920,000 publicly available songs have already been analyzed by using music-understanding technologies, users can immediately choose among them for music-sync applications. To use error-free annotation of musical elements, users can also make voluntary error corrections on the web-based interactive editor.

The concept of the proposed framework is general and useful not only for music and its analysis but also for other types of multimedia content, such as images and videos, and their analysis. Future work will therefore include the development of a web-based development framework for applications driven by a mixture of any multimedia content on the web.

Acknowledgments

The authors would like to extend their appreciation to Takahiro Inoue for the player/API/application implementation of the Songle Widget framework, and Utah Kawasaki for the server/client implementation of the Songle web service. This work was supported in part by CREST, JST.

References

[1] M. Goto, K. Yoshii, H. Fujihara, M. Mauch, and T. Nakano, "Songle: A web service for active music listening improved by user contributions," in *Proc. of ISMIR 2011*, 2011, pp. 311–316.

[2] A. Klapuri and M. Davy, Eds., *Signal Processing Methods for Music Transcription*. Springer, 2006.

[3] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 668–696, 2008.

[4] M. Muller, D. Ellis, A. Klapuri, and G. Richard, "Signal processing for music analysis," *IEEE J. of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1088–1110, 2011.

[5] M. Goto and Y. Muraoka, "A beat tracking system for acoustic signals of music," in *Proc. of ACM Multimedia 94*, 1994, pp. 365–372.

[6] J. Foote, M. Cooper, and A. Girgensohn, "Creating music videos using automatic media analysis," in *Proc. of ACM Multimedia 2002*, 2002, pp. 553–560.

[7] F. Kurth, M. Müller, D. Damm, C. Fremerey, A. Ribbrock, and M. Clausen, "SyncPlayer — an advanced system for multimodal music access," in *Proc. of ISMIR 2005*, 2005, pp. 381–388.

[8] M. Goto, "A chorus-section detection method for musical audio signals and its application to a music listening station," *IEEE Trans. on ASLP*, vol. 14, no. 5, pp. 1783–1794, 2006.

[9] M. Hamasaki, M. Goto, and T. Nakano, "Songrium: Browsing and listening environment for music content creation community," in *Proc. of SMC 2015*, 2015, pp. 23–30.