

Songle Sync: 音楽に連動させて多様なデバイスを 大規模に制御できるプラットフォーム

尾形 正泰^{1,a)} 井上 隆広^{1,b)} 加藤 淳^{1,c)} 後藤 真孝^{1,d)}

概要: 本稿では、音楽の再生にインターネット経由で同期して多様な機器を制御することで一体感のある演出ができる大規模音楽連動制御プラットフォーム「Songle Sync」について述べる。従来、複数機器の同期制御は専用システムが個別に開発されて拡張性が乏しく、スマートフォン・パソコンのような汎用機器を自在に組み合わせたり、音楽連動演出を手軽に開発したりできなかった。Songle Sync では、数百台を超える多数のスマートフォン・パソコン・IoT デバイスを組み合わせて同時制御し、音楽と物理空間を融合した総合体験を生み出すことができる。さらに開発キットも提供し、音楽理解技術で得たビートやサビなどの音楽的要素に応じて表示・動きが変化する多様な演出を手軽に制作可能にする。2017年8月に一般公開し、汎用機器を含む100台以上の規模でCG映像・ロボット・照明などを音楽と連動させる実証実験により有効性を確認した。さらに、音楽ステージで数百人のスマートフォンに音楽に連動した映像を表示する実証実験も実施し、混雑した通信環境下で支障なく動作して音楽体験が拡張できることを確認した。

1. はじめに

本研究は、音楽に合わせて多種多様な機器が連動して一体感のある空間を、人々が手軽に創り出して体験できるようにするプラットフォームを構築・提供することを目的とする。そうしたプラットフォームがあれば、さまざまな場面で音楽に連動した演出を生み出すことができる。例えば、知り合い同士で好きな音楽を流しながら、各自のスマートフォンの画面に、その音楽に合わせて変化するCG（コンピュータグラフィクス）アニメーションを一斉に表示して、従来にはない一体感のある音楽体験を楽しむことが可能になる。より大規模な例では、ライブやイベントの会場で数百人の来場者が、会場の音楽と連動した演出をステージ上のスクリーン表示や各自のスマートフォンの画面で楽しむことも可能になる。さらに、ショッピングモールや飲食店等で、BGM（背景音楽）に連動して店舗内の壁面や床などのディスプレイ・照明を変化させたり、ロボットを踊らせたり、来場者のスマートフォンの画面を連動させたりして、一体感のある特別な空間を作り出すことが可能になる。

しかし従来は、そのようにさまざまな機器が音楽に連動

する演出を、ユーザの好みやイベントの内容に合わせて任意の楽曲で実施することは難しかった。また、ライブコンサートで多数のペンライトの光を一斉に変更する専用システムや、スマートフォンの画面がペンライトのように単色発光するような専用アプリケーションが個別に開発されたことはあったが、拡張性が乏しく多種多様な機器を自在に組み合わせることができなかった。そのため、一般の人々が個人のスマートフォンのような汎用機器を持ち寄って、好みの楽曲に応じた演出を、アプリケーションのインストールの手間なく楽しめるような仕組みはなかった。さらに、開発者が音楽に連動した演出を作ろうとすると、楽曲中で連動させる箇所を個別に手作業で指定した上で、多数の機器を音楽再生に同期させて制御するシステムを構築する必要があり、手間が大きく難易度が高かった。

そこで本研究では、音楽の再生にインターネット経由で同期して多様な機器を制御することで一体感のある演出ができる大規模音楽連動制御プラットフォーム「Songle Sync（ソングルシンク）」を提案する。Songle Sync は、図1のように、Web上のさまざまな楽曲の再生に合わせて、人々が既に所有する多種多様な携帯端末（スマートフォン、タブレット）・パソコン・Internet of Things (IoT) デバイス（インターネットに接続されて情報交換をしながら処理を進めるデバイス）を、インターネットを介して連動させて制御できるプラットフォームである。例えば、イベント会場などで数百人がプラットフォームの提供するQRコードを読

¹ 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology (AIST)

a) masa.ogata [at] aist.go.jp

b) takahiro.inoue [at] aist.go.jp

c) jun.kato [at] aist.go.jp

d) m.goto [at] aist.go.jp

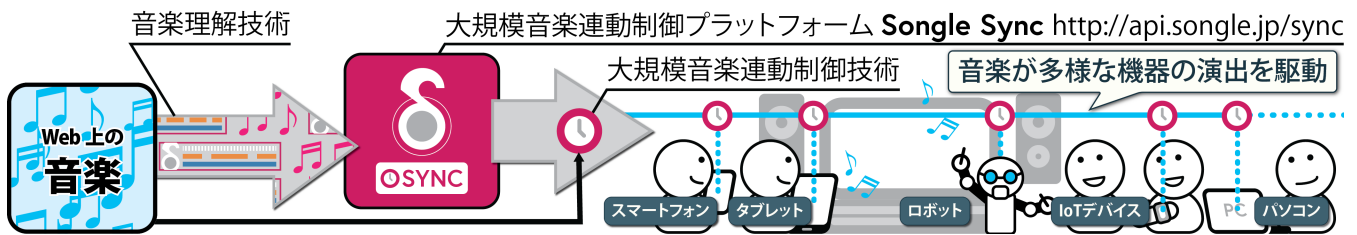


図1 音楽の再生に連動して複数の機器を制御できるプラットフォーム「Songle Sync (ソングルシンク)」(<http://api.songle.jp/sync>)

み取って Web アプリケーションにアクセスすると、会場で流れる楽曲と同期して手元の画面にアニメーションが表示され、多人数で一体感のある演出を楽しめる。

このような体験を可能にするために、我々は以下の3つの問題を解決した。

(1) 即座に連動: 楽曲を選ぶだけでどうやって多数の機器を一斉に連動させるか

音楽理解技術を用いて、膨大な楽曲のビート構造や楽曲構造等の音楽的要素(音楽情景記述)を事前に自動推定しておく。その結果に基づく演出が実施されるように同期対象の機器群を自動制御することで、ユーザが楽曲を選べば即座に連動制御が可能になった。

(2) スケーラブルに連動: 多種多様な汎用機器をいかに大規模に連動させるか

Songle Sync のサーバに対して、多数の機器がクライアントとしてアクセスする際に、インターネットを通じて非同期かつ断続的に通信するだけで動作する仕組みを実現した。これにより演出実施中の通信帯域が少なく、混雑した通信環境でも運用可能になり、スケーラビリティが増した。

(3) 容易に開発: 大規模な音楽連動演出をいかに容易に開発可能にするか

プログラマ(開発者)向けのチュートリアルを含む開発キットを提供した。ビートのような各音楽的要素をイベントとして、イベント駆動型プログラムを書くだけで、Songle Sync が音楽を再生しながら他の機器と自動的に連動制御してくれる。プログラマが、他の多数の機器の存在や音楽再生制御を意識する必要がない点が優れている。

我々は Songle Sync (<http://api.songle.jp/sync>) を、チュートリアルサイト (<http://tutorial.songle.jp/sync>) と共に 2017 年 8 月 2 日に一般公開し、実証実験を開始した。既に、100 台以上の多種多様な機器を同時につないで有用性を確認した。同年 9 月 2 日には、幕張メッセで開催されたイベントの 40 分間のライブステージで初めて活用された。来場者のスマートフォンによる最大同時接続数が 275 人以上の状態、それらの画面上の映像がビートや小節、サビ区間等に応じて一斉に変わる演出に成功した。

2. Songle Sync プラットフォームの概要

Songle Sync は音楽連動制御による演出を体験・提供・開発支援するためのプラットフォームである。そのプラットフォームを利用して、多くのプログラマが多種多様な音楽連動制御アプリケーション(以下、音楽連動制御アプリと略す)を開発して公開し、一般ユーザがそれを利用して楽しむようにすることを目指している。本章ではその狙いを音楽体験、プラットフォーム、開発支援に大別して述べる。

2.1 多数の機器が連動して可能になる新たな音楽体験

音楽は、音だけでも楽しめるが、音楽と映像が連動するミュージックビデオや音楽に合わせて踊るロボットのように、音楽に連動した演出と共に楽しむことも多い。Songle Sync では、そうした演出をさらに大規模に多種多様な機器を連動させて実現可能にすることで、新たな音楽体験を生み出し、文化的・産業的に多様な価値を生むことを狙っている。

そのための第一歩として、既に Web サイト上で公開されていて、不特定多数の人々がアクセスして聴くことができる楽曲群のうち一曲を再生しながら、スマートフォン・パソコン・IoT デバイス等の機器を多数組み合わせで同時制御できるようにする。その楽曲の中身を事前に自動解析しておき、その解析結果に基づいて各機器の表示や動き等の演出を生成することで、楽曲の再生にすべての機器が一斉に連動する制御が可能になる。このように Web 上の音楽をユーザが選んで、それに多数の機器を連動させて新たな音楽体験を生み出すための汎用的な仕組みは、これまでなかった。さらに、図 2 のように、スマートフォン・パソコンの画面表示だけでなく、照明機器や光る衣装、IoT 化した小型ロボット等の多様な機器を、音楽のビート等に合わせ一斉に変化させることで、音楽と物理空間を融合した総合体験を生み出すことができる。以上述べてきたような体験を可能にするアプリケーションや利用シーンについては、3 章で詳述する。

2.2 プラットフォームの主要機能を手軽に体験

Songle Sync のプラットフォームが提供する主要な機能を一般ユーザが Web ブラウザだけで手軽に体験できるよ



図2 Songle Sync を利用した各種機器の制御事例（既に開発して動作している機器の様子）

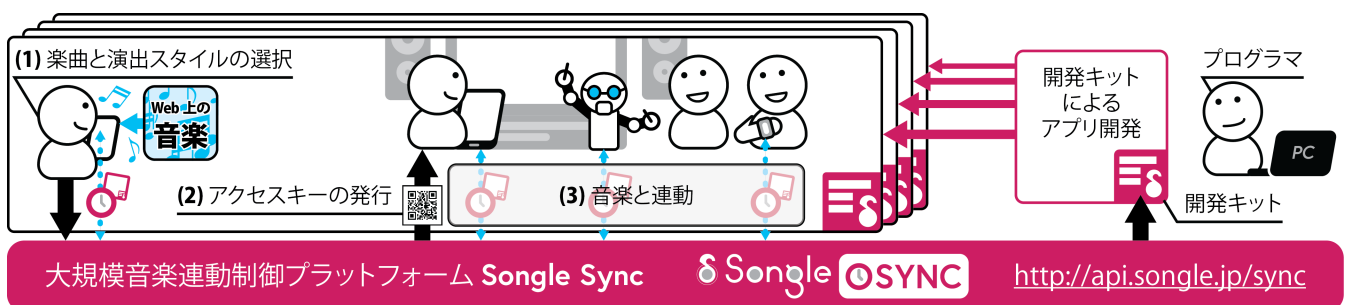


図3 Songle Sync の利用方法

うに、チュートリアルの一部として入門的な音楽連動制御アプリを提供する。このアプリは、図3に示した以下の手順で利用できる。これらの機能の具体的な実現手法や技術については4章で述べる。

(1) 楽曲と演出スタイルの選択

ユーザはスマートフォンやパソコンのWebブラウザを使って、Webサイト上で公開されている楽曲（MP3形式の音響信号ファイルや楽曲・動画共有サービス上の楽曲・動画）と、楽曲全体にわたる演出の仕方を決める数種類の「演出スタイル」を好みに応じて選択する。その選択後に音楽再生プレーヤが表示されて、Webサイト上にある楽曲がストリーミング再生されるとともに、ディスプレイには選択した演出スタイルに応じて、その楽曲のビートやサビなどに合わせて変化するCGアニメーションが表示される。こうして、楽曲を目と耳の両方で楽しむことができる。

(2) アクセスキーの発行

ユーザが他の人々と一緒に同じ音楽と演出を楽しむために、Songle Syncのサーバを介して、この音楽再生プレーヤと連動するためのアクセスキー（文字列、URL、QRコード）を発行できる。

(3) 複数の汎用機器がインターネットを介して音楽に連動

ユーザからアクセスキーを受け取った他の人々が、それを自分のスマートフォンやパソコンのWebブラウザ

に入力すると（QRコードの場合にはQRコードリーダーで読み取ると）、Songle Syncのサーバに自動的に接続される。すると、同じ楽曲の音楽再生プレーヤが表示され、場所を問わず全員が同じ瞬間に同期して楽曲とアニメーションを楽しむことができる。楽曲の途中で接続しても同期でき、ユーザが楽曲を変更したり再生・停止したりすれば、それに連動して、他の人々の楽曲やアニメーションも追従する。

また、他の人々が接続する際には、楽曲再生とアニメーション表示のどちらか一方だけを選択してもよい。例えば後者を選択すれば、スピーカーで鳴らしている音楽と一緒に聞きながら、その場にいる人々のスマートフォンの画面に同じアニメーションを表示するといった使い方ができる。

2.3 プログラムのための開発支援

プログラマによる開発を支援して敷居を下げ、多種多様な音楽連動制御アプリが生み出されることを期待して、Songle Syncではチュートリアルを含む開発キットを提供する。従来、専用機器ではないインターネット接続可能な汎用機器を用いて大規模な音楽連動制御を可能にするのは技術的に難易度が高く、プログラマが開発するのは容易ではなかった。それは、音楽の再生や演出を同期制御しようとしても、個別に非同期通信が断続的に繰り返される通信

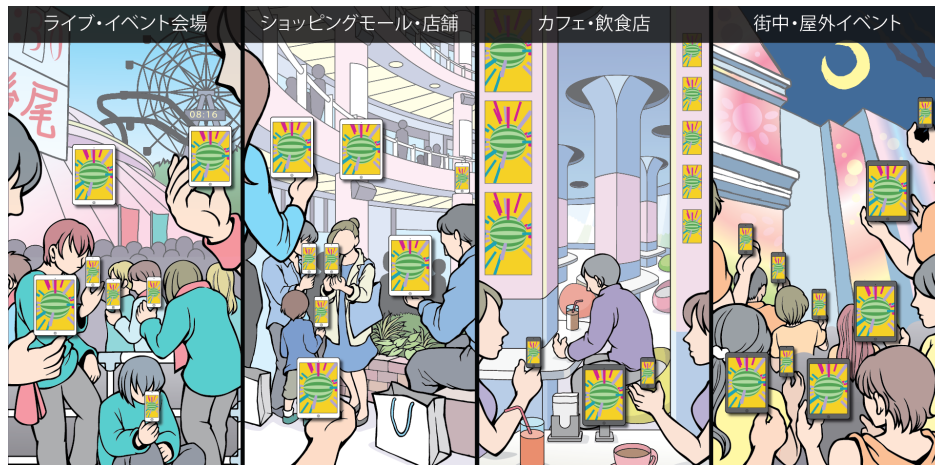


図4 さまざまな利用シーンで大規模な音楽連動制御が活用される際のイメージ画像

環境と多様な実行環境においては、連続的な再生が必要な音楽とそれに合わせた演出が機器ごとにずれやすく、技術的に困難であったからである。ユーザが、異なる演出を手軽に組み合わせたり、演出を変更したりすることも従来の技術では実現されておらず、大規模な音楽連動制御に基づく新たな演出を、プログラマが容易に開発できるような開発キットも存在していなかった。

Songle Sync では、プログラマが開発キットを使って演出スタイルのサンプルプログラムを変更するだけで、楽曲を再生して演出を描画する音楽連動制御アプリを容易に開発できる。プログラマは、1 台の機器を制御するつもりでプログラミングをするだけでよく、音楽を再生しながらインターネットを介して他の機器と連動する制御は、Songle Sync が自動的におこなってくれる点が優れている。そのため、プログラマは複数台の機器制御を意識する必要がなく、複数の機器間の通信や時刻同期に関して専門知識のないプログラマであっても、音楽連動制御アプリの開発に取り組むことができる。これらを可能にする開発キットの詳細については 5 章で述べる。

3. Songle Sync が可能にする多様な音楽連動制御・利用シーン

Songle Sync は、これまでにない音楽体験を生み出し、さまざまな利用シーンで価値を創出できる可能性を持っている。以下では、そうした Songle Sync が切り拓く未来について議論する。

3.1 音楽理解技術による解析結果に基づく多様な連動制御

Songle Sync では、個々の楽曲に対して手作業でビート時刻等のラベリングをしなくても、Web 上の楽曲を選択して再生すれば即座に、多種多様な携帯端末・パソコン・IoT デバイスが連動制御できるという従来にはない体験を可能にする。そのためには、事前に Web 上の楽曲の中身（ビート

や小節、サビ区間など）を音楽理解技術（音楽の音響信号中のさまざまな要素を自動的に推定できる技術）で解析しておく必要がある。

本研究ではその解析を、我々の研究グループがこれまでに研究開発して一般公開してきた能動的音楽鑑賞サービス「Songle (ソングル)」(<http://songle.jp>) [1-3] を活用して実現した。これは、Web 上の楽曲の中身を音楽理解技術で解析して可視化するサービスで、既に 111 万曲を解析済みである。MP3 形式の音響信号ファイルに加え、コンテンツ投稿サイトの「ピアプロ」(<http://piaapro.jp>) や、動画共有サービスの「ニコニコ動画」(<http://www.nicovideo.jp/>) と「YouTube」(<https://www.youtube.com/>) 等の楽曲の音響信号を対象に、ビート構造（ビート、小節）、楽曲構造（サビ区間、繰り返し区間）、コード等を自動解析できる。Songle は解析誤りをユーザーが手動で訂正して共有できるインターフェースも備えており、Songle Sync による音楽連動制御で用いた楽曲に解析誤りがあって困る場合には、Songle 上で誤り訂正ができる。

これらの解析結果を利用することで、楽曲ごとに異なる中身に連動した演出を楽しむことができる。再生したい楽曲が自動解析されていない場合は、Songle に楽曲の URL を登録して自動解析することで、その楽曲も Songle Sync で再生できるようになる。楽曲は元の Web サイト上からユーザのブラウザに直接ストリーミング再生される仕組みとなっている。

2.2 節で紹介した入門的な音楽連動制御アプリやその「演出スタイル」は、一般的な Web ブラウザから利用できて汎用性が高いため、異なる OS (Windows, macOS) を搭載したパソコンや、異なる OS (iOS, Android) を搭載したスマートフォン・タブレットのように多種多様な機器を連動制御することができる。さらに、小型軽量端末 Raspberry Pi や、IoT デバイス（ロボットや照明機器など）を活用し

た演出もできる。

6.1 節でも後述するように、実際に図 2 のように多様な利用事例で有効性が確認されている。画面表示では、パソコン、iPhone、iPad、iPod touch、Android スマートフォンなどの汎用機器を用いて、画面上で音楽に連動したアニメーションを表示できた。空間演出制御では、IoT 化した照明機器や発光ダイオード (LED) を内蔵した光る生活雑貨 (花瓶、キャンドル) などを用いて、部屋・環境の照明・光を音楽に連動して変化させられた。ファッション制御では、IoT 化した LED を装着・内蔵した光る衣装や光る腕輪を用いて、音楽に連動した発光制御ができた。さらにメカ制御では、IoT 化した小型ロボットや電動カーテンを用いて、音楽に連動させながらロボットを動かしたり、カーテンを開閉したりできた。今後も、他の利用事例が開発されることが期待できる。

3.2 さまざまな利用シーンにおいて価値を創出

人々が Songle Sync を利用すると、楽曲の再生と同時に、個人が所有している携帯端末・パソコンなどのスピーカーから同時に同じ楽曲が流れ、しかもそれらの画面に、その音楽に合わせて動く多様で変化に富むアニメーションが表示できる。従来の技術では実現できなかったこのような音楽体験は、今後、例えば以下のような利用シーンで新たな価値を創出できる可能性がある (図 4)。

(a) ライブ・イベント会場

会場に設置されたスピーカーから流れる音楽の再生に合わせて、来場者のスマートフォンの画面に変化に富んだアニメーションが表示するような新たな演出が可能になる。さらに、会場の複数の大型スクリーンにも同じアニメーションを表示させたり、ステージ登壇者の衣装や腕輪を発光させたりするような演出もできる。ライブ開始前やイベントの入場待機列などの場面で来場者に一体感を持って楽しんでもらう演出として、音楽連動制御を活用しても効果的である。

(b) ショッピングモール・店舗

ショッピング中の BGM に合わせて通路や店舗の壁面や床などのさまざまな箇所に演出ができる。大型ディスプレイや照明機器、プロジェクタの投影映像などを連動させた演出で、モールや店舗全体に一体感を出すことができる。さらに来店者のスマートフォンと連動させても効果的である。

(c) カフェ・飲食店

飲食時の BGM に、間接照明やディスプレイ・照明機器を連動させることで、楽曲に合わせてムードを変えたり、店舗ごとに異なる印象を与えたりするなどの空間演出が可能になる。

(d) 街中・屋外イベント

街中全体を使った屋外イベントのような機会に、既に

設置されているさまざまなディスプレイや照明、プロジェクションマッピングなどを組み合わせて、音楽連動制御を実施することも考えられる。主催者側が、特別な演出を体験可能な QR コードを事前に用意して配布すれば、その街を訪れている人々が同時に音楽に連動した演出を楽しむことができる。

4. プラットフォームの実現手法

Songle Sync のプラットフォームに必要な機能群を実現するために、上述の Songle による音楽理解技術に加え、大規模音楽連動制御技術を新たに開発し、それらを融合させることで、音楽に連動した演出を容易に利用・開発可能にした。以下では、大規模音楽連動制御技術の具体的内容について述べる。

4.1 一般的な常時接続アーキテクチャの問題点

本研究の大規模音楽連動制御のように、多数の機器を連動制御したい場合には、マスターと呼ばれる制御の主導権を握る側の機器から、多数のスレーブと呼ばれる制御される側の機器に短時間で通信 (情報伝達) が可能な状態を確保する常時接続アーキテクチャが一般的である。マスターがスレーブの状態を変えたいときに、そのタイミングで毎回変更指示を通信し、各スレーブはそれを受信してから状態を変更する (例えば、描画する、動かす、光らせる)。

ライブやイベント会場等で多数の照明機器 (舞台照明) を制御するために用いられる通信プロトコル DMX512 は、このアーキテクチャに基づいている。また、ライブコンサートで多数のペンライトの光を一斉に変更する専用システムの多くもこのアーキテクチャに基づいており、マスターからの制御情報を会場内の電波アンテナから発信し、各ペンライトがその電波を受動的に受信して光る。

他にも、スマートフォンの画面がペンライトのように単色発光するような専用アプリケーションは同様のアーキテクチャのことが多く、専用アプリケーションと専用送信サーバとの間に常時通信可能なコネクション (論理的な通信路) を WebSocket プロトコル等を用いて確立しておき、マスターからの色指定指示を専用送信サーバから受信して、アプリケーションが受動的に画面更新する。

しかし、上記の常時接続アーキテクチャで時間精度の良い音楽連動制御を実現するには、リアルタイムに近い短時間で安定した通信が必須となる。マスターからスレーブへの通信遅延時間、スレーブ同士の受信時刻のずれが、音楽に対するずれに直結するからである。Songle Sync では、スマートフォン等の人々が既に所有する多種多様な汎用機器を中心に、多数の機器を自由自在に組み合わせる必要があり、多様な実行環境・通信環境で動作させるためにも、そうした通信を前提にできない。

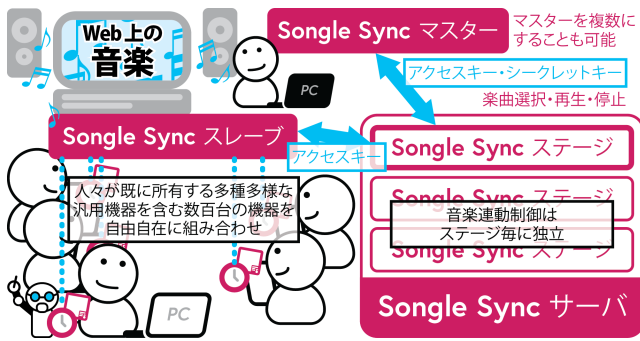


図5 Songle Sync の機能およびシステム構成図

4.2 Songle Sync の自律制御アーキテクチャ

本研究では、常時接続アーキテクチャは採用せず、Songle Sync の Web サーバと、マスター、多数のスレーブが断続的に非同期な通信をしながら、各スレーブが楽曲の再生時刻に応じて自律的にイベント駆動で演出を呼び出して連動する自律制御アーキテクチャを用いる。この新たな仕組みでは、マスターとスレーブの間で基準時間軸（基準となる時間軸）が共有されていることを前提とする。この共有は、マスターとスレーブのそれぞれの内部時計が、基準時間軸に対してどれぐらいずれているかという時刻差を測定することで実現できる。

まず、楽曲の再生をマスターが指示する際には、その楽曲が基準時間軸上でいつ再生開始するか（したか）を設定する。すると、楽曲中の音楽的要素は事前の自動解析により確定しているため、それらの音楽的要素に対応するすべてのイベント（例えば、ビート、小節の先頭、サビ区間の開始と終了）の基準時間軸上での時刻（イベント時刻）も確定する。

次に、自立的に動作する各スレーブは、内部時計の現在時刻が基準時間軸上のどの時点かを随時把握し、それが上記の各イベント時刻に到達したら、そのイベントに対して定義されている演出を実行する。この仕組みならば、イベントのタイミングで通信が発生しないため、スレーブの台数が増えても、スレーブ間で演出のタイミングがずれにくい点が優れている。

音楽連動制御の仕組みが機種依存にならず、相互接続を容易にし、スマートフォン等に専用アプリケーションをインストールする手間をかけさせないために、この自律制御アーキテクチャは、Web 標準技術のみに基づいて実現する必要がある。

4.3 Songle Sync ステージ、シークレットキー、アクセスキー

上記の自律制御アーキテクチャで前提としたマスターとスレーブ間での基準時間軸の共有を実現し、さまざまな音楽連動制御を可能にするためのシステム構成図を図5に示す。Songle Sync プラットフォーム上では、これまで説明

してきたマスターを「Songle Sync マスター」、スレーブを「Songle Sync スレーブ」、Web サーバを「Songle Sync サーバ」と呼ぶ。Songle Sync サーバを多数のユーザ・利用事例が相互に干渉せずに利用できるように、利用事例ごとに「Songle Sync ステージ」（以下、ステージ）と名付けた管理の枠組みを設けた。

Songle Sync サーバ上でユーザ（ステージ管理者）が申請すると、そのユーザ専用のステージを利用事例毎にいくつでも作成できる。各ステージに対して、サーバから「シークレットキー」と「アクセスキー」が発行される。これらは乱数をエンコードした文字列を用いて表現される。両者を用いてステージに接続するとマスターとみなされ、後者のみだとスレーブとみなされる。各ステージは、両者のキーと、マスターとスレーブの接続状況（識別 ID、接続台数）、楽曲の再生状況（楽曲 URL、再生・停止状態、基準時間軸上の再生開始時刻）を管理する。

マスターは演出全体の時間軸を制御できるため、シークレットキーは利用事例の実施者（イベントの主催者・運営者）のみが把握し、他の一般の利用者に知られてはならない。一方、スレーブ用のアクセスキーは一般の利用者が全員入手する必要がある。そのためにはアクセスキーを含む URL や、その QR コードを配布すると効果的であり、ステージにはそのための QR コード生成機能もある。

4.4 Songle Sync マスターの動作

マスターはステージの管理権限を持ち、ステージに対してそこで再生する楽曲 URL を指定した上で、その再生や停止の状態の設定、基準時間軸上の再生開始時刻の設定ができる。マスターが楽曲の先頭から再生を開始したいときには、再生状態に設定し、その再生開始時点でのマスターの内部時計の現在時刻を、基準時間軸上の再生開始時刻に変換して設定する。シーク（楽曲内の再生位置移動）の場合には、シーク位置に応じて、基準時間軸上の再生開始時刻を過去に設定する。例えば楽曲の先頭から1分の位置にシークすると、基準時間軸上の再生開始時刻は1分前に設定される。停止したいときは、停止状態に設定するだけでよい。

複数のマスターをステージに接続してもよく、その場合には上記の楽曲の再生状況をどのマスターからでも更新できる。これにより、例えば数百人の音楽ステージ等で利用しているときに、1台のマスターに不具合が生じてても、他のマスターをバックアップ手段にして制御ができて問題が起きにくくなる。

なお、マスターは次に説明するスレーブのすべての機能を兼ね備えており、マスターを操作するユーザは、楽曲の再生音を聴いたり演出を見たりしながら、ステージの楽曲の再生状況を変更できる。

4.5 Songle Sync スレーブの動作

各スレーブがステージに接続した後は、定期的（一定時間以上のランダムな時間間隔）に楽曲の再生状況を取得し、その楽曲 URL に対応する楽曲解析結果（すべてのイベント）を取得する。再生状態であれば、楽曲の再生開始時刻からすべてのイベント時刻を算出して確定し、スレーブの内部時計の現在時刻が、それに対応する基準時間軸上のイベント時刻に達したら、その演出を自律的に実行する。停止状態であれば、何も演出を実行しないようにする。

スレーブが各イベントでどのような演出を実行するかは、スレーブ側のプログラムとして実装されている（例えば事前にダウンロードされている）ので、イベントの時点で一切通信する必要がない点が常時接続アーキテクチャとは大きく異なる。マスターとスレーブや、スレーブ同士も一切通信する必要がなく、サーバ上のステージの楽曲の再生状況のみを介して連動する。そのため、スレーブが楽曲再生中に途中から接続して参加したり、途中で脱退したりすること容易である。

スレーブが楽曲も再生する場合には、楽曲 URL に基づいて適切な音楽再生プレーヤを呼び出し（生成し）、基準時間軸上の再生開始時刻から現在再生されているべき楽曲中の再生時刻を求めて、その時刻を同期して再生するように音楽再生プレーヤを制御する。

4.6 基準時間軸の共有

マスターとスレーブ間で共有する基準時間軸は、仮想の時間軸でもよいが、現在の実装では Songle Sync サーバの内部時計とした。マスターとスレーブのそれぞれは、Songle Sync サーバと通信して、自分の内部時計と基準時間軸との時刻差（オフセット）を測定する。この測定方法は、NTP (Network Time Protocol) からヒントを得ており、基本的には、送信側が自分の内部時計の送信時点での時刻を送り、受信側が自分の内部時計の受信時点での時刻との時刻差を計算する。これは実際の内部時計の時刻差に加え、通信遅延時間も含まれている。そこでこれを送受信を入れ替えて往復させることで通信遅延時間を打ち消し、内部時計の時刻差を求める。通信環境によっては通信遅延時間が極端に大きな外れ値も含まれることを考慮し、これを複数回繰り返して、内部時計と基準時間軸との時刻差をロバスト推定する（中央値を求める）。

この基準時間軸の共有のための通信は、ステージに接続して最低一回実施すれば、その後は各スレーブは楽曲の再生状況だけを取得しながら、自律的に動作が可能である。しかし、通信環境の変動を考慮して、現在の実装では、マスターとスレーブのそれぞれが非同期に、ステージ接続後に定期的に基準時間軸の共有（基準時間軸に対する内部時計の時刻差の測定）を実施するようにした。

4.7 マスターとスレーブの接続状況の管理

各ステージに対するマスターとスレーブの接続台数は動的に変動するため、それぞれを区別する識別 ID（乱数をエンコードした文字列）によって管理する。初回の接続時に発行され、以後は、マスターとスレーブ側が識別 ID を自動付与して毎回通信する（ユーザは意識する必要がない）。ステージ側では識別 ID と最終接続時刻の一覧を記録しておき、一定時間接続がなければ、その識別 ID を無効にする。無効な識別 ID を付与して通信した場合には、初回接続と同様に扱われる。

これにより、同一機器が断続的に通信を繰り返しても、同時接続台数を把握できる。また、識別 ID の発行数を制限すれば、あるステージの接続台数に上限を設けることも可能になる。

4.8 通信量削減によるスケーラビリティ向上

楽曲再生開始後の演出中は、個々のスレーブからは、楽曲の再生状況の取得と、内部時計の時刻差の測定の二種類の通信しか発生しない。いずれも通信量が少ない上に、それぞれの通信タイミングも任意で非同期でよく、演出に直結するイベント時刻には無関係な点が優れている。これらの性質は、音楽連動演出をする機器の数を増やす、つまり、スケーラビリティを向上する上で有利になるように、意図的に設計されている。

それに加えて、いずれの通信の頻度も自在に変更できる特長を持っている。頻度を下げることでさらに通信量を削減し、スケーラビリティを向上できる。まず、楽曲の再生状況の取得頻度を下げると、マスターによって楽曲を再生開始したりシークしたりした際に、スレーブの反応がばらつくリスクがあるが、最初に楽曲再生開始した後はマスターが何も操作しないのであれば、それでも支障がない。しかも Songle Sync サーバが返す楽曲の再生状況には、現在時刻のような変動する値が一切なく、マスターが変更を加えていないときには毎回同じ通信内容になるので、サーバ側でのキャッシュが効きやすい利点もある。次に、内部時計の時刻差の測定頻度を下げると、測定時にたまたま通信環境が悪化した際に影響を受けやすくなったり、内部時計の精度が低いときにずれやすくなったりするリスクがあるが、実行環境や演出の内容によっては支障がない場合も多い。仮に混雑した通信環境下で一時的に通信が不能になっても演出は続行できる。

上記で述べてきた Songle Sync の通信はすべて、Songle Sync サーバに対する一般的な HTTP 通信で実装されている。個々の通信が終わるごとに切断し、次の通信まで HTTP 接続（コネクション）を保持しない。これは音楽連動制御の規模を数万台以上に大規模化する際に有利で、コネクションを保持し続ける WebSocket プロトコルでは OS のファイルディスクリプタが枯渇する恐れがある。Songle

Sync では、既に DNS ラウンドロビンにより複数サーバで負荷分散しており、個々の通信も低容量で短時間なので大規模化しやすい。

5. プログラマのための開発支援

プログラマが大規模音楽連動制御の利用事例（音楽連動制御アプリ）を容易に開発するための支援として、Songle Sync の開発キットは図 6 のように、API（Application Programming Interface）やそのドキュメントに加え、それを効率的に習得するためのサンプルプログラム、チュートリアルで構成される。本開発キットを用いたプログラミングには以下の特長がある。

(a) プログラマは音楽に同期させるための時間管理を意識する必要がない。

小節の先頭ではメリハリのある演出、サビに入ったら盛り上がる演出のように、楽曲の再生状況に応じたイベント駆動なプログラムを記述するだけで、楽曲中の再生時刻が各イベントの時刻に達すると自動的に指定した演出が実行される。しかも楽曲ごとに手作業でイベントを設定する必要がなく、Songle の解析結果（ビート、小節、サビ区間、繰り返し区間、コードなど）をイベントとして利用できる。

(b) プログラマは他の機器との連動制御を意識する必要がない。

1 台の機器を念頭においてアプリをプログラミングするだけで、他の多数の機器と自動的に連動できる。しかも、スマートフォンやパソコンの Web ブラウザで動作するアプリ、照明を制御するアプリ、ロボットを制御するアプリのように、個別に開発されたさまざまな種類のアプリを、ある楽曲の再生と一緒に連動させることができる。このようにプログラマは、通信や他の機器との連動制御を全く意識せずに、大規模な音楽連動制御を実現できる。

以上により、ユーザにとって自分の機器が容易に他の機器と連動する新たな音楽体験を可能にするだけでなく、プログラマにとっても大規模な音楽連動制御による演出を容易にプログラミングできる新たな体験を可能にする。

5.1 Songle Sync API とサンプルプログラム

Songle Sync プラットフォームを Web 標準技術で実装し、開発キットの Songle Sync API を JavaScript で提供した。プログラマは、JavaScript で記述した音楽連動制御アプリを幅広い実行環境（JavaScript が実行できる一般的な Web ブラウザ、IoT デバイス上で動作する JavaScript 環境 Node.js など）で動作させることができ、ユーザは、QR コードなどで手軽に利用できる。

API の使用方法は、サンプルプログラムとして GitHub (<https://github.com/SongleJp/songle-widget>) で公開し

た。Web ブラウザ上で動作するプログラムだけでなく、IoT デバイス用にハードウェア（Raspberry Pi, Intel Edison 等）を必要とするプログラムも用意した。

5.2 オンラインチュートリアル

Web ブラウザで試せるチュートリアルを、一般ユーザとプログラマの両者を意識して、以下の 4 つのステップに分けて設計した。

(1) 音楽連動制御を体験

このページを開いて演出スタイルの選択肢の中から一つを選ぶと、Songle Sync スレーブが動作し、楽曲が鳴りながら CG アニメーションが表示される。このために、デモ楽曲が無限に繰り返し自動再生される専用の Songle Sync ステージを用意した。複数の機器の Web ブラウザから開けば、それらがすべて同一楽曲に連動していることが確認できる。

(2) 好きな曲で連動

2.2 節 で述べた入門的な音楽連動制御アプリに対応している。

(3) Songle Sync アプリを開発

このページ内のコードエディタに、マスターとスレーブのサンプルプログラムがセットされており、それを書き換えて Web ブラウザ上で実行結果を確認しながら、アプリ開発を習得できる。

(4) Songle Sync アプリを公開

開発した Songle Sync アプリを GitHub で公開する手順や、Node.js を使って Songle Sync マスターとして動作する Web サーバを開発・公開する方法を紹介している。

以上の設計では、Web 上でのプログラミングのためのチュートリアルに関する調査研究 [4] も参考にした。

6. 実証実験

Songle Sync プラットフォームの有効性を確認するために、100 台以上のスマートフォン・パソコン・IoT デバイス（ロボット・照明等）を音楽連動制御する実証実験と、音楽ステージで数百人のスマートフォン音楽連動制御する実証実験を実施した。

6.1 実証実験 1: 多種多様な機器の音楽連動制御

図 2 に示した多様な利用事例のすべてを同じ部屋で一斉に動作させる実証実験を実施した。そのために図 7 に示した複数種類の音楽連動制御アプリを開発し、本実証実験のための専用の Songle Sync ステージにスレーブとして接続させた。同ステージに接続したマスターから再生する楽曲を部屋のスピーカーから鳴らし、10 種類、100 台以上の機器がその楽曲の中身に同期することを確認した。

以下では、これらの利用事例を 4 つに分類して説明し、

開発キット API (Application Programming Interface)

Songleの解析結果に基づくイベント駆動プログラミング

```

player.on("play", listener);
player.on("beatEnter", listener);
player.on("chorusSectionEnter", listener);
...

```

多様な機器を容易に連動させられるJavaScriptライブラリ

```

var SW = require("songle-widget"), player = new SW.Player({ ... });

```

プログラマ

Webブラウザ搭載機器
Chrome, Safari, Firefox, Edgeなど

JavaScript動作機器
Raspberry Pi, Intel Edisonなど

チュートリアル

<http://tutorial.songle.jp/sync>

サンプルプログラム

- Webブラウザ向け (携帯端末・パソコン)
- IoTデバイス向け
- ロボット向け

<https://github.com/SongleJp/songle-widget>

図6 手軽で容易な大規模音楽連動制御を可能にする開発キット



図7 多様な音楽連動制御アプリが利用されている状況でのシステム構成図

得られた結果を述べる。

6.1.1 画面表示

音楽に連動したCGアニメーションを、パソコン (Windows, macOS), Phone, iPad, iPod touch, Android スマートフォン, Raspberry Pi の画面に表示した。このための音楽連動制御アプリを Web 上に用意し、それを各機器の Web ブラウザで読み込むと演出が開始される。

CGアニメーションの演出スタイルは5種類用意し、ビートや小節の先頭で変化するだけでなく、楽曲構造のサビ区間で繰り返し区間で種類を切り替え、単調ではない映像表現となるように工夫した。また、描画性能が低い Raspberry Pi のような端末にも配慮し、WebGL を用いない軽量の演出スタイルも用意した。

6.1.2 空間演出制御

IoT化した照明機器やLED内蔵の生活雑貨(花瓶, キャンドル)を開発して制御し、部屋・環境の照明・光を音楽に連動して変化させた。そのために、DMX512対応の照明機器をDMX拡張ボードを介してArduinoに接続した。花瓶とキャンドルは、市販機器に対してLEDを置き換えたり信号線を変更したりしてArduinoに接続した。その上で、いずれもパソコン上のJavaScriptプログラムからArduino経由でLEDの色を制御した。

6.1.3 ファッション制御

LEDで装飾した光る衣装や、個別のLEDを制御して光る腕輪を開発して制御し、音楽に連動して光らせた。光る衣装は、服にLED (Adafruit NeoPixel) 計200個を貼り付けてArduinoに接続した。光る腕輪は、腕輪にLEDを埋め込んでArduinoに接続した。その上で、ビートや小節等に連動して色と明るさが変化するように制御した。

6.1.4 メカ制御

音楽に連動して、IoT化した小型ロボットを動かす、電動カーテンを開閉させた。小型ロボットには市販のRapiroとWebmoを使用した。RapiroとWebmoには、モータ制御用にそれぞれRaspberry PiとIntel Edisonが内蔵されており、それらの上でNode.jsを動かして、JavaScriptプログラムを実行した。

カーテンは、市販の電動カーテンのリモコン上で、カーテン開閉ボタンの回路電圧をArduinoで制御した。これにより、パソコン上のJavaScriptプログラムからArduino経由で開閉を制御でき、サビ区間の開始時点でカーテンを開き、終了時点で閉じるようにした。

6.1.5 実証実験結果

100台を超える機器を同時にスレーブとして制御した状態で、マスターで再生した楽曲に同期した制御が可能であることを確認した。Songle Sync ステージで管理しているマスターとスレーブの接続状況の接続台数を画面に表示し、それが110台を超え続けている状態を維持した。

ただし、同一の部屋に設置した100台以上の機器がSongle Sync サーバに円滑にインターネット接続するためには、有線LANと無線LANを併用し、高性能なネットワーク機器を用いる必要があった。さらに、冷却ファンを持たない機器が熱暴走を起こすのを防ぐため、室温管理とファン等による冷却・排熱が不可欠だった。その結果、すべてのスレーブが繰り返し安定に動作し、実証実験時のSongle Sync サーバの負荷も充分低く抑えられていることを確認した。

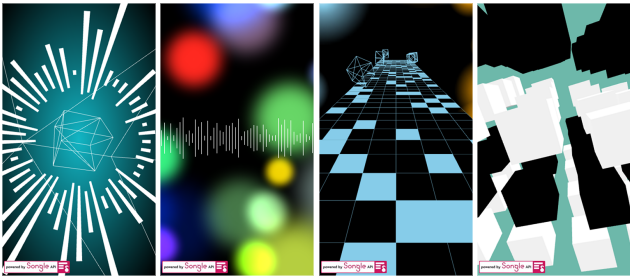


図8 DJ ステージの実証実験で表示されたCGアニメーションの例



図9 実証実験中のDJステージ会場の様子

6.2 実証実験 2: 音楽ステージでの音楽連動制御

幕張メッセで開催されたイベント『初音ミク「マジカルミライ 2017」』の企画展DJステージ『keisei ×ミクダヨーDJステージ - Songle Sync コラボ -』（2017年9月2日開催）において、Songle Syncによる演出を来場者が体験する実証実験を実施した。QRコードが書かれたポストカードを来場者に配布し、各来場者がそのQRコードをスマートフォンから開くと、ステージの音楽に同期して画面のCG映像が一斉に変わる演出を提供した。その結果、40分間の演奏中に数百人が同時アクセスしても支障なく楽しめることを確認した。

6.2.1 DJステージ特別版による演出

今回のDJステージ専用の音楽連動制御アプリを開発し、40分間に使用される各楽曲に適したCGアニメーションを用意した。その際、JavaScriptのプログラムのサイズが小さく、極力多くの機種で動作するように工夫した。そのスクリーンショット例を図8に示す。

会場のスピーカーから流れる音楽に同期して、このCGアニメーションがステージ上の大型ディスプレイと、来場者のスマートフォンに表示されるようにした。開演前は、ステージ上のディスプレイにQRコードを表示し、参加を呼びかけた。

6.2.2 実証実験結果

実証実験中のライブ会場の様子を図9に示す。QRコードが書かれたポストカードを900人以上の来場者に配布し、途中まで、あるいは、途中から部分的に参加する来場者も多かったが、40分間の最初から最後まで Songle Syncサーバが安定して動作し、意図していた演出に成功したことを確認した。のべ600人以上は接続していたと推測され、最大同時接続数として、少なくとも275人以上の来場者のスマートフォンが同時に接続されたことを確認した。会場内の4G回線等による来場者のスマートフォンのインターネット接続には支障がなく、Songle Syncサーバの負荷も十分に低いことが確認できた。これらは実証実験を実施しなければ確認できない点であり、サーバに対する負荷テストは事前に可能でも、実際に数百台のスマートフォンを接続した状態での動作検証は困難なので、本実証実験は有意義であった。

実証実験中の来場者の様子を見て気づいた点、考察した点を以下に述べる。

- ペンライトを持参している来場者が多く、音楽に合わせて片手でペンライトを振りながら、反対側の手にスマートフォンを持って Songle Sync の演出を楽しんでいる姿が多く見られた。
- ステージから遠い来場者は、図9の上の写真のようにアーティストやステージ上のディスプレイを充分に見ることができないことがあり、手元でステージ上と同じCGアニメーションを見られる Songle Sync の画面を注視する姿が目立っていた。
- ステージに近い来場者は、一時的には参加しても、Songle Sync を注視し続けることは少なかった。スマートフォンの画面を見続けて、目の前のアーティストを見ないのは考えにくく、その行動は自然だと考えられる。実際、アーティスト側から見たときにも、目の前の来場者が下を向いて画面を見ていたら望ましくない可能性が高く、必ずしも来場者全員に参加してもらうことを前提にしない方がよいと考えられる。3.2節の利用シーンで提案したような、ライブ開始前やイベントの入場待機列などの場面で用いるのであれば、こうした問題が起きにくい。
- 来場者が同じCGアニメーションを見るだけでなく、ステージと連動したインタラクティブな要素を音楽連動制御アプリに加え、アーティストが呼びかけて反応するような演出をするのも効果的だと考えられる。

7. 関連研究

音楽に連動した表現や制御は高い価値を持ち、人々の関心も高い。そのため、音楽を自動解析した結果に基づいて、音楽に連動した映像やアニメーションを生成する研究も数多く取り組まれてきた。例えば、音楽のビート等の中身に

応じて三次元 CG キャラクタを踊らせる技術 [5,6] は、既にさまざまな形で開発されて産業界で一般的になっている。音楽の中身に応じて、ビデオや写真を編集してミュージックビデオを生成する研究事例も多い [7-9]。楽譜や歌詞を含むさまざまな視覚的な素材を音楽に連動させて表示したり [10]、歌詞の内容に基づいて画像を表示したり [11, 12] する研究もなされてきた。

そうした音楽に連動した表現・制御をプログラマが容易に開発できるように、我々は、上述の能動的音楽鑑賞サービス Songle [1-3] で曲の中身を解析した結果を、1台のパソコンの Web ブラウザ上で利用して音楽に連動した演出ができる開発フレームワーク「Songle Widget (ソングルウィジェット)」(<http://widget.songle.jp>) [13] を研究開発し、公開してきた。この Songle Widget の API は、音楽視聴支援サービス「Songrium (ソングリウム)」(<http://songrium.jp>) [14, 15] や歌詞アニメーション制作支援サービス「TextAlive (テキストアライブ)」(<http://textalive.jp>) [16] でも活用され、音楽に連動した映像表示を可能にしている。Songle Widget は音楽連動制御のためのブラウザベースの開発フレームワークであり、本研究の Songle Sync の土台となっているが、Songle Sync のように複数の機器の制御はできなかつた。

また、プログラミングによって即興で音楽を奏でる Live Coding においては、音楽だけでなく、ビートなどに同期して表示する映像を同時にプログラミングすることも珍しくない。例えば Gibber [17] では Web ブラウザ上でそのような即興演奏が可能である。また、そうしたプログラミング環境をインターネット越しに共有し、複数人でセッションできるようにする試みも存在する [18]。しかしながら、これらはあくまで Live Coding のための試みであり、本研究のように既存の好みの楽曲に合わせて多種多様な機器を制御するアプリケーション開発はできない。

本研究でも音楽イベントでの活用例を議論したが、来場者にスマートフォンで参加を促す音楽イベントは既に実施されている。2013年1月に東京で開催されたイベント「FULL CONTROL TOKYO」では、来場者がスマートフォンに専用アプリケーションをインストールし、事前に決められた特定の楽曲の進行に合わせて、各画面上で単色の色が変わる、という演出が実施されていた。また、スマートフォンを入力コントローラーとする演出も取り入れられていた。2016年8月にオランダで開催されたイベント「Lowlands Festival」では、Smartphone Orchestra (<https://smartphoneorchestra.com>) では、楽曲の構成要素を事前に分割しておき、その音を来場者のスマートフォンのスピーカーから別々に鳴らすことで、全体で音楽を構成するコンサートが実施されていた。音楽に連動して生成される映像表現はなく、各画面は単色で様々な色が表示されていた。専用アプリケーションはなく、Web ブラ

ウザだけで動作する点は Songle Sync と共通である。しかし、これらの事例で用いられている技術は、Songle Sync のような汎用のプラットフォームではなかつた。

Songle Sync では、一般ユーザがいつでもどこでも手軽に大規模音楽連動制御を体験でき、プログラマが開発キットで手軽に開発に参加できる点が、従来の研究事例や取り組みとは大きく異なる。

8. おわりに

本稿では、Web ブラウザから利用できる拡張性の高い大規模音楽連動制御プラットフォーム「Songle Sync」を提案した。新たに開発した大規模音楽連動制御技術を音楽理解技術と融合させることで、多種多様な機器を用いた大規模な音楽連動制御を手軽に利用・開発できる。実証実験では、ハードウェアの性能や特性が異なる多種多様な 100 台以上の機器を同時に制御でき、音楽ステージで数百人のスマートフォンの画面を一斉に変える制御が可能であることを確認した。

今後も Songle Sync の持続的な研究開発・運用に取り組み、スケーラビリティの検証や、より多様な音楽連動制御アプリでの有効性の確認を進めていく予定である。そして、インターネット経由で音楽に連動制御される端末・デバイス群によって構成されるネットワークを、音楽版の Internet of Things (IoT) の意味を込めて「Internet of Musical Things (IoMT)」と名付け、音楽と物理空間を融合した利用事例を開拓していく。そのために産業界等と連携しながら、ライブ・イベントや、ショッピングモール・店舗、カフェ・飲食店、ファッション、街中など、さまざまな利用シーンに応用展開していければと考えている。

謝辞 Songle Sync の音楽連動制御アプリの実装を担当して頂いた石田 啓介 氏 (CG アニメーション)、土田 修平 氏 (電飾衣装制御)、中村 裕美 氏 (照明機器制御)、および、Songle Sync の土台となる Songle サービスのサーバとクライアントの実装を担当して頂いた川崎 裕太 氏に感謝する。本研究の一部は JST ACCEL (JPMJAC1602) の支援を受けた。

参考文献

- [1] Goto, M., Yoshii, K., Fujihara, H., Mauch, M. and Nakano, T.: Songle: A Web Service for Active Music Listening Improved by User Contributions, *Proc. of ISMIR 2011*, pp. 311-316 (2011).
- [2] 後藤真孝, 吉井和佳, 藤原弘将, Mauch, M., 中野倫靖: Songle: 音楽音響信号理解技術とユーザによる誤り訂正に基づく能動的音楽鑑賞サービス, *情処学論*, Vol. 54, No. 4, pp. 1363-1372 (2013).
- [3] 後藤真孝, 吉井和佳, 中野倫靖: Songle: Web 上の楽曲の中身を音楽理解技術で推定する能動的音楽鑑賞サービス, *情処研報音楽情報科学*, Vol. 2013-MUS-100, No. 16, pp. 1-9 (2013).
- [4] Kim, A. S. and Ko, A. J.: A Pedagogical Analysis of Online

- Coding Tutorials, *Proc. of ACM SIGCSE 2017*, pp. 321–326 (2017).
- [5] Goto, M. and Muraoka, Y.: A Beat Tracking System for Acoustic Signals of Music, *Proc. of ACM Multimedia 94*, pp. 365–372 (1994).
- [6] Sauer, D. and Yang, Y.-H.: Music-driven Character Animation, *ACM Trans. Multimedia Comput. Commun. Appl.*, Vol. 5, No. 4, pp. 1–16 (2009).
- [7] Foote, J., Cooper, M. and Girgensohn, A.: Creating Music Videos Using Automatic Media Analysis, *Proc. of ACM Multimedia 2002*, pp. 553–560 (2002).
- [8] Hua, X.-S., Lu, L. and Zhang, H.-J.: Automatic Music Video Generation Based on Temporal Pattern Analysis, *Proc. of ACM Multimedia 2004*, pp. 472–475 (2004).
- [9] Shamma, D. A., Pardo, B. and Hammond, K. J.: MusicStory: A Personalized Music Video Creator, *Proc. of ACM Multimedia 2005*, pp. 563–566 (2005).
- [10] Kurth, F., Müller, M., Damm, D., Fremerey, C., Ribbrock, A. and Clausen, M.: SyncPlayer — An Advanced System for Multimodal Music Access, *Proc. of ISMIR 2005*, pp. 381–388 (2005).
- [11] Geleijnse, G., Sekulovski, D., Korst, J., Pauws, S., Kater, B. and Vignoli, F.: Enriching Music with Synchronized Lyrics, Images and Colored Lights, *Proc. of Ambi-Sys 2008*, pp. 1–8 (2008).
- [12] Funasawa, S., Ishizaki, H., Hoashi, K., Takishima, Y. and Katto, J.: Automated Music Slideshow Generation Using Web Images Based on Lyrics, *Proc. of ISMIR 2010*, pp. 63–68 (2010).
- [13] Goto, M., Yoshii, K. and Nakano, T.: Songle Widget: Making Animation and Physical Devices Synchronized with Music Videos on the Web, *Proc. of IEEE ISM 2015*, pp. 85–88 (2015).
- [14] Hamasaki, M., Goto, M. and Nakano, T.: Songrium: Browsing and Listening Environment for Music Content Creation Community, *Proc. of SMC 2015*, pp. 23–30 (2015).
- [15] Hamasaki, M., Goto, M. and Nakano, T.: Songrium: A Music Browsing Assistance Service with Interactive Visualization and Exploration of a Web of Music, *Proc. of WWW 2014*, pp. 523–528 (2014).
- [16] Kato, J., Nakano, T. and Goto, M.: TextAlive: Integrated Design Environment for Kinetic Typography, *Proc. of ACM CHI 2015*, pp. 3403–3412 (2015).
- [17] Roberts, C. and Kuchera-Morin, J.: Gibber: Live coding audio in the browser, *Proc. of the International Computer Music Conference (ICMC)*, pp. 64–69.
- [18] Roberts, C., Yerkes, K., Bazo, D., Wright, M. and Kuchera-Morin, J.: Sharing Time and Code in a Browser-Based Live Coding Environment, *Proc. of the First International Conference on Live Coding*, pp. 179–185 (2015).