

インタラクティブキャラクタにおける 手先の追いかけて動作の再現

清水 ありさ^{1,a)} 三武 裕玄^{1,b)} 長谷川 晶一^{1,c)}

概要: 人間とキャラクタが協調して行動できるためには、キャラクタが意図を感じさせる動作を行う必要がある。キャラクタの意図表現としては視線・表情・ジェスチャ等の生成が知られるが、人間は相手の手先到達動作から到達予定位置を推定し、意図を読み取る手がかりとすることもできる。

本研究では目標刺激の時系列に対する人間の上肢到達動作の再現を目的とする。まず人間が目標刺激を手で追いかける動作を記録し、躍度最小軌道をフィッティングすることで到達目標位置の特徴（経由点・位置の誤差・反応遅れ時間等）を抽出する。次にこれらの特徴を用いて到達運動の状態遷移モデルを構築する。さらに動作生成システムを実現し、人間の追いかけて動作の特徴を再現する動作が生成されることを確かめた。

Reproducing hand chasing trajectory for interactive character

SHIMIZU ARISA^{1,a)} MITAKE HIRONORI^{1,b)} HASEGAWA SHOICHI^{1,c)}

Abstract:

To realize cooperative work with characters, they have to express their intention by their motion. Gaze, facial expression and gestures are known for expression but human also can presume other's intention by estimating reaching point from observed hand motion.

Our goal is to reproduce human hand motion for virtual characters reacting on time sequences of position stimuli. We recorded hand chasing motion and fitted minimum jerk trajectory to extract characteristics of objective points. We made state transition model for hand motion. And then, we made hand motion reproduce system. Finally, we confirmed similarity between reproduced and original motion.

1. はじめに

ロボットやCGエージェントなどの人工物が、生活の中で人を支え、親しまれる存在となる事が期待されている。こうした場面において人間とエージェントが協調して行動できるためには、エージェントが人間の意図を理解し、同時にエージェントの意図を動作や表情で人間に分かりやすく表出する事が求められる。

そのような協調作業を実現するAIを手作りで作成することには自然さに限界がある。そこで人間同士の協調作業

における動作の実例から協調作業をモデル化する状態遷移機械やHMM等を構築し、協調作業を実現するAIの作成に役立てることを考える。特に協調作業中の手の動作からは、操作や指示の対象という、作業者の意図を示す情報が得られると考えられる。

本研究では、そのような協調作業中の手先の動作を実現するために、人間が目標刺激を手で追いかける動作を記録し、躍度最小モデルに基づいたフィッティングを行うことでその動作の特徴を抽出し、この特徴を用いて状態遷移モデルを構築する。さらに、このモデルに基づいて手先の到達運動を行うCGキャラクタの動作を生成し、人間の手先の追いかけて動作の再現を行うことを目的とする。

¹ 東京工業大学
Tokyo Institute of Technology, 4259 Nagatsuta-cho Midori-ku Yokohama-si Kanagawa 226-8503, Japan

a) ashimizu@haselab.net

b) mitake@haselab.net

c) hase@haselab.net

2. 先行研究

投げられたボールをキャッチするときの手先の運動や視線を解析し、人型のキャラクタにより再現した研究として Yeo らの研究 [1] が挙げられる。この研究ではボールをキャッチするときのような素早い動きを再現するために手先の到達運動を躍度最小モデルに基づき解析を行い、再現を行っているが、人の意図を解析・再現するという観点では行っていない。また、Wada ら [2] は躍度最小モデルを文字を書くときの動作に適用することで、書字動作が少数の経由点で説明できることを示した。この研究でも躍度最小モデルを用いているが、到達運動ではなく文字を書く動作に適用している。草野ら [5] は、指差し動作に躍度最小モデルを適用することで指差しが止まる位置の予測を行ったが、複数の躍度最小軌道が足しあわされる軌道については検討を行っていない。また、速度のベル型波形の加算でヒューマノイドロボットのアームの関節角を指定し、軌道を生成した研究としては Williamson ら [6] の研究がある。

本研究では、手先の追いかかけ動作の分析・再現を行うことで、協調作業中の作業者の操作や意図を反映した動作の生成を目指す。

3. 経由点の抽出

提案手法では、手先座標の連続な時系列データを入力とし、到達目標位置の列を出力する。まず、手先運動の接線方向の速度の時系列データを、開始・終了時刻と最大速度の異なる複数の躍度最小到達運動の和であると仮定してフィッティングを行い、連続な手先運動を複数の到達運動に分解する。次に得られた各到達運動の開始時刻・終了時刻を用いて躍度最小軌道をフィッティングし、各到達運動の到達目標位置を算出する。

さらに、本計測における手先運動を以上のように分析した結果を用い、モデルの構築を行う。

3.1 躍度最小モデル

手先の到達運動を説明するモデルとして、Flash[3] らは躍度最小モデルを提案した。

$$C_J = \frac{1}{2} \int_0^{t_f} \left(\frac{d^3x}{dt^3} + \frac{d^3y}{dt^3} + \frac{d^3z}{dt^3} \right)^2 dt \quad (1)$$

(1) 式が最小になるように軌道が計画されるモデルが躍度最小モデルである。到達位置の座標は時間の 5 次多項式で表され、2 点間の運動において、始点と終点では速度、加速度が 0 として多項式に現れるパラメータを決めると、軌道は次式で表される。

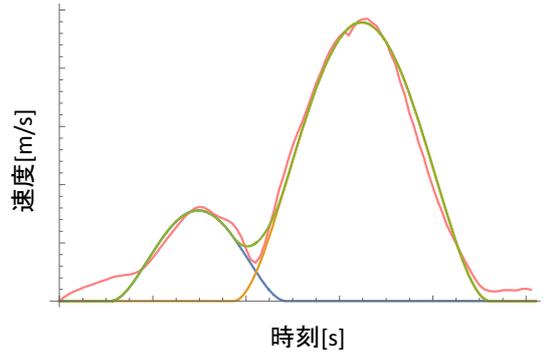


図 1 到達途中で目標が切り替わった場合の到達速度波形の例（ピンクの線：モーションデータ、その他の線：単一のベル型速度波形）

$$\begin{aligned} x(t) &= x_0 + (x_f - x_0)(15\tau^4 - 6\tau^5 - 10\tau^3) \\ y(t) &= y_0 + (y_f - y_0)(15\tau^4 - 6\tau^5 - 10\tau^3) \\ z(t) &= z_0 + (z_f - z_0)(15\tau^4 - 6\tau^5 - 10\tau^3) \end{aligned} \quad (2)$$

ただし、 x_0, y_0, z_0 は各方向における運動の初期位置、 x_f, y_f, z_f は終端位置を表し、 $t_0 \leq t \leq t_f, \tau = \frac{t-t_0}{t_f-t_0}, 0 \leq \tau \leq 1$ である。式を時刻 t の 1 階微分で表すとで速度については次式で与えられる。

$$v_x(t) = (x_f - x_0) \left\{ 30 \frac{(t-t_0)^4}{(t_f-t_0)^5} - 60 \frac{(t-t_0)^3}{(t_f-t_0)^4} + 30 \frac{(t-t_0)^2}{(t_f-t_0)^3} \right\} \quad (3)$$

(y, z についても同様)

なお、(3) 式をプロットするとベル型の波形で表される。

3.2 躍度最小モデルを適用した手先のモーションデータのフィッティング

人の手先到達運動の速度波形は、単一または複数のベル型速度波形の線形和で表されることが知られている (図 1)。ある目標座標に向けて到達運動を行っている最中に目標地点が切り替わった場合、新たな目標地点への到達運動が開始され、速度は当初の到達運動と新たな到達運動の和となる。

このことから、手先動作データの速度を、複数のベル型速度波形の線形和であると仮定し、その個数と重み係数を決定することで、手先動作データから到達運動の時系列を推定することが可能である。

具体的な方法を次に示す。モーションデータから求められる手先の接線方向の速度を $v_{\text{tangential}}(t)$ 、フィッティング対象の関数を $v_{\text{synthesized}}(t)$ 、両者の二乗誤差を E とし、

$$v_{\text{tangential}} = \sqrt{v_x^2 + v_y^2 + v_z^2} \quad (4)$$

$$v_{synthesized}(t) = \sum_{n=0}^{n_b} A_n \left\{ 30 \frac{(t-t_{n0})^4}{(t_{nf}-t_{n0})^5} - 60 \frac{(t-t_{n0})^3}{(t_{nf}-t_{n0})^4} + 30 \frac{(t-t_{n0})^2}{(t_{nf}-t_{n0})^3} \right\} \quad (5)$$

$$E = \sum_{t=t_0}^{t_f} \{v_{tangential}(t) - v_{synthesized}(t)\}^2 \quad (6)$$

と定義する. このとき, フィッティングは, 二乗誤差 E を最小化するような n_b, A_n, t_{n0}, t_{nf} を求めることによって行った.

また, 得られる経由点の数が多くなり過ぎないようにするため, フィッティング誤差がある閾値以下となった時点で経由点数 n_b の増加を打ち切るものとした. 打ち切りの判定は, 二乗誤差 E を運動に要した時間及び速度の最大値で割ることで正規化した $E_{normalized}$ (式 (7)) によって行った.

$$E_{normalized} = \frac{E}{(t_f - t_0) \max(v_{tangential})} \quad (7)$$

以上の方法により求めたパラメータを用い, 到達予想位置を求める. 手先位置のモーションデータを $x(t)$, フィッティング対象の関数を $x_{synthesized}(t)$, 両者の二乗誤差を E_x とし,

$$x_{synthesized}(t) = \sum_{n=0}^{n_b} A_n \left[x_{n0} + (x_{n0} - x_{nf}) \left\{ 15 \left(\frac{t}{t_f} \right)^4 - 6 \left(\frac{t}{t_f} \right)^5 - 10 \left(\frac{t}{t_f} \right)^3 \right\} \right] \quad (8)$$

(y, z についても同様)

$$E_x = \sum_{t=t_0}^{t_f} \{x(t) - x_{synthesized}(t)\}^2 \quad (9)$$

(y, z についても同様)

と定義する. このとき, フィッティングは, 二乗誤差 E_x を最小化するような $x_{n0}, x_{nf}, y_{n0}, y_{nf}, z_{n0}, z_{nf}$ を求めることによって行った.

3.3 計測

本分析・再現の対象の計測として, 40 インチのモニター画面 (BRAVIA KDL-10V2500, SONY) に表示した円形の図形を 0.1~1.6 秒のランダムな間隔で, ランダムな座標に移動させ, この図形の中心の点を利き手の人差し指で追いかける時の手先の運動をモーションキャプチャした. (図 2)



図 2 計測の様子

3.4 結果

3.2 節で述べた手法により 3.3 節の計測における手先の到達運動の分析を行った. 図 3 にモーションデータの接線方向における速度波形と本手法により行ったフィッティング結果を重ねて表示したグラフの一例を示す. ピンクの線が実際のモーションデータの速度波形を示し, その他の線はフィッティングにより求めた速度波形を表す. モーションデータの速度波形が複雑であっても, 本手法を適用することにより複数のベル型波形に分解できることが確認できた.

図 4 上にモニターに表示した到達目標の図形の位置 (T とする) とモーションデータの手先の軌道 (灰色の線), フィッティングにより再現した軌道 (ピンク色の線), フィッティングの結果得られた到達予想位置 (F とする) を xz 平面に表した図を示す. T, F に割り振った数字は到達した順番を示す. また, 図 4 下に T1 を時刻 0 としたときの各 T が表示された時刻, 分解した到達運動が最後まで完了する場合に手先が各 F を通過したと推測される時刻を示す. 到達目標である T に到達完了する場合には, T と F の位置が一致すると考えられるが, 実際には T と F の位置が一致しないことも多くあり, これはモーションデータ取得の際に, モーションキャプチャの座標系がモニター画面の座標系と完全には一致しなかったことや, フィッティングの誤差が原因であると考えられる.

本分析により, この計測における被験者の手先の各時刻における到達目標位置 (図 4 では F1~F5) という点列が得られたことがわかる. このような点列をもとにすると, 本計測における手先の到達運動には以下のような特徴があることが確認できた.

- (1) モニター上に表示された図形の位置 A に直線的に, 正確に手を到達させる.
- (2) モニター上に表示された図形の位置 B に向けて直線的に手を到達させようとしたが, 図形の位置 B とは少し離れた位置 C に到達完了し, 図形の位置 B に向けて微調整するように再び到達運動を行う.
- (3) モニター上に表示された図形の位置 D に向けて到達運動を開始したものの, 図形の位置 D が到達の途中で位置 E に切り替わったために, 手先の軌道が修正され位置

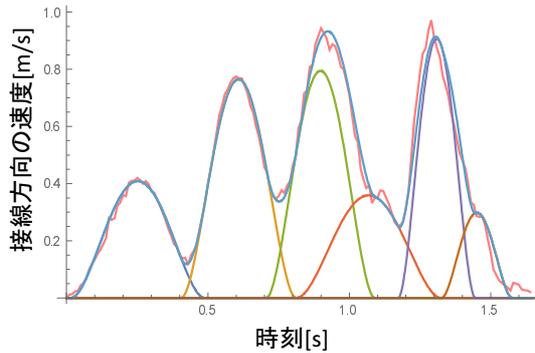
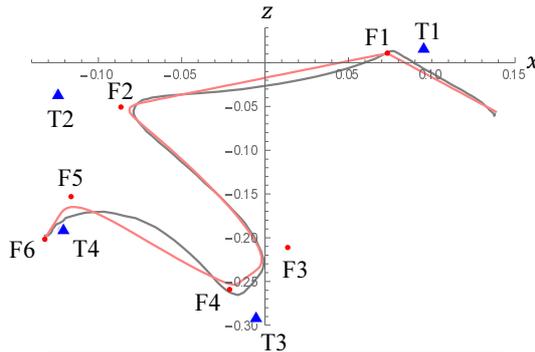


図 3 速度波形のフィッティング結果の例



順	位置	時刻	位置	TF	時刻
1	T1	0	6	F3	1.0897
2	T2	0.4829	7	T4	1.2159
3	F1	0.4886	8	F4	1.3242
4	T3	0.783	9	F5	1.4451
5	F2	0.8114	10	F6	1.5758

図 4 図 3 の運動の軌道・フィッティング結果 (xz 平面)

E に向けて手を到達させる。このとき、曲線を描く軌道となる。

- (4) モニタ上に表示された図形の位置 F に向けて直線的にはなく、曲線を描くような軌道で到達させる。このような場合には図形の位置 F 以外に経由点が発生する。

4. 手先動作の再現

3.4 節の結果から得られた特徴をもとに、3.3 節の計測における被験者の手先の運動を再現するモデルを構築し、このモデルに基づいて手先の到達運動の再現を CG キャラクタにて行った。

4.1 運動を再現するモデル・CG キャラクタにおける再現

3.3 節の計測と同様の状況において、被験者の手先の運動の再現を行った。(図 7) CG キャラクタにおいて、3.3 節の計測と同様に、モニタを模した板型 CG オブジェクトの上に表示した円形の図形を 0.1~1.6 秒のランダムな間隔で、ランダムな座標に移動させ、この円形の図形を到達目標位置とした。なお、円形の図形は任意の位置に、任意のタイミングでユーザの操作によって移動させることも可能とした。

3.4 節に述べた特徴 (1) ~ (3) を再現するためのモデルとして、図 5 のように設定した目標位置に、3 節の (8) 式により求まる軌道を描きながら到達するというモデルを考案した。このモデルに沿って行動するためのパラメータとして、図形が表示されてから手を動かし始めるまでの時間 $T_{reaction}$ 、到達開始から到達完了までにかかる時間 T_{reach} 、 T_{reach} の誤差とする時間の範囲 ϵ_{reach} 、到達位置の誤差 ϵ_{target} という 4 種類のパラメータを設定した。

各パラメータの値の求め方としては、 $T_{reaction}$ はモニタに図形が表示されてから到達運動が開始されるまでの時刻について複数回分の到達運動から得られた中央値を用いた。 T_{reach} は到達する距離を $l(cm)$ とすると、 l に比例する関数

$$T_{reach} = al + b \quad (10)$$

を仮定し、 a , b は計測データから求めた。 ϵ_{reach} には標準偏差を使用した。 ϵ_{target} には計測中に発生した到達位置の誤差の平均値 c を使用したが、ある距離 d 以下ではこの誤差は到達する距離 l と比例の関係があると仮定し (式 19)、計測データをもとに c , d , h を求めた。

$$\epsilon_{target} = hl \quad (l \leq d) \quad (11)$$

$$\epsilon_{target} = c \quad (l > d) \quad (12)$$

また、到達を実行中の到達目標リストを常に設定しており、そのリストには到達目標位置と到達開始・終了時刻等を格納している。その時刻における到達目標リストにあるすべての目標について、(8) 式より求まる軌道を描く。

円形の図形が表示されてから $T_{reaction}$ 秒経過後に、到達目標リストにその図形の位置が追加される。その際、到達の誤差を含めた到達目標位置を $\Delta i'$ 、到達開始時刻を s' 、誤差を含んだ到達完了時刻を g' 、 $rand$ を -1 から 1 の間の乱数とすると、

$$\Delta i' = \Delta i + \epsilon_{target} [rand, rand, rand]^T \quad (13)$$

$$s' = s \quad (14)$$

$$g' = T_{reach} + \epsilon_{reach} rand \quad (15)$$

と表され、到達目標リストには $(\Delta i', s', g')$ が追加される。 $T_{reaction}$ 秒よりも早く図形の位置が変化した場合には、到達目標リストへの追加は行われない。

上記のように設定された到達目標リストから、手先の軌道を次のように求めた。まず、到達目標が一つの場合、その目標に対する到達運動の時刻 t における位置を

$$P_{reach}(\Delta P_{target}, T_0, T_{reach}, t) \quad (16)$$

と定義する。ただし、 P_{reach} は図 6 のように表され、図 6 は式 (8) より求まる手先の軌道である。

N 個の目標地点への到達運動の軌道については、 i 個目の目標地点への運動について目標位置を Δi 、到達開始時

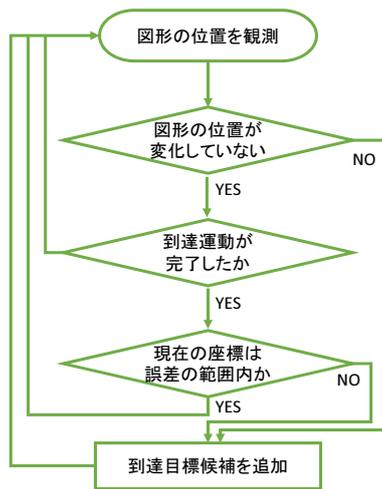


図 5 目標位置の設定の流れ

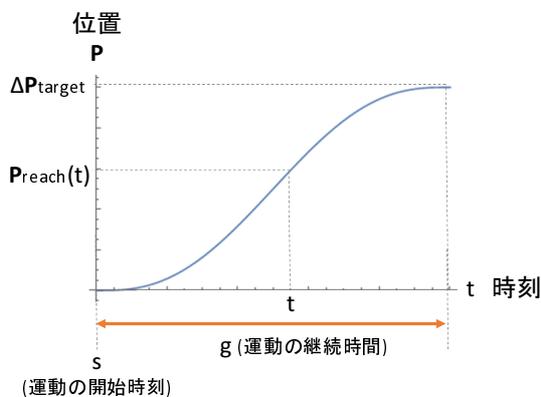


図 6 手先軌道

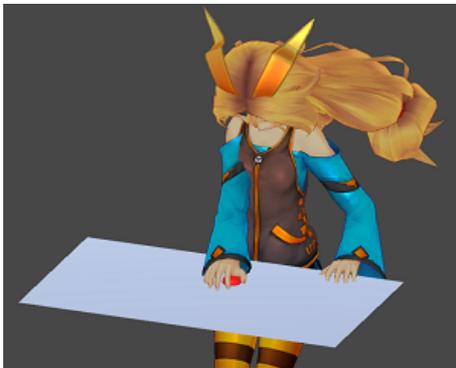


図 7 CG キャラクタで計測の再現を行った様子

刻を s_i , 到達終了時刻を g_i , 現在時刻を t とすると, 時刻 t における手先の位置は

$$\sum_{i=1}^N P_{reach}(\Delta i, s_i, g_i, t) \quad (17)$$

と表される.

4.2 評価

4 節のモデルをもとに CG キャラクタで運動の再現を Unity 上で行った. 運動の再現において 3D モデルにはユニティちゃんを, ユニティちゃんライセンス条項 [7] にし

たがって使用した. CG キャラクタには, 前述のモデルに従って手先の到達位置を指定し, Unity の IK を使用して動作生成を行った.

4.1 節に述べたそれぞれのパラメータは, 計測データから, $T_{reaction} = 0.155(s)$, $T_{reach} = 0.4008l + 0.3637(s)$, $\epsilon_{reach} = 0.13(s)$, ϵ_{target} は

$$\epsilon_{target} = 0.5l \text{ (cm)} \quad (l \leq 5 \text{ (cm)}) \quad (18)$$

$$\epsilon_{target} = 5 \text{ (cm)} \quad (l > 5 \text{ (cm)}) \quad (19)$$

と求めた.

次に, モーションデータから得られた軌道と本手法により再現した軌道の例を図 8(a)~(d) に示し, それぞれにおいて比較を行う. 図 8(a), (b) はモーションデータの軌道, 図 8(c), (d) は本手法により再現された軌道である. $T1 \sim T5$ はそれぞれ到達する目標位置 (3.3 節の計測では図形が表示される位置) を表し, 図 8(c)(d) における $T1' \sim T5'$, $T3''$, $T5''$ は到達の誤差を含めた到達目標位置を表しており, 図形が表示される目標位置とはずれた位置となる. なお, $T1 \sim T5$ はそれぞれ, 数字の順に目標位置が表示されたことを表し, $T1' \sim T5'$, $T3''$, $T5''$ は数字の順に目標位置として設定されたことを表す.

モーションデータから得られた軌道図 8(a) では, 最初に $T1$ へ到達しようとしたが, 到達の途中で目標位置が $T2$ へ切り替わったために曲線の軌道を描いたことが分かる. これは再現した軌道図 8(c) においても, $T1$ へ到達しようとしたが, 途中で目標位置が $T2$ に切り替わってしまったために曲線の軌道を描いており, モーションデータの軌道と似た状況を再現することに成功している.

また, モーションデータから得られた軌道図 8(a) で $T2$ へ到達完了する前に $T3$ が表示されたが, すぐに $T4$ に目標位置が変わってしまったために $T3$ への到達が省略されている. 同様に, 再現した軌道図 8(c) においても, $T2'$ に到達した後 $T3$ が表示されたが, 目標位置がすぐに $T4$ へ切り替わったために $T3$ への到達が省略されている.

図 8(a) では, $T1$ 付近と $T2$ 付近において曲率の違う曲線軌道が描かれているが, これは到達途中のどのタイミングで次の到達目標が表示されたかやどの位置に次の目標位置が表示されたかに依存する. 再現した軌道図 8(d) においても, $T1$ 付近や $T2$ 付近において図 8(a) の場合と似たように曲率の違う曲線を描くことに成功している.

さらに, モーションデータ図 8(b) においては, 一度 $T1$ とは少し離れた位置に到達したものの, 到達の誤差を修正するように $T1$ に向けて再び到達を行っていることがわかる. 再現した軌道図 8(c) においても, 一度 $T5'$ に到達したものの, $T5''$ へ到達の誤差が修正されており, 8(d) $T3'$, $T3''$ においても同様の状況が再現できていることがわかる.

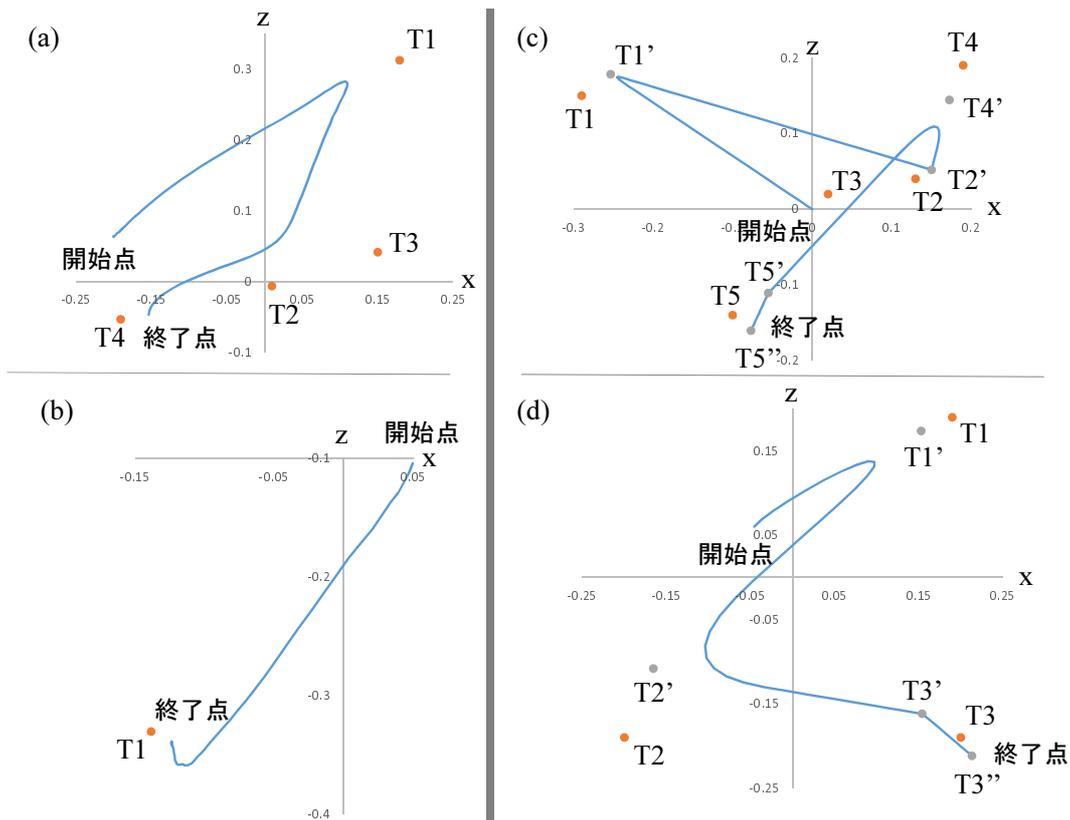


図 8 モーションデータから得られた軌道 (a)(b), 再現した軌道 (c)(d)

5. まとめ・展望

今回設定した計測において、手先のモーションデータに本分析手法を適用し、予想到達目標位置の列という時間的に離散なデータを得ることで、手先の到達速度にみられる特徴を確認することができた。さらに得られた特徴から手先の到達運動時のモデルを構築し、CG キャラクタでその運動の再現を行うことに成功した。

今後は、CG エージェントなどとの共同作業の実現に向け、本手法による到達運動の再現を共同作業時の CG キャラクタの動作に適用させたいと考えている。

参考文献

- [1] Yeo, Sang Hoon, et al. "Eyecatch: simulating visuomotor coordination for object interception." *ACM Transactions on Graphics (TOG)* 31.4 (2012): 42.
- [2] Wada, Yasuhiro, and Mitsuo Kawato. "A theory for curvilinear handwriting based on the minimization principle." *Biological Cybernetics* 73.1 (1995): 3-13.
- [3] Flash, Tamar, and Neville Hogan. "The coordination of arm movements: an experimentally confirmed mathematical model." *The journal of Neuroscience* 5.7 (1985): 1688-1703.
- [4] 川人光男, 脳の計算理論, 産業図書
- [5] 草野祐樹, 大村廉. "Jerk 最小モデルを用いた人の指さし行動予測手法の検討" *インタラクシオン* 2012
- [6] Williamson, Matthew M. "Postural primitives: Interactive behavior for a humanoid robot arm." *Fourth Inter-*

national Conference on Simulation of Adaptive Behavior. 1996.

- [7] UNITY-CHAN! OFFICIAL WEBSITE, 「ユニティちゃんライセンス条項 1.01 バージョン」, [<http://unity-chan.com/contents/license-jp/>] (最終検索日: 2015 年 7 月 27 日)