

# Nightbird Audience Node: 観客がVJの 即興映像プログラミングを支援するシステム

小淵 豊<sup>1</sup> 加藤 淳<sup>2</sup> 濱崎 雅弘<sup>2</sup> 後藤 真孝<sup>2</sup> 福地 健太郎<sup>1</sup>

**概要:** 本稿では、Visual Jockey (VJ) によるプログラミングを用いた即興映像パフォーマンスにおいて、スマートフォンから映像の操作に介入できるインタフェースを観客に提供することで、観客に映像の操作を楽しんでもらいながら、VJのパフォーマンスを支援するシステム Nightbird Audience Node を提案する。即興プログラミングを用いた VJ プレイは、複数の映像をクロスフェードなどの決まった効果で切り替える一般的な VJ プレイに比べ、より複雑で魅力的な表現が可能となる。しかしながら、VJ が即興プログラミングをしている最中は、プログラムが未完成で映像の変化が単調になりがちである。そこで、提案システムは、プログラムの振る舞いに関わるパラメータ調整を、VJ だけでなく観客がスマートフォンを用いてできるようにする。また、提案システムは、複数の観客がパラメータ操作した結果を統合したり、観客自身が新たなパラメータを提案したりできる。これにより、観客は映像表現を好みに応じて調整でき、楽しみながら即興映像パフォーマンスに参加できる。結果として、即興映像パフォーマンスが、より多彩な映像の変化を持ったものになることが期待できる。

## Nightbird Audience Node: Adding Audience Support to VJ Performance Based on Visual Programming

YUTAKA OBUCHI<sup>1</sup> JUN KATO<sup>2</sup> MASAHIRO HAMASAKI<sup>2</sup> MASATAKA GOTO<sup>2</sup> KENTARO FUKUCHI<sup>1</sup>

**Abstract:** In this paper, we present *Nightbird Audience Node*, an extension to visual programming environments for visual jockey performances. It allows audiences to participate in the performance with their smartphones to help visual jockeys create programs. While typical visual jockey performances connect multiple videos with simple visual effects such as cross-fading, improvisation of visual jockey programming has a potential to create more complex and appealing visuals. However, such performances tend to produce simple effects and could be boring in the beginning when the program is incomplete. To address the issue, we propose Audience Node that allows audiences to tweak parameters of program components through their smartphones. It is capable of simultaneous operations of a single parameter by multiple audiences, whose results are merged to have a single value. It also allows the audiences to propose pseudo parameters to suggest the VJ to implement new Audience Nodes. As a result, the audiences can customize visual effects, participating in programming with fun to enrich visual performances.

### 1. はじめに

Visual Jockey (VJ) とは、ナイトクラブなどにおいて、音楽に合わせて映像パフォーマンスを観客に提供する者を

指す。多くの VJ は、予め用意された複数の動画をクロスフェードなどの限られた数の効果で切り替えるスタイルの VJ プレイをしている。一部の VJ は、即興で映像プログラミングを行うことで、クロスフェードなどに留まらず、複数の動画を同時に加工して表示するような、バリエーションに富んだ映像表現を実現している。しかし、即興でこのようなプログラミングを行うことは容易ではない。特に、複雑なアルゴリズムを実装している最中は、プログラムが

<sup>1</sup> 明治大学  
Meiji University

<sup>2</sup> 産業技術総合研究所  
National Institute of Advanced Industrial Science and Technology (AIST)



図 1 Nightbird のインターフェイス全体 (左) と出力結果 (右).

Fig. 1 Overview of Nightbird (left) and an example output from the program (right).

不完全で映像の変化が単調になりがちである。

そこで本研究では、観客にプログラムの振る舞いに関わるパラメータを操作するためのインタフェースを提供し、観客に楽しみながらプログラミングに参加してもらうことで、より多彩な映像の変化を持った映像パフォーマンスを支援するシステム Nightbird Audience Node を提案する。

## 2. 関連研究

### 2.1 VJ ソフトウェア

多くの VJ ソフトウェアは、大量の動画ライブラリから複数の動画を選択し、クロスフェードなどの効果を用いてつなげながら表示することで VJ プレイを行う。Müller らは、VJ プレイの準備中及びパフォーマンス中に、事前にメタデータが付与された映像メディアを効率的に管理する技術 [1] を提案した。「あのタグで待ってる」[2] は、事前に動画へタグを付与しておくことで、VJ プレイ中に関連する動画を効率的に提示でき、連想ゲームのように動画を選択できる VJ ソフトウェアである。

本研究では、動画の選択手法ではなく選択済み動画の加工手法に着目しており、これらの既存手法と組み合わせることも可能である。

### 2.2 ライブコーディング・ビジュアルプログラミング

ライブコーディングとは、コードの変更が即座に実行結果として反映される開発環境を利用した即興パフォーマンスである。ライブコーディングは主に音楽の即興パフォーマンスに用いられる手法である。多くのライブコーディング用開発環境は主に音楽の即興パフォーマンス用に設計されている。例えば、ChucK[3] はその先駆けであり、SuperCollider[4] など後続研究も多い。

ライブコーディングの考え方は映像の即興パフォーマンスにも適用され、音楽と映像を同時にプログラミングできる Overtone[5] や、Web ブラウザ上で動作する Gibber[6]、映像表現に特化しておりカメラを利用した Augmented Reality の表現が可能な Fluxus[7] などが開発されている。

これらのライブコーディング用開発環境はテキスト

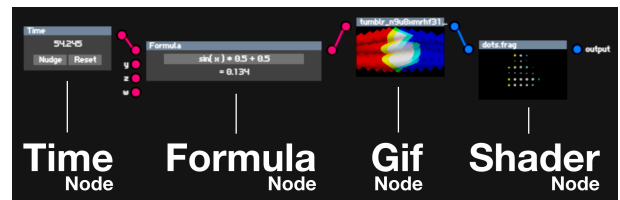


図 2 Nightbird における Node と Link の例。

Fig. 2 Examples of Nodes and Links on Nightbird.

ベースのプログラミング言語を利用する必要がある。Pure Data[8] や vvvv[9]、本研究で利用する Nightbird は、ビジュアルプログラミング言語を利用する開発環境であり、文字を入力せず直接操作で映像表現を行うことができる。

### 2.3 観客参加型パフォーマンス

観客を巻き込んだ即興パフォーマンスに関する研究は、これまでも数多くなされている。Carpenter らは、観客に赤と緑の札を挙げさせ、撮影した映像を画像処理することで得票比率を計算できるシステム [10] を提案している。Maynes-Aminzade らは観客参加型インタラクティブシステム [11] を提案し、実験を通じて観客参加型エンタテインメントについて考察した。磯山らは、一般参加者が演者となり、参加者が身につけたセンサ、壁や床に投影された映像、スピーカーとヘッドホンからの音を組み合わせ、演技にインタラク션을付与するシステム [13] について報告した。Laursen らは、観客から送信された Like アイコンやコメントを DJ インタフェース上に表示する手法 [14] を提案している。本研究は、観客をパフォーマンスへ参加させることで、観客を楽しませるだけでなく、観客が VJ のパフォーマンスを直接支援できる点に特徴がある。

## 3. Nightbird

本章では Nightbird について紹介する。Nightbird のスクリーンショットを図 1 に示す。Nightbird は映像パフォーマンスのためのビジュアルプログラミング言語であり、GitHub でオープンソース公開されている [15]。

図 2 は Nightbird を用いて構築した簡単なプログラムで

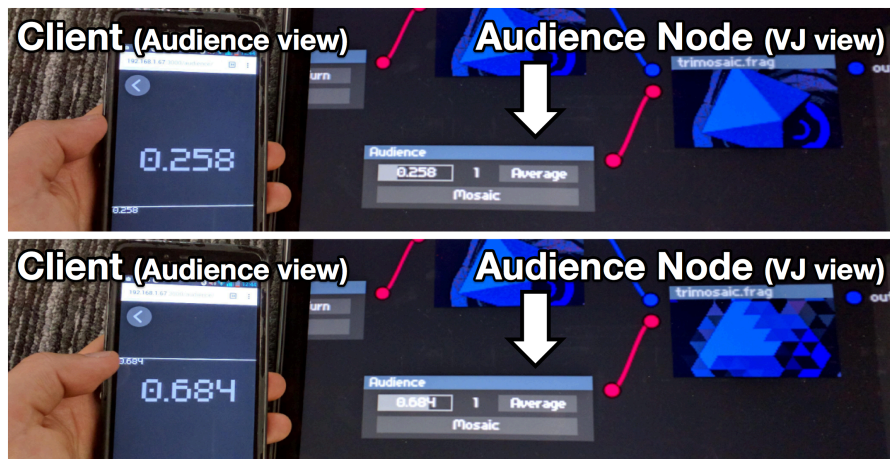


図 3 スマートフォン上で動く Audience Node Client と、Nightbird 上の Audience Node.  
 Fig. 3 Audience Node Client on smartphone and Audience Node on Nightbird.

ある。ウインドウのような外見を持つ 4 つの要素をそれぞれ Node と呼び、Node 同士を結ぶ赤や青の線を Link と呼ぶ。各 Node は入出力端子を持つ。Node 同士を Link で連結することにより、Node から Node へ様々な情報を伝達し、プログラムを構築することができる。赤い Link は数値の情報、青い Link は映像の情報をそれぞれ示す。

図 2 のプログラムは Time Node, Formula Node, GIF Node, Shader Node という 4 種類の Node を組み合わせたものである。時間を出力する Time Node と、入力数値に任意に定義できる数式を適用させ出力する Formula Node を組み合わせることにより、時間に  $\sin$  関数を適用させ、0.0 から 1.0 を往復する数値出力を得ている。この数値出力を、GIF Node という、入力数値で GIF アニメーションの再生フレームを制御できる Node に入力することにより、アニメーションの動きを往復させている。この映像出力を、入力映像にフィルタを適用させる Shader Node と組み合わせることにより、さらに複雑な映像効果を得ている。

Nightbird はこの他にも、定数を出力する Value Node, 複数の映像を合成する Layer Node, 能動的音楽鑑賞サービス Songle[16] による音楽理解情報のうち、ビート情報と楽曲構造情報を数値として出力する Songle Node など、様々な Node が実装されている。Nightbird は、このような Node の組み合わせを次々につなげていくことにより、図 1 のような複雑で多彩な映像表現を可能とする。さらに、JavaScript が扱えるユーザは新たな Node を実装することもでき、表現の幅をより拡張することができる。

#### 4. Audience Node

提案手法は、Audience Node という、Nightbird の新たな Node として実装されている。

#### 4.1 VJ 用インタラクション

Audience Node は、VJ が他の Node と同様に扱うことができる (図 3 中央)。Audience Node は、Time Node のように 1 つの数値の出力を持ち、観客がスマートフォンを用いて操作する Audience Node Client (以下 Client, Client については後述する) から送られた数値が出力され、Formula Node や GIF Node などの他の Node の数値入力につなげることができる。Audience Node は、Client に対してサーバの役割を持っている。VJ は Audience Node に対して名前を指定することができ、観客に Audience Node の役割を名前により伝えることができる。

Audience Node は Activate 状態と Deactivate 状態を切り替えることができる。Audience Node を追加した時点では Deactivate 状態となっていて、Audience Node の出力が他のどの入力にもつながっていない未実装の Audience Node や、不具合が生じている Audience Node を、観客が操作できないように設定できる。Audience Node の実装が完了した時点で、Activate 状態にすることにより、観客にその Audience Node を操作してもらえるようになる。

#### 4.2 観客用インタラクション

Audience Node Client は、観客が持っているスマートフォン上で動作する Graphical User Interface (GUI) である。Client は Audience Node に対してクライアントの役割を持っている。観客はスマートフォンの Web ブラウザを通じて Client にアクセスし、Audience Node が出力する数値を操作できる。

観客はまず、Client に表示される選択画面 (図 4) から自分の操作したい Audience Node を選択する。各 Audience Node は VJ によって指定された名前前で表示されているため、観客は名前から Audience Node の役割を把握することになる。Audience Node は、VJ が追加した時点では

Deactivate 状態となっているため、観客は Deactivate 状態の Audience Node を選択し操作することはできない。

選択画面で Audience Node を選択すると、操作画面 (図 3 左側) が表示される。操作画面は上下に動かせるスライダーとなっていて、観客はこのスライダーを操作し、選択した Audience Node から出力される数値を 0.0 から 1.0 の間で入力できる。

### 4.3 複数の観客による Audience Node の操作

ひとつの Audience Node の操作には複数の観客が参加できる (図 5)。このとき、Audience Node は、複数の観客の入力した数値を統合し、ひとつの数値として出力しなければならない。本システムはこの統合手法として、Average mode と Turn mode という二種を用意し、VJ が用途に応じて選べるようにした。

Average mode は、操作に参加するすべての観客の意見を取り入れることができるモードである。Average mode を選択した場合、全参加者の操作する数値を平均して出力する。Turn mode は、参加者を時間などで区切って Audience Node の操作に割り当てることができるモードである。Turn mode を選択した場合、Audience Node に数値の入力が一つ増え、Time Node などの他の Node から出力された数値を Audience Node の入力に接続することができる。数値を入力することにより、参加者の操作する数値のうちいずれか一つを、時間などの任意の数値により決定することができる。

### 4.4 観客による Audience Node の追加

本システムは、観客が VJ の用意した既存の Audience Node を物足りないと感じた場合のため、観客が Audience Node を追加し、操作したいパラメータを VJ に提案でき

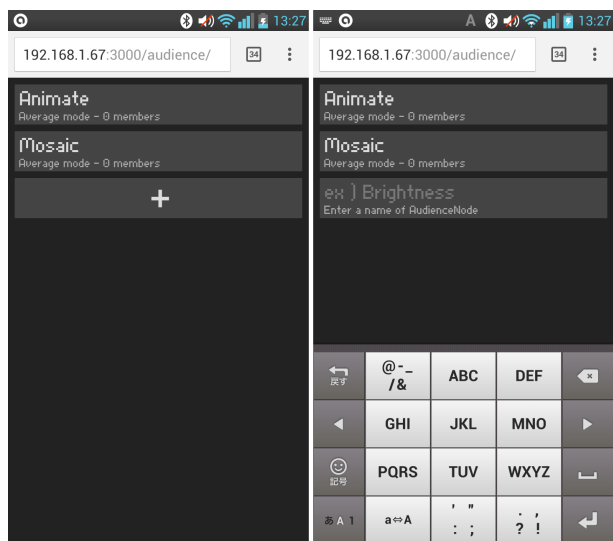


図 4 Audience Node Client の Audience Node 一覧画面。  
 Fig. 4 Audience Nodes listed on an Audience Node Client.

るようにした。

観客は、Client の Audience Node 選択画面 (図 4) の一番下にある + ボタンより、新たな Audience Node を追加できる。このとき、Audience Node には VJ が追加した場合と同様に名前をつけることができる。どのように Audience Node を命名すれば良いかわからない観客向けに、Audience Node の名前を入力フォームには、いくつか名前が例示される。観客が追加した Audience Node は、Deactivate 状態で VJ 側のインタフェースに現れ、VJ が追加した Audience Node と同様に扱うことができる。

これにより、観客が VJ が用意した既存の Audience Node が物足りないと感じた場合に、観客が操作したいパラメータを VJ に伝えることができる。VJ が観客より提案された Audience Node を気に入れば、VJ がその Audience Node を実装し Activate 状態にすることで、観客は提案した Audience Node を実際に操作することができる。

## 5. 考察

### 5.1 観客によるパラメータ操作

プログラミングができないユーザがプログラム上のパラメータを操作する手法は、Kato ら [12] がアニメーションのパラメータを調整するためのスライダーなどの GUI を容易に実装できるフレームワークを、プログラマに対して提供する手法を提案している。本研究は、この手法をビジュアルプログラミング言語を用いたライブコーディング環境に適用し、即興プログラミングの最中に、観客が操作可能な新たなパラメータを動的に追加できるようにした。これにより、VJ があらかじめ用意していなかった Audience Node を映像パフォーマンス中に追加することによって、より即興性の高い映像パフォーマンスが可能となる。

### 5.2 観客による映像パフォーマンスに対する提案

本システムは、4.4 節で述べたように、VJ が用意した Audience Node だけでなく、観客が新たな Audience Node を追加できるようにした。もし観客が既存の Audience Node では物足りないと感じ、他の映像パラメータを操作

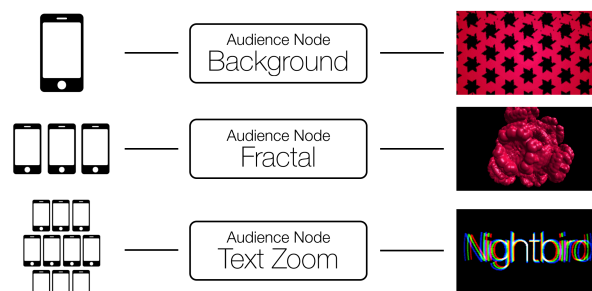


図 5 各 Audience Node には同時に複数の観客が参加できる。  
 Fig. 5 Multiple audiences can simultaneously join each Audience Node.



したいと感じた場合、観客は操作したい映像パラメータ名を Audience Node に命名し、VJ に希望を伝えることができる。

観客がパフォーマーに対してフィードバックを送信する手法は、Laursen ら [14] が DJ インタフェース上に観客から送信された Like アイコンやコメントを表示する手法を提案している。本研究では、Audience Node によって観客が映像についてのより具体的な提案を送信することができ、提案が受諾された場合は、観客が Audience Node を操作することによって、より多彩な映像表現が可能となる。

## 6. おわりに

本稿では、VJ の即興プログラミングによる映像パフォーマンスにおいて、観客がスマートフォンを用いて映像の操作に介入できるインタフェースを提供することで、観客に映像の操作を楽しんでもらいながら、VJ のパフォーマンスを支援するシステム Nightbird Audience Node を提案した。Audience Node は、Nightbird に限らず、Node を複数つなぐことでプログラムを構築する、一般的なビジュアルプログラミング言語に対して実装できるものである。

今後は、より適切な複数の観客からの値の統合手法や Client の操作方法、観客からの VJ へのフィードバック手法を検討し、評価実験を通してより有用なシステムへと発展させたい。さらに、Audience Node の手法を、テキストベースのプログラミングを含む、より一般的なプログラミング環境へと適用させ、リアルタイムにおけるプログラマとユーザのコミュニケーションについて考察したいと考えている。

## 参考文献

- [1] P. Müller, S. Müller Arisona, S. Schubiger-Banz and M. Specht. Interactive Media and Design Editing for Live Visuals Applications. In *International Conference on Computer Graphics Theory and Applications (GRAPP)* (2006).
- [2] あのタグで待ってる.  
<http://www.cobaltbombalphaomega.com/anotag/>.
- [3] G. Wang and P. R. Cook. ChuckK: A concurrent, on-the-fly audio programming language. In *Proceedings of the International Computer Music Conference* (2003).
- [4] J. McCartney. Rethinking the Computer Music Language: SuperCollider. *Computer Music Journal* (2002).
- [5] S. Aaron. Overtone.  
<http://overtone.github.io/>.
- [6] C. Roberts and J. Kuchera-Morin. Gibber: Live Coding Audio in the Browser. In *Proceedings of the International Computer Music Conference* (2012).
- [7] D. Griffiths. Fluxus.  
<http://www.pawfal.org/fluxus/>.
- [8] M. Puckette. Pure Data.  
<https://puredata.info/>.
- [9] vvvv.  
<http://vvvv.org/>.
- [10] R. Carpenter and L. Carpenter. Cinematrix.  
<http://www.cinematrix.com/>.
- [11] D. Maynes-Aminzade, R. Pausch and S. Seitz. Techniques for interactive audience participation. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces* (2002).
- [12] J. Kato, T. Nakano and M. Goto. TextAlive: Integrated Design Environment for Kinetic Typography. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems* (2015).
- [13] 磯山直, ウォーリー 木下, 出田 怜, 寺田 努, 塚本 昌彦. "観客参加型演劇 YOUPLAY におけるインタラクション設計", エンタテインメントコンピューティングシンポジウム 2014 論文集, Vol.2014, pp.168-179 (2014).
- [14] L. F. Laursen, M. Goto and T. Igarashi. A Multi-Touch DJ Interface with Remote Audience Feedback. In *Proceedings of the 22nd ACM International Conference on Multimedia* (2014).
- [15] Nightbird.  
<https://github.com/FMS-Cat/nightbird>
- [16] 後藤 真孝, 吉井 和佳, 藤原 弘将, Matthias Mauch, 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", 情報処理学会論文誌, Vol.54, No.4, pp.1363-1372 (2013).