第 24 回日本バーチャルリアリティ学会大会論文集（2019 年 9 月）

VGentEditor: 操作部位と空間目標点を動作表現として用いたインタラクティブキャラクタの動作生成

VGentEditor: Interactive character motion synthesis represented by controlling body parts and target space points

佐藤裕仁 [1)]，三武裕玄 [1)]，杉森健 [1)]，長谷川晶一 [1)]
Hirohito SATO, Hironori MITAKE, Ken SUGIMORI, and Shoichi HASEGAWA

1）東京工業大学 (〒 226-8503 神奈川県横浜市緑区長津田町 4259)

概要: キャラクタのインタラクティブな振る舞いには，シーン内のオブジェクトやセンサで取得した人間の位置などに合わせて動作が変化する必要がある．本研究では，動作での操作部位と空間内の目標空間の組を記述することで，動的にモーションを生成する手法を提案する．また，提案手法の編集 GUI を構築し，編集した動作例を紹介する．多様な振る舞いを編集でき，インタラクティブキャラクタの構築を大きく手助けできることを目指す．

キーワード： 動作生成，ステートマシン，ヒューマンエージェントインタラクション

1. はじめに

我々の身の回りには多様なキャラクタが存在している．ゲームやアニメといったコンテンツで，作品世界を彩る重要な要素である．近年では，HMD と付属コントローラにより，仮想空間内に自らが没入するような体験が手軽に行え，キャラクタに至近で対面しているかのような体験も可能である．コントローラにより自身の手や頭の動作が直接入力されるため，より自由なインタラクションが可能である．また，キャラクタを社会的な場でエージェントとして活用する事例もある．

こうしたキャラクタとのインタラクションの中で，キャラクタの動作もこちらの動きや環境に合わせて変化すべきである．そうすることで，人間同士のインタラクションで起こるような相手の振る舞いに応じて自己の振る舞いを変化させる相互フィードバック的な振る舞いの変化がキャラクタと人間でのコミュニケーションでも起こせると期待できる．これは事前に作成した静的なモーションをただ再生するだけでは難しく，動的に変化する動作制御が必要となる．

本研究では，インタラクション中の人間や環境内のオブジェクトの位置関係に応じて動的に変化する動作を記述する手法を提案する．キャラクタ動作の制作者が記述内容を把握しやすく，調整のしやすい手法を目指した．提案する記述手法を用いた動作編集インタフェースを構築し，実動作例を示す．

2. 関連研究

動作生成の手法として，最適化問題に落とし込むという方法がある．キャラクタの関節角度などを変数として，動作において満足させるタスクや消費エネルギーを目的関数に書き下し，動力学的な拘束などを考慮しつつ，動作全体にわたる最適化を行う．出力は計算時に用いたキャラクタの身体制御の時系列である．適用可能範囲が広く，歩行やアクロバット [1]，複数キャラクタの協調動作 [2] など多くの動作に利用されている．しかし，変数の次元が高く，最適化計算に時間がかかる．オフラインでの運用が基本で，変化する環境の中で実時間で動作を得るような場面では使えない．

オンラインで動作をさせるには，環境の変化を許容する必要があるため，単なる動作シークエンスではなく制御器を生成することとなる．手作業で構築する [3]，機械学習を用いるなどの方法で制御器を得られる．手作業で構築する場合，自分で制御の流れを追いやすい．しかしながら，パラメータのチューニングや設計そのものなどが困難である．機械学習では，近年はディープラーニングを利用したものが多くみられる．実人間の動作から取得したモーションデータを利用しリアルな歩行動作を生成する例 [4] や，ごく少数のリファレンス動作をもとに強化学習による歩行，アクロバットを学習させた例 [5] がある．機械学習は，強力な手段ではあるが，学習された動作制御器は人間が理解できるような形式から逸脱しており，可読性はない．また，学習には相応の時間を要する．

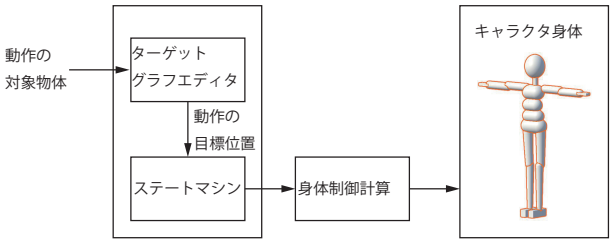


図 1: 提案手法

3. 提案手法

提案手法の概要を図 1 に示す．キャラクタの行動決定 AI から，動作のターゲットとなるオブジェクトを受け取ると，まずターゲットグラフエディタにより，対象オブジェクトに応じて動作の操作部位が向かうべき目標位置を出力する．目標位置はステートマシンに渡され，適切なタイミングで操作部位とその目標位置および動作時間からなる動作指示を出す．動作指示に従い，操作部位の空間内での軌道を計算し，IK により各時間のキャラクタ身体関節の目標関節角度を得ることで，キャラクタ身体を制御する．動作の設計の際には，主にステートマシンとターゲットグラフエディタの二つを編集する．

提案手法の特徴を述べる．提案手法は，キーフレームアニメーションとモーショングラフに対応するような記述方法をとっている．キーフレームアニメーションは，キャラクタのポーズをキーフレームで指定し，その疎な時系列を補間することで動作を得る手法である．モーショングラフは，動作データ間の遷移関係をステートマシンで記述し，動作データを制御する手法である．比べると，動作の対称オブジェクトに応じて変化する目標位置を疎に指定するという点がキーフレームに対応し，時間や条件に応じて目標位置を切り替えるステートマシンを用いるという点がモーショングラフと対応するといえる．キーフレームアニメーション及びモーショングラフはよく利用される方法であり，それに似た構造をとっていることで，記述内容を把握しやすく，調整しやすい動的に変化する動作の記述法となると期待できる．

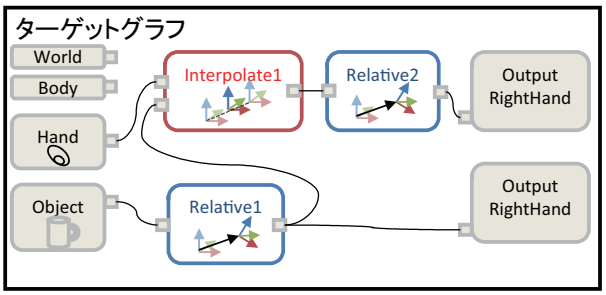
以下ではキャラクタ身体，ステートマシン，ターゲットグラフエディタについて説明する．

3.1 ターゲットグラフエディタ

AI などから渡された動作の目標物体の情報をもとに，動作指示のための目標位置へと変換する役割を担う．目標位置は例えばカップの取っ手のように対象物体のローカル座標で指定されるある点であったり，キャラクタからみて物体の手前の一点であるなど幅広い．こうした幅広い変換に対応しつつ，グラフィカルに編集可能な環境として，座標変換を行うノードからなるグラフを用いることとした．

グラフのノードは大きく分け下記の 3 種類ある．

1. 入力ノード
2. 変換ノード
3. 出力ノード



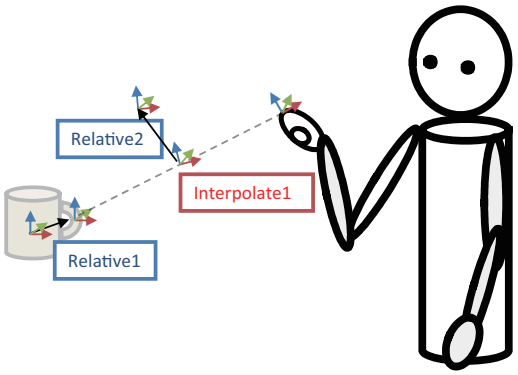


図 2: ターゲットグラフエディタによる目標位置への変換の様子

入力ノードは，対象物体やパラメータを AI 等から受け取るためのノードである．変換ノードは，対象物体やキャラクタの身体部位の位置をもとに，ローカル座標内での位置変換や補間を行うノードである．変換ノードをいくつか経由させ，座標情報を変換していくことで最終的に必要な動作の目標位置へと変換する．図 2 に変換の様子を示す．そうして得られた目標位置の情報が出力ノードからステートマシンの各対応ステートへと渡される．

3.2 ステートマシン

ステートマシンの役割は動作全体の制御である．時間経過により目標位置を切り替えたり，条件を満たすまで同じ動作を繰り返したり，状況に応じて異なる目標位置に向けて動作をさせたりといった制御を行う．各ステートは，動作させる部位とターゲットグラフエディタから得られる目標位置及び目標到達までの目標時間を保持している．ステートに遷移した際，それらの情報を用いてキャラクタ身体に向けて動作指示を出す．ステートからステートへの遷移は，時間経過や特定条件を満たした際に発生する．

3.3 キャラクタ身体制御

物理エンジン上に，キャラクタを模した物理モデルを用意し，シミュレーションにより動作を生成する．キャラクタの身体の物理モデルとして多関節剛体モデルを用いる．関節は，球関節とヒンジ関節を用いる．各関節は，目標角度及び角速度を設定することで PD 制御により駆動させる．

動作指示は時間的に疎であるため，各時刻における操作部位の目標位置を得るために補間を行う必要がある．この補間は躍度最小軌道 [6] により行う．これは現在の制御部位

図 3: 手を振る動作の高さ変化

図 4: 追従による手合わせ動作

位置と目標位置，動作時間から下記の式により導出される軌道である．

$$p(t) = p_s + \left(p_s - p_{t_f}\right)\left(15\tau^4 - 6\tau^5 - 10\tau^3\right) \qquad (1)$$

なお，$\tau = \frac{t}{t_f}$ は運動時間で正規化した時間 $(0 \leq \tau \leq 1)$ である．軌道の導出段階では状態空間モデルなどを必要とせず，緩やかな加減速を実現するモデルであることから，ロボット制御や動作計算の初期解で利用もされる軌道モデルである．

各時刻の操作部位の目標位置が得られたところで，IK により関節の目標角度へと変換する．IK にはヤコビ行列 IK を用いる．また，各関節の動かす程度の重みを設定できるほか，零空間内を基準姿勢に近い値が得られるように探索している．

4. 編集環境

以上の提案手法を実用化するための編集インタフェースを実装した．開発は Unity のエディタ拡張として行い，使用言語は C#である．キャラクタ身体の物理シミュレーションには，C++で構築された物理エンジン Springhead[7] を用いた．

5. 動作の実装例

提案手法を用いて実装した動作の例を示す．

5.1 ジェスチャ

相手に対して手を振る動作であれば，高く振るときと低く振るときの指示を補間することでキャラクタの気分といったパラメータ等を参照して振る高さをシームレスに変えることができる．相手のいる位置に合わせた方向の変換も空間内で行うことができる．

5.2 追従

キャラクタと手を合わせるときや物の受け渡すときなどは，人間の手にキャラクタの手を追従させる．一定間隔で目標位置を更新しつつ，ステートマシンをループさせることで実現できる．更新間隔の調整により，気づくのが遅れて追従が遅れるなどの動作ができる．

6. 結論と今後

動作における重点操作部位とその目標位置を記述することで動作を生成する手法を提案した．

今後は，実用に向けた編集環境の整備を行うとともに，実際にキャラクタエージェントに向けた各種動作の実装を本格的に行う予定である．また，躍度最小軌道に代わる補間方式や拘束条件の変更にロバストな逆動力学解法の検討なども行う．

参考文献

[1] Mazen Al Borno, Martin de Lasa, Arron Hertzmann : Trajectory Optimization for Full-Body Movements with Complex Contacts, IEEE Transactions on Visualization and Computer Graphics, Vol. 19, No. 8, pp. 1405–1414, 2013.

[2] Igor Mordatch, Enanuel Todorov, Zoran Popović : Discovery of Complex Behaviors through Contact-Invariant Optimization, ACM Transactions on Graphics, Vol. 31, No. 4, Article 43, 2012.

[3] Wenjia Huang, Mubbasir Kapadia, Demetri Terzopoulos : Full-Body Hybrid Motor Control for Reaching, Motion in Games. MIG 2010. Lecture Notes in Computer Science, Vol. 6459, pp. 36–47

[4] Daniel Holden, Taku Komura, Jun Saito : Phase-functioned Neural Networks for Character Control,

ACM Transactions on Graphics, Vol. 36, No. 4, Article 42, 2017.

[5] Xue Bin Peng, Pieter Abbeel, Sergey Levine, Michiel van de Panne : DeepMimic: example-guided deep reinforcement learning of physics-based character skills，Phase-functioned Neural Networks for Character Control，ACM Transactions on Graphics, Vol. 37, No. 4, Article 143, 2018.

[6] Tamar Flash, Neville Hogan : The Coordination of Arm Movements: An Experimentally Confirmed Mathematical Model，The Journal of Neuroscience，Vol. 5, No. 7, pp. 1688–1703, 1985.

[7] 長谷川晶一，三武裕玄，田崎勇一 : 動作行動開発のための物理エンジン Springhead，日本ロボット学会誌，Vol. 30, No. 9, pp. 841–848, 2012