

# 移動型投影システムを用いた現地開催バーチャルライブにおける キャラクター接近演出の提案

筒井匠<sup>†1</sup> 栗原渉<sup>†1</sup> 三上浩司<sup>†1</sup>

**概要：** 近年、幕張メッセやインテックス大阪などといった現実空間に存在する会場において、キャラクターが現実空間で歌やダンスを披露する「バーチャルライブ」が数多く存在している。現行の多くの演出はステージ上に固定されたディスプレイを前提としているため、会場内の観客の位置によりキャラクターの見え方や没入感に差が生じ、来場者にとって公平な体験が提供されていないという問題がある。ディスプレイを会場内で移動させることで個々の来場者に近接した視覚体験を与える可能性はあるものの、キャラクターのモーションとディスプレイの物理的移動が適切に同期されていない場合、キャラクターが滑って見えるという技術的課題が生じる。この滑り現象は、意図した没入感を損ない、安全上や演出的な問題を引き起こすため、単にディスプレイを動かすだけでは解決できない複合的な課題である。

関連研究を調査した結果、この特定の問題に直接取り組んだ研究は限られており、既存の試みの多くは目的設定が異なるか、キャラクターのモーションとディスプレイ移動の厳密な同期を考慮していないことが分かった。

そこで本研究では、Unity とペンプロッタを用い、視覚的ずれを生じさせないためのハードウェア設計と制御アルゴリズムの組合せを開発することを目指す。

## 1. はじめに

近年、3DCG で作成されたキャラクターが実空間の会場において歌唱やダンスを行う「バーチャルライブ」が、エンタテインメント表現の一形態として広く普及している。バーチャルライブは、単に映像を鑑賞するコンテンツではなく、観客が同一の空間に集い、音響や照明、映像演出を含めた環境の中で、キャラクターがあたかも「その場に存在している」かのような感覚を共有する体験である点に特徴がある。

このような体験において重要となるのは、キャラクターの存在感や空間的な位置関係の知覚である。キャラクターが会場内のどこに立ち、どの方向へ移動しているのかといった情報は、観客の没入度に大きく影響する。そのため、バーチャルライブではキャラクター映像と実空間との対応関係が適切に設計されていることが求められる。一般的には、ステージ上に透明なディスプレイを設置し、プロジェクターによってキャラクターを投影する方式や、LED ディスプレイにキャラクター映像を表示する方式が用いられている。観客は客席からこれらのディスプレイを鑑賞することで、あたかもキャラクターがステージ上に立ってパフォーマンスを行っているかのような視覚的印象を得る。

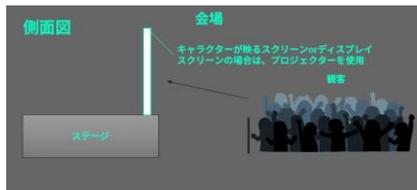


図 1 バーチャルライブの基本的なステージ構成  
しかし、このような構成にはいくつかの課題が存在する。

第一の課題として、キャラクターが観客に近づく演出が物理的に制限される点が挙げられる。ディスプレイが固定されている場合、前方の観客はキャラクターを大きく、かつ細部まで鮮明に確認できる一方で、後方の観客では表示が小さくなり、視認性や没入度に差が生じやすい。また、キャラクターが画面端から端へ移動する演出を行ったとしても、それはあくまで映像内部での位置変化にとどまり、実空間における距離変化としては知覚されにくい。その結果、会場全体で均一に高い没入感を得ることが難しいという問題がある。第二の課題として、これらの制約を解消する手段の一つとして考えられる「移動型ディスプレイ」を用いた演出において、新たな問題が生じる点が挙げられる。ディスプレイそのものを会場内で移動させることで、キャラクターが実際に観客へ近づいているような演出が可能となる一方で、キャラクター映像の動きとディスプレイの物理的な移動が適切に同期していない場合、キャラクターが本来の動作軌跡からずれて見える「滑り現象」が発生する。そのため、キャラクター映像の動きとディスプレイの位置および移動速度を高精度に同期させる制御手法が不可欠である。

そこで本研究では、キャラクター映像の動きとディスプレイの物理的な移動を適切に同期させることで、滑り現象を抑制する移動型投影システムの開発を目的とする。また、投影サイズが小さい場合であっても滑り現象が発生せず、観客がキャラクターの会場内での移動を明確に知覚できることを実証することを目指す。

## 2. 関連研究

本章では、関連研究の調査結果について述べて本研究

<sup>†1</sup> 東京工科大学

との差異および新規性を述べる。

土田ら[1]は、ダンスパフォーマンスの隊形練習において、ディスプレイ自体を自走させることで練習効率を向上させるシステムを提案している。スクリーンに映像を投影し、それを自律移動させることで、ダンサーは固定的な舞台セットを想定する必要なく、実際の隊形や位置関係を確認しながら練習できる。自走スクリーンは位置制御によって所定の場所へ移動可能であり、振付のタイミングや動線確認が容易になる。

Onishi ら[2]は、柔軟な伸縮式パーティションを自律的に移動・展開させることで、必要に応じて安全でプライベートな作業空間を生成するシステムを提案している。パーティションはロボット台車上に設置され、周囲環境や利用者の配置に応じて形状や位置を変えることが可能である。人や障害物を検知しながら移動・展開できる点や、可搬性の高さは本研究の移動型ディスプレイにも応用可能な要素である。

巻口ら[3]は、キャラクターを映すディスプレイを山車に搭載し、舞台演出などにおいて映像の表示位置を柔軟に変更できるシステムを提案している。空中像表示技術により、観客からディスプレイを意識させずに映像を浮かび上がらせることができ、装置を移動させることで演出の自由度が向上する。舞台パフォーマンスへの応用事例も示されており、演出の臨場感向上に寄与している。

以上の関連研究はいずれも、ディスプレイや構造物を移動させることで、練習支援や空間構成、舞台演出の自由度を高める点に主眼を置いている。一方で、移動体に表示されたキャラクター映像の動作と、ディスプレイの物理的な移動との関係性については十分に検討されていない。特に、キャラクターの歩行モーションにおいて、接地動作とディスプレイの移動量・移動速度が一致しない場合に生じる視覚的な不整合、すなわち滑り現象を明確な課題として扱った研究は見当たらない。

そこで本研究は、移動型ディスプレイを用いた表現において、キャラクターの動作と表示位置の同期精度そのものを研究対象とし、滑り現象の発生要因を整理した上で、その抑制を目的とした手法を提案する点に新規性がある。これは、移動型ディスプレイを単なる演出装置として扱うのではなく、キャラクター表現の成立条件として実空間との整合性を明示的に位置づける試みである。

### 3. 提案手法

本章では、第1章で提示した課題と第2章で提示した関連研究を踏まえ、ディスプレイの物理的移動とキャラクターの映像内の移動を精密に同期させる手法を提案する。

#### 3.1 提案手法の概要

本手法では、事前に収録したキャラクターモーションの座標情報と移動量に基づき、滑りが発生しないディスプレ

イの移動先座標を算出する。この座標をもとに、ディスプレイの動きを制御することで、移動ディスプレイ上で歩行やダンスなど移動をしているキャラクターの滑りを解消する。

#### 3.2 精密同期の考え方

提案手法の中核は、バーチャルキャラクターのモーションを、実空間における移動量および速度へ変換する処理にある。

バーチャルキャラクターの移動は、連続的なモーションとして定義されているが、本研究ではこれを一定時間間隔  $\Delta t$  ごとにサンプリングし、離散的な位置系列として扱う。隣接する2時刻におけるキャラクターの位置を比較することで、次の情報を得る。

- 移動方向
- 移動距離
- 移動に要した時間

これらの情報を用いることで、キャラクターの移動を「どの方向に」「どれだけの距離を」「どの速度で」移動したかという形で定量的に表現できる。

#### 3.3 システム要件

本研究で提案する手法を適用するためには、バーチャル空間と実空間の間で、キャラクターの移動情報を対応付け可能なシステム構成が必要となる。

まず、バーチャル空間においては、キャラクターの移動が位置変化として定義されており、アニメーション再生中の位置情報を一定時間間隔で取得可能であることが求められる。これにより、キャラクターの移動距離および移動に要する時間を算出できる。

次に、実空間側には、表示位置を連続的に変化させることが可能な提示手段が存在し、バーチャル空間で算出された移動情報を反映できる必要がある。これにより、キャラクターの移動に応じた表示位置の変化を実空間上で提示できる。

さらに、バーチャル空間で算出された移動情報を実空間側へ逐次反映可能な仕組みが存在し、キャラクターのモーション再生と表示位置の変化が同一の時間軸上で進行することが求められる。

以上の要件を満たすことで、本研究で提案するディスプレイの物理的移動とキャラクターの映像内の移動を精密に同期させる手法が適用可能となる。

#### 3.4 バーチャルライブの定義

最後に、本研究で扱うバーチャルライブの対象範囲を明確にするため、以下の条件を満たすライブ形態を対象とする。

- 開催される空間は、現実空間の会場であること。
- ステージに立つ演者は、3DCG でモデリングされたモデルに事前に収録したモーションを適用しパフォーマンスを行うキャラクターであること。ただ

し、バンドメンバーなど必要に応じて、同一のステージに生身の人間が立つことは可能である。

- 観客は、現実空間の会場に会場し、で鑑賞すること。

## 4. 実装

本章では、第3章で提示した提案手法を実現するために、構築したシステムについて述べる。

### 4.1 システムの全体構成

まず、システムの全体図を図2に、実機の画像を図3に示す。

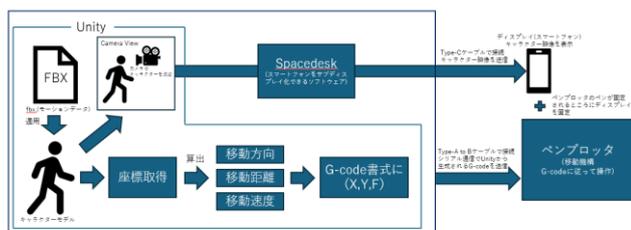


図2 システム全体図



図3 実機の画像

本手法は、キャラクターの移動軌跡をペンプロッタで描画するために、3Dアニメーションの位置情報をG-codeに変換する処理が中心となる。G-codeは、NC工作機械や3Dプリンタなどにおいて位置や動作を制御するための標準的な命令形式である。例えば、G1 X10 Y20 F1500のような命令は、「送り速度(F)1500 mm/minで、X座標10mm、Y座標20mmの位置まで直線的に移動せよ」という意味を持つ。G-code命令は、座標系の移動だけでなく、スピンドル回転や補間方法、リミット制御なども記述可能であるが、本研究では主に2次元平面上でのX-Y軸座標移動指令と、その速度に関する命令(G0, G1, F値)を利用している。この処理において、アニメーションに含まれるキャラクターの動きを正確に抽出し、Unity上での時間軸や座標系に応じて実空間の動作へ変換する必要がある。

### 4.2 使用機材

- Unity 2022.3.22: 仮想空間でキャラクターを動かす及びG-codeデータの生成をするために使用。
- Spacedesk: スマートフォンやタブレットなどをサブディスプレイ化できるアプリケーション。

キャラクターの映像をディスプレイ(スマートフォン)に送信するために使用。

- 3Dモデル「止丸式初音ミクNT」: 歩行モーションの適用およびキャラクター動作の可視化するために使用。
- VIGOTEC VG-A4 XY ペンプロッタ: 実空間でディスプレイを動かすために使用。
- USBケーブル type-A to B: ノートPCとペンプロッタを接続するケーブルとして使用。
- USBケーブル type-C to C: ノートPCからディスプレイに映像信号を送るために使用。
- HP ENVY x360 Convertible 15-ed (ノートPC): Unity及びペンプロッタの制御のために使用。
- Google Pixel 5(スマートフォン): ノートPCから送られてきたキャラクター映像を表示するために使用。
- SELECOCO マイクスタンド用 スマホホルダー: ペンプロッタのペンがもともとついていた部分にディスプレイを固定するために使用。

### 4.3 入力に扱うアニメーション

使用するアニメーションデータは主に次の2つの形式で取得可能である。

- MixamoなどのWebサービスで提供されるアニメーションデータ(通常は.fbxファイル形式)
- Unity Asset Storeで配布されているアニメーションパッケージ(Unity専用の.prefabやcontrollerとして構成)

キャラクターの動きをペンプロッタで再現するには、アニメーションの再生によってキャラクターの座標がどのように変化するかを正確に把握する必要がある。ここで重要になるのが「Rootモーション」の概念である。Rootモーションとは、アニメーションの再生に合わせてキャラクターの基底(通常はHipボーン)が空間的に移動する情報を持つ構造である。その情報がキャラクターの位置、速度を算出するための入力値となる。一方で、Rootモーションが無効なアニメーションでは、キャラクターがその場で動いているように見えても、実際の座標は変化せず、G-codeに出力すべき移動情報が得られないため、モーションを制作、選定する際は注意が必要である。

### 4.4 G-code出力処理の設計(スケーリング・速度補正)

キャラクターの動きに対して、ペンプロッタが追従するには、移動距離と速度をG-codeとして記述する必要がある。そのために、座標の変化から移動距離と速度を求め、それをG-codeの書式に変換する。例えば、ある時刻におけるキャラクターの位置が(x1, y1)、次の時刻に(x2, y2)であるとき、2点間の距離dは次の式で求められる。

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

この距離  $d$  とフレーム間の時間差  $\Delta t$  から、1 分あたりの移動速度 (Feedrate)  $F$  を次の式で計算する。

$$F[\text{mm/min}] = \left(\frac{d}{\Delta t}\right) \times 60 \times S$$

ここで、 $S$  はスケール係数であり、Unity 空間 (1.0 が 1 メートル) とペンプロッタ上の実距離 (ミリメートル) との変換を担う。例えば、キャラクターの身長を 10cm (Unity 単位で 0.1) と見積もり、ペンプロッタ上で約 100mm の高さに相当させる場合、おおよそ 10 から 100 が妥当となる。上記情報を用いて、G1 X... Y... F... 形式の文字列に変換する。出力タイミングは updateInterval (例: 0.1 秒) で制御される。

#### 4.5 ディスプレイ搭載のペンプロッタの構造と制御方法

本手法で用いるペンプロッタは、XY 平面上にスマートフォンを載せて動かす構造を持つ自作デバイスである。各軸はステッピングモーターで駆動され、G-code 命令に従って制御される。USB 経由で PC と接続され、Unity 上の C# スクリプトから直接シリアル通信で命令を送信できるように設計した。これにより、従来必要とされた専用ソフトウェア (例: Candle や bCNC) を介さず、Unity のアニメーション再生とリアルタイムで連動可能となる。

#### 4.6 G-code 生成処理におけるパラメータ設計

本手法では、図 4 のようにキャラクターの動作や表示条件に応じて、ディスプレイの移動挙動を調整可能とするため、Unity の作成したゲームオブジェクトの inspector に G-code 生成処理に複数のパラメータを設けている。



図 4 パラメータ調整画面

G-code 生成処理において使用する主なパラメータを以下に示す。

- Target

モーションを適用したキャラクターモデルのうち、どの Transform を基準として座標を取得するかを指定するパラメータである。本研究では、キャラクターの全体的な移動を代表する部位を基準として設定した。

- Scale

Unity 空間と実空間との距離対応を調整するためのスケール係数であり、キャラクターの大きさや表示距離に応じて任意に設定可能である。

- Port Name

PC と XY プロッタ間のシリアル通信経路を指定するための識別子である。

- Baud Rate

シリアル通信におけるデータ転送速度を示す指標である。

使用した XY プロッタの制御ファームウェアにおける標準設定であり、G-code 命令を安定して送信可能な通信速度である。

- Update Interval

位置情報の取得および G-code 出力の時間間隔を制御するためのパラメータである。値を大きくすることで移動指令の発行頻度を下げ、小さくすることでより細かな追従が可能となる。

- MaxX, MaxY

ペンプロッタの移動範囲の上限値を設定するためのパラメータであり、装置の可動範囲を超えないよう制御するために用いる。

#### 4.7 ユーザー検証

提案システムの有効性を予備的に確認することを目的として、簡単なユーザー検証を行った。

そこで、本研究では提案システムによる演出が、第三者にとっても成立しているかを確認するため、簡単なアンケート調査を行った。本検証は、大規模な評価実験に先立つ探索的な位置づけとし、キャラクター映像と実空間におけるスクリーン移動の同期が視覚的に破綻していないか、およびキャラクターが観客に接近してくる演出として認識されるかを確認することを目的とした。

実験には、本研究で実装した移動型投影システムを用いた。Unity 上でキャラクターを動作させ、その位置情報を基に生成した G-code を用いて XY プロッタを制御し、スマートフォンに表示したキャラクター映像を実空間で移動させた。提示映像にはキャラクターの歩行モーションを用い、映像上の移動とスクリーンの物理的移動が同期するよう設定した。

研究内容を知っている 20 代学生 4 名 (うち男性 3 名女性 1 名) に、システムが動作している様子を動画で視聴してもらい、体験後に簡単なアンケートへの回答を求めた。

アンケート結果から、4 分の 3 が「おおむね大きな滑りは感じられない」と回答しており、キャラクター映像とスクリーン移動の同期は視覚的に大きな破綻を生じていないことが確認された。また、「キャラクターがとてもそこにいるように感じた」とする回答も得られ、他の被験者において同様の傾向が見られた。一方で、「歩き終わりが滑って見える」という自由記述も確認された。また、映像とスクリーン移動の同期についても、主観的評価の範囲では大きな滑り現象は確認されなかった。

## 5. 今後の展望

本研究では、実空間におけるバーチャルライブにおいて問題となる「滑り現象」に着目し、その要因をキャラクター映像の動きと表示媒体の物理的移動との同期精度の不足にあると整理した。その上で、スクリーン自体を会場内で移動させ、キャラクターのモーションとスクリーンの位置・

速度を精密に同期させる手法を提案・実装した点に本研究の特徴がある。

提案手法では、Unity 上で取得したキャラクターの位置情報を基に G-code を生成し、XY ペンプロッタを用いて実空間のスクリーンを制御するシステムを構築した。これにより、滑り現象を回避し、よりキャラクターが実際に会場内を移動しているかのような一体感のある表現がある程度可能となった。ユーザー検証からも、キャラクター映像とスクリーン移動の同期が視覚的印象に与える影響が確認され、まだ完全とは言えないが、提案手法の有効性が示唆された。

今後は、さらなる精密な同期、よりユーザーが使いやすくなるための UI 化やダンスモーションなどの複雑な動きをするモーションへの対応や実際のライブハウスやホールなどといった規模を想定した移動機構の巨大化、照明・音響との統合制御、ディスプレイ部分をよくバーチャルライブで用いられているポリットスクリーンにするなどが課題として挙げられる。本研究は、実空間におけるバーチャルキャラクター表現の新たな可能性を示す一例であり、今後のバーチャルライブ演出や実空間 XR 表現の発展に寄与する基礎的知見を提供するものである。

## 参考文献

- [1]土田 修平,寺田 努,塚本 昌彦 (2015) ダンスパフォーマンスにおける自走型ディスプレイを用いた隊形練習支援,情報処理学会論文誌, 56(5),pp. 1428-1441.
- [2]Yuki Onishi,Kazuki Takashima 高島 和樹,Kazuyuki Fujita,Yoshifumi Kitamura.(2021)Self-actuated Stretchable Partitions for Dynamically Creating Secure Workplaces,ACM CHI Conference on Human Factors in Computing Systems,294,pp .1-6.
- [3]巻口 誉宗,高田 英明,坂本 大介,小野 哲雄 (2019) 両面透過型多層空中像表示技術の提案と実装情報処理学会,8(1),pp .1-10.