



対話的なキャラクターエージェントのための 動作データの到達目標列への自動変換

Automatic Conversion of Movement Data into Target Sequences for Interactive Character Agents

上野瑞生¹⁾, 三武裕玄²⁾, 佐藤裕仁¹⁾, 杉森健¹⁾, 長谷川晶一¹⁾

Mizuki UENO, Hironori MITAKE, Hirohito SATO, Ken SUGIMORI, and Shoichi HASEGAWA

1) 東京工業大学 工学院 情報通信系 (〒 226-8503 神奈川県横浜市緑区長津田町 4259 R2-20, order@haselab.net)

2) 明治大学 先端メディアサイエンス学科 (〒 164-8525 東京都中野区中野 4-21-1 1010 室, mitake@meiji.ac.jp)

概要: VR などの普及により対話相手や環境に調和するアニメーションの需要が増加している. そのようなアニメーションの表現方法の一つである到達目標列を用いた手法では, 既存の動作データやモーションキャプチャデータを活用することができず, 手作業で到達目標を設定していく必要がある. 本研究では, 既存の動作データやモーションキャプチャデータから到達目標列を自動抽出するアプリケーションを作成した. 実際に動作データを本手法で到達目標列へ変換したところ, 通常のキーフレームアニメーションのキーフレーム数と同程度の到達目標数でほぼ元と変わらないモーションを再現することができた.

キーワード: プロシージャルアニメーション, 動作制作支援, 躍度最小軌道

1. はじめに

バーチャルリアリティの普及やリアルなキャラクターを操作することができるゲームの登場により, ユーザーの動きや周囲の環境によって動作を変え, 自然な身体的インタラクションを行えるキャラクターエージェントの需要が増加している.

このような身体的インタラクションを行うためのアニメーション形式として, 例えば複数のアニメーションをブレンドするといった手法がある. この手法では複数のモーションをあらかじめ用意し, それぞれのモーションが行ったインタラクションの位置を設定し, 実際にインタラクションを行いたいターゲットの位置によってどのモーションをどの比率でブレンドするかを自動で算出し, モーション同士の内分点を出力することでインタラクションを実現している. しかしこの手法ではあらかじめ用意したモーション群の間の状態で表現できるモーションしか出力することができず, また, これに使用したモーションの特徴的な部分 (例えばハイタッチをする動作ならば手を前に突き出す部分) のタイミングを完全に合わせる必要がある.

一方で佐藤ら [2] の研究では, 躍度最小軌道を重ね合わせることで身体的インタラクションを実現させるアニメーション作成ツールを提案している. 躍度とは速度を時間で二階微分したものであり, 躍度最小軌道とはすなわち加速度の変化が最小となるような直線軌道のことである. この軌道は直線軌道ながら滑らかな速度変化を持ち, 複数の軌道を重ね合わせることで曲線を表すことができる. また, 一つの軌道を変更しても軌道の重ね合わせにより変更した周辺のモーションも滑らかに補間するため, 動作対象の変化に

よるモーションの変化が自然になりやすい特徴を持ち, 対話的動作の作成に優れている.

到達軌道の重ね合わせによるアニメーションでは各到達軌道の終点を必ずしも通る必要がないため, キーフレーム間を補間してモーションを作成する従来の一般的なアニメーションやモーションキャプチャデータを直接使用することができず, 再現したいアニメーションの特徴を自分で想像し, 手動で到達目標の位置や回転・影響範囲といったパラメータを直接入力する必要があった.

そこで本手法では入力された動作データを躍度最小軌道の重ね合わせで再現し, かつ到達目標の操作がしやすい軌道群を推定するための分割手法を提案する.

2. 先行・関連研究

Yeo らの研究 [1] では, 人の捕球動作が躍度最小軌道を重ね合わせることで近似再現できることを示し, 自然な捕球動作の作成に成功した. この研究における各躍度最小軌道の到達目標は動作対象となるボールの到達予測位置であり, 到達予測が更新されるたびに新たな躍度最小軌道の到達目標が追加され, 軌道同士を重ね合わせることで随時アニメーションを変化させることでインタラクティブな捕球動作を作成している.

佐藤らの研究 [2] では, 躍度最小軌道を重ね合わせた到達軌道が持つ到達目標の変化に滑らかな補間ができるといった性質に注目し, これを用いることで手先だけでなく足や腰, 頭などの位置や回転までも表現するアニメーション手法を提案した. このアニメーション手法では従来のアニメーションをブレンドする手法などでは表現しきれない動作対象の

大きな変化にも対応したアニメーションを作成することができ、より環境と協調したアニメーションが行える。

動作データを躍度最小軌道の到達目標列へ変換する研究として、清水らの研究 [3] があげられる。この研究では、動作データの持つ速度情報から接線方向の速度 $v_t = \sqrt{v_x^2 + v_y^2 + v_z^2}$ を計算し、速度の躍度最小軌道の重ね合わせによるモーションの速度データとフィッティングをさせることで到達目標列の持つタイミング情報を取得できることを示している。位置や速度の情報は x, y, z の 3 次元であるのに対して、接線方向の速度は 1 次元の情報となるため、タイミング情報を取得する際に変化させるパラメータの数を減少させることができる。

3. 前提知識

3.1 表記

本稿では次の記号と用語を用いる。

- ・データ : 座標や回転角を時系列順に記録したもの
- ・到達目標 : ヒトが意識する運動の経由点や終了地点
- ・姿勢 : 位置や回転情報をまとめたもの
- ・ x_0 : 躍度最小軌道の開始姿勢
- ・ x_f : 躍度最小軌道の終了姿勢
- ・ t_0 : 躍度最小軌道の開始時間
- ・ t_f : 躍度最小軌道の終了時間
- ・ x_{timing} : x_f を一次元に落とし込んだもの

3.2 躍度最小軌道

ヒトの手先の運動軌跡をよく表すモデルとして、Flash ら [4] は躍度最小モデルを提唱した。このモデルでは手先の点間運動は直線のベル型の速度プロファイルを持っており、加速度の時間微分である躍度の二乗を点間運動全体で足し合わせたものが最小になる、つまり手にかかる力の変化が最小になるように設計されたモデルである。

このモデルは、到達運動の開始時刻と終了時刻をそれぞれ t_s, t_f 、手先の初期位置と終了位置を x_s, x_f とした 4 つのパラメータを持ち、軌道の初期速度・初期加速度・終了速度・終了加速度を 0 とした場合、到達運動は式 (2) で表される。

$$\tau = \begin{cases} 0 & (t \leq t_s) \\ \frac{t - t_s}{t_f - t_s} & (t_s < t < t_f) \\ 1 & (t_f \leq t) \end{cases} \quad (1)$$

$$MJ(\tau) = x_s + (x_f - x_s)(6\tau^5 - 15\tau^4 + 10\tau^3) \quad (2)$$

躍度最小軌道の重ね合わせでは相対座標を用いるため $x_s = 0$ となる。また、計算の過程で x_f にかかる時間に依存する項を使用するため、次の式 (3) を用意する。

$$mj(\tau) = 6\tau^5 - 15\tau^4 + 10\tau^3 \quad (3)$$

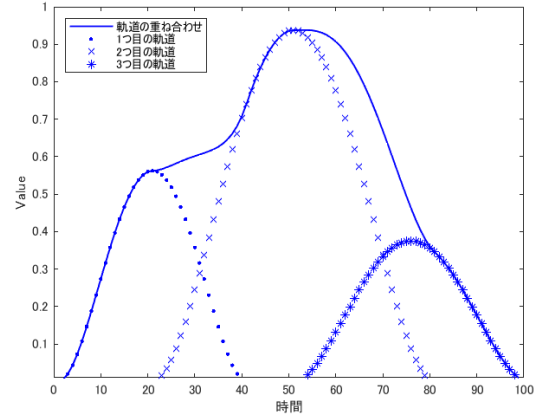


図 1: 個々の躍度最小軌道 (点線) とそれらを重ね合わせた軌道 (実線) の一例

3.3 回転角に関する躍度最小軌道

躍度最小モデルは手先位置を表すために提唱されたが、式 (3) を到達目標までの変化率ととらえると、クォータニオンで表された回転角に関する補間を球面線型補間を用いて式 (4) で表すことができる。

$$MJ_{Rotation}(\tau) = \text{Slerp}(mj(\tau), x_f) \quad (4)$$

ここで $\text{Slerp}(r, q)$ は単位クォータニオン (無回転) からクォータニオン p への変化率 r での球面線型補間を表す。

3.4 躍度最小軌道の重ね合わせ

本稿で扱う躍度最小軌道の重ね合わせは、Yeo らの研究で示された位置に関する軌道の重ね合わせと、佐藤らが開発したアプリケーションで実装されている回転に関する軌道の重ね合わせの二種類がある。これらの重ね合わせは以下の式 (5)、式 (6) で表される。

$$Position(t) = \sum MJ(\tau) \quad (5)$$

$$Rotation(t) = \prod \text{Slerp}(mj(\tau), x_f) \quad (6)$$

図 1 は式 (5) へ式 (2) を代入した一例である。図中の点線は個々の躍度最小軌道の速度波形であり、実線がこれらを重ね合わせた軌道の速度波形となる。

4. 提案手法

4.1 軌道分割の最小化問題への定式化

本研究で行う軌道分割は、システムへ入力された動作データを *Input*、システムが出力した動作データを *Output* としたとき、以下のような最小化問題を解くことで行う。

$$\arg \min_{Output(t)} E(Input(t), Output(t)) \quad (7)$$

同時に推定するパラメータ数が増えると計算量が指数的に増加するため、清水らの研究 [3] で行われたタイミング推定を用い、タイミングの推定 (t_s, t_f, x_{timing} の 3 パラメータ) と、それを用いた位置や回転角の推定 ($t_s, t_f,$

$x_f = (x, y, z)$ の 5 パラメータ) へ分け、初期解としてのタイミング情報 t_s, t_f をある程度確定している状態での推定にすることで高速化を図っている。タイミング推定の際の x_{timing} は、位置の推定では入力データの接線方向の速度の積分値、回転角の推定では入力データのクォータニオンの実成分を用いている。

清水らの研究 [3] では回転角に関してのタイミング推定は行っていないが、本研究ではタイミング情報の初期解として使用した。また、回転の合成にはクォータニオンを用いるため、本来 4 つのパラメータが必要だが、自由度の関係上 3 次元の回転ベクトルで代用できるため、合成の際には回転ベクトルから変換した値を使用している。

また、本研究で扱う最小化問題は多数の極小解を持った多峰性関数のため、最急降下法等の単純なアルゴリズムでは評価関数の値がある程度で収束してしまい、入力したデータとはまったく異なる動作を出力することがある。そこで本研究では乱択アルゴリズムの一種である CMA-ES (共分散行列適用進化戦略) [5] を用いた。このアルゴリズムは多変量正規分布から標本群を生成し、それぞれの標本に対して評価値を計算し、より評価関数の値が小さくなるようにその正規分布の中心や共分散を変更する進化戦略であり、評価関数の微分などを必要としないパラメータ最適化手法である。この手法は前世代の最良のパラメータ解の周辺に標本群をランダムに生成するため、評価関数の局所的最適解に陥りづらくなっている。

4.2 評価関数

パラメータ最適化の際に使用する評価関数は、最小二乗法を元に、到達目標列が作る軌道が人の手先の軌道として自然な軌道になるような指標を付け加えたものとした。

位置に関する具体的な指標は次の式のとおりである。

$$\alpha_1 = \exp(1 - \prod \cos(\theta_{adj})) \quad (8)$$

$$\alpha_2 = \exp(\sum (\frac{|x_f| - \text{Max}(|\text{Input}|/2)}{\text{Max}(|\text{Input}|)})^2) \quad (9)$$

$$\alpha_3 = \exp(\sum (t_{out})) \quad (10)$$

$$E = \alpha_1 \alpha_2 \alpha_3 \sum_t (\text{Output}(t) - \text{Input}(t))^2 \quad (11)$$

α_1 は到達目標列を滑らかにするための指標である。式中の θ_{adj} は隣り合った軌道の成す角であり、隣り合った軌道とは、躍度最小軌道群を t_s でソートしたときに隣り合う軌道である。これの成す角が大きい場合には到達目標がバラバラな位置にあることを示しており、個々の到達目標が人間の意図したものであることを考えると不自然なものであり、また、そのような軌道は後に編集しづらい。そのため、隣り合った軌道の成す角が大きい場合には推定に不利になるような指標となっている。

α_2 は x_f の値が大きくなりすぎないように制限するための指標であり、余計な数値を検証しないことによる推定の高速化と編集しづらい到達目標の推定を回避することを目的としている。

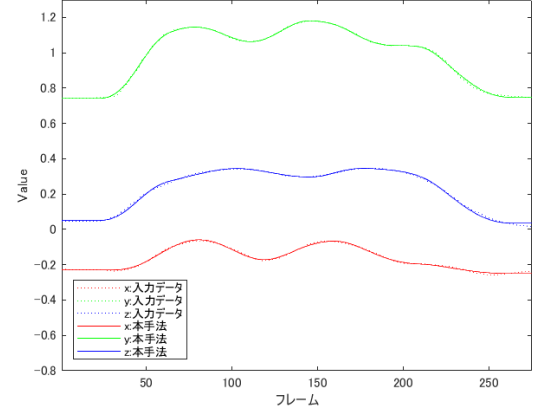


図 2: 入力データ (点線) と本手法 (実線) の出力データの位置情報の比較。

α_3 は動作範囲外にタイミング情報がずれないようにするためのものである。 t_{out} は入力された動作の長さの外側に値を持ってしまう t_s, t_f のことであり、動作範囲外の過剰分を評価関数に乗算するものとなっている。これにより、動作開始前や終了後に残る余分なモーションデータを発生させないようにしている。

回転角に関する指標は次の式のとおりである。

$$\theta = 2 \arccos(|\phi_{adj}|) \quad (12)$$

$$\beta_1 = \exp(\sum \text{if}(\theta > \text{threshold}) : \theta \text{ else } : 0) \quad (13)$$

$$\beta_2 = \exp(\sum (t_{out})) \quad (14)$$

$$E = \beta_1 \beta_2 \sum_t (\text{Output}(t) * \text{Input}(t)^{-1})^2 \quad (15)$$

β_1 は大きな回転を抑制するための指標であり、式中の ϕ_{adj} は隣り合った軌道の x_f の差分である。大きすぎる回転角を球面線型補間で補間する際、単精度浮動小数点型と倍精度浮動小数点型を変換した際に起きるほんの少しの変化でさえも補間軌道が大きく変わってしまうため、推定の際のパラメータ変化やその後編集する際にも扱いづらいものになってしまう。また、手先などを一回転させるような動作は日常ではほぼ起こり得ず、好ましくない軌道となるため、本手法では閾値として $\frac{15}{8}\pi$ 以上の回転があった時に評価関数を大きくするようにしている。

β_2 は α_3 と同じ指標であり、タイミング情報を制限する役目がある。

また、乱択アルゴリズムのため動作の開始時間 t_s が動作の終了時間 t_f を上回る場合などが考えられ、その場合には正しい到達目標の重ね合わせができなくなるので、起きた場合には t_s と t_f を交換するなどの処理を行っている。

5. 評価

本手法を適用したプログラムへ右手先の座標と回転角を時系列順に記入した csv ファイル (モーションキャプチャデータ) を入力し、実際に到達目標列へと変換をした。入力対象のモーションキャプチャデータは目の前の物体を右

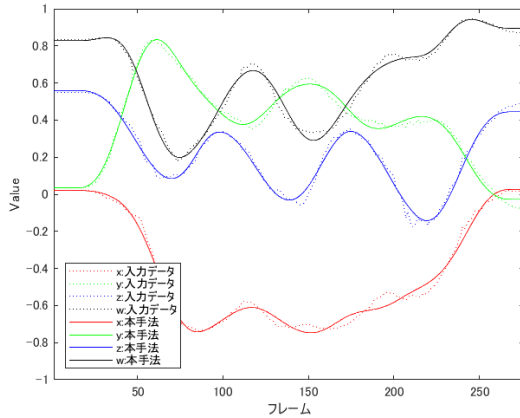


図 3: 入力データ (点線) と本手法 (実線) の出力データの回転角情報の比較

手で撫でるといったモーションであり、一秒間に記録するデータは 50 個、全長は 5 秒 (250 フレーム) 程度のデータである。結果は位置に関しては図 2, 回転角に関しては図 3 の通りとなった。

このグラフはそれぞれ位置・回転に関する比較となっており、位置に関しては 3 次元、回転に関してはクォータニオンのでの比較となっている。グラフで見ると、ほぼ元データ通りの軌道を再現できていると思われる。細かい部分の差異はあるが、モーションキャプチャの測定誤差や生命活動によるノイズと考えることができる。

入力データと本手法で到達目標列に変換されたアニメーションを比較するための動画ファイルのリンクを附録に置く。動画中の左の動作が入力データ、右側の動作が本手法で推定された到達目標列を用いたアニメーションである。

6. まとめ

本研究では、CMA-ES を用いたパラメータ推定を行うことで、動作データを滑らかな到達目標の変化を持つ編集しやすい到達目標列へ変換し、同じような軌跡を持つ動作を再現することができた。

また、躍度最小軌道の重ね合わせで再現することで動作データが滑らかになるため、動作の特徴を失うことが心配されるが、動作の一部に細かい動きが集中しているような動作でない限り、軌道数が十分であれば元の動作を再現できた。

6.1 課題と展望

今後の課題としては、まず位置の推定に比べて回転角の推定が若干不安定なことである。理由として考えられるのは、位置の推定よりも回転の推定に利用した指標が少ないことである。回転角は位置と違いループするものであるため、式 (9) のように回転角を絶対座標でのとり得る範囲を制限をしていない。そのため、元動作の回転角データに対して大きな回転角を推定してしまうことがある。また、式 (8) のように隣り合った軌道の到達目標位置 x_f を結んだベクトル同士が成す角を、回転角では考えることができないため、滑らかな到達目標の変化という指標を回転角の推定に導入できていない。このような部分が回転角の推定を不安定なものにしていると思われる。

また、本研究では推定の際に軌道数 n 個の場合での最適な軌道を計算し、次は軌道数 $n + 1$ 個、その次は $n + 2$ 個と増やしながら評価値を計算し、評価値が収束したら出力するといったプログラムになっている。もしも最初から軌道数がわかっていた場合、それ以外の軌道数での推定をする必要がない。そのため、入力データから軌道数を正確に予測できる手法があれば計算量が減ることとなるため、そのための手法を考えたい。

付録 比較動画 (左側が入力、右側が出力)

参考文献

- [1] Sang Hoon Yeo and Martin Lesmana and Debang R. Neog and Dinesh K. Pai: Eyecatch: Simulating Visuo-motor Coordination for Object Interception, ACM TOG 31(4), 2012.
- [2] 佐藤 裕仁, 三武 裕玄, 杉森 健, 長谷川 晶一: 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, 第 24 回日本バーチャルリアリティ学会大会論文集, 4D-07, 2019.
- [3] 清水 ありさ, 三武 裕玄, 長谷川 晶一: 意図の理解と行動の再現のための上肢到達運動の分析, 第 20 回日本バーチャルリアリティ学会大会論文集, 2015.
- [4] Flash, T and Hogan, N: The coordination of arm movements: an experimentally confirmed, mathematical model, 10.1523/JNEUROSCI.05-07-01688.1985, 1985.
- [5] N, Hansen and A, Ostermeier: Adapting arbitrary normal mutation distributions in evolution strategies: The covariance matrix adaptation, 10.1109/ICEC.1996.542381 1996.