



アバタの構造・動作の実装を効率化するツールキット 「PoseSynth」の提案

山口周¹⁾, 畑田裕二¹⁾, 橋浦健太¹⁾, 鳴海拓志¹⁾

1) 東京大学 (〒 113-0033 東京都文京区本郷 7-3-1, { yamaama, hatada, k.hashiura, narumi } @cyber.t.u-tokyo.ac.jp)

概要: 現実の身体のあり方を超越した構造・動作を有するアバタは様々な研究領域で役立てられている一方で、その実装方法は様々であり、再利用が困難である。本研究では、アバタの構造・動作を扱うための規格を策定し、多様な構造を有するアバタの多様な動作を効率良く実装できるツールキット「PoseSynth」を提案する。PoseSynth はアバタの姿勢を変換して別のアバタの姿勢に代入する処理を繰り返すことで、様々なアバタの動作を効率良く実装できる。

キーワード: アバタ, ツールキット, 融合身体, 分身体

1. 序論

バーチャル環境におけるユーザの身体であるアバタは、その構造や動作を自由に編集できる。そのため、アバタを活用すると、身体認知メカニズムの調査や、身体変容が認知や発揮能力に与える影響の検証と工学的応用を実環境よりも広範かつ容易に行える。このような研究では、例えば、融合身体 [1] や分身体 [2] のように実身体と異なる動作をするアバタや、身体部位の追加や変更によって実身体と異なる構造を持つノンヒューマノイドアバタ等、現実の身体のあり方を超越した構造・動作を有する多様なアバタが利用されてきた。他方、これらの多様な構造・動作を有するアバタは個別の研究で独自に定義され、その実装も様々であるために、再現や再利用が困難であるという問題がある。

本研究では、アバタの構造・動作を扱うための規格を策定し、多様な構造を有するアバタの多様な動作を効率良く実装可能にするツールキット「PoseSynth」を提案する。PoseSynth におけるアバタの構造と姿勢は Pose というデータによって表現され、Pose に対する複数種類の変換器によって所定の変換がなされる。PoseSynth のソースコードおよびサンプルは 2024 年 9 月現在 GitHub¹⁾ 上で公開されており、誰でも利用することができる。変換器のうち広範な用途が見込めるものは予め実装されている他、PoseSynth のユーザが新たに変換器を実装することで機能を容易に拡張することができる。本稿では本研究の対象となるアバタとその実装方法について紹介した後、PoseSynth のシステムについて説明する。その後、実際に PoseSynth で実装できるアバタの例を示す。

2. 関連研究

2.1 融合身体

複数人で 1 体のアバタを動かす融合身体は、共同行為における主体感の調査 [1] や、先生から生徒への技能の伝達 [3]

等の研究に使用されている。融合の方法は加重平均型と部位選択型の 2 種類に大別される。加重平均型融合身体はユーザごとに重みを定め、それぞれの姿勢の加重平均をとることでアバタの姿勢を決定する方法である [4]。対して部位選択型融合身体は、アバタの部位をユーザがそれぞれ選んで動かすことで姿勢を決定する方法である。例えば、パートナーの意図を視覚的に知ることの重要性を示した Hapuarachchi and Kitazaki の研究 [5] では 2 人のユーザがそれぞれ右腕と左腕を動かす部位選択型融合が使用された。

2.2 分身体

1 人のユーザが複数のアバタを同時に動作させる分身体は、ヒトの身体認知メカニズムや新しい身体・環境に対する適応の柔軟性を調べる研究等に利用されてきた。複数の身体それぞれに所有感や主体感が生起することを示した Miura et al. の Multisoma [2] では、異なる場所に配置された 4 体のアバタがそれぞれ同じ姿勢を取る。

2.3 ノンヒューマノイドアバタ

これまでに挙げてきたアバタの構造はユーザと概ね一致している (ヒューマノイドアバタ) が、身体部分の追加 [6] や変更 [7] によってユーザとは大きく異なる構造を持つノンヒューマノイドアバタを用いた研究も存在する。その動作の方法は、腕の追加のようにヒューマノイドアバタで用いられている動作がそのまま流用できる場合もあれば、関節同士の対応付け等の特殊な動作生成を用いる必要がある場合もある。前者の場合は入力自由度の補完のため 2 人目のユーザやアニメーション等を用いることがある。また、後者は現在も研究が進められており、動作の方法に関する明確な合意は未だなされていない。

3. PoseSynth

3.1 PoseSynth の要件

上述の先行研究をまとめると、アバタの構造はヒューマノイドを基本としつつも、一般には不定である。また、動作方法は様々であり、時に複数人での操作が必要になる。こ

¹⁾<https://github.com/Cyber-Interface-Lab/PoseSynth>

れを踏まえ、本研究で提案するツールキットの要件を以下のように定める。

1. 任意の構造のアバタに対応すること
2. 容易に拡張できる複数の処理の連結によってアバタの動作を実装できること
3. マルチユーザに対応すること

3.2 アバタの構造と姿勢のデータ化

PoseSynth ではアバタの構造と動作を扱うために Pose という規格でデータを扱う。Pose は各種の変換器によって参照・代入され、その処理結果はアバタの姿勢に即座に反映される。

Pose はアバタの足元の位置・姿勢を表す 3 次元ベクトル・クォータニオンと、各関節の姿勢を表すクォータニオンからなる。各関節は関節グループというラベル付きのグループに分割される。例として、サンプルにおけるヒューマノイドアバタの Pose の関節グループの取り方を図 3.2 に示す。

ラベル	関節数
Spine	4
Head	3
Left Arm	4
Right Arm	4
Left Finger	20
Right Finger	20
Left Leg	5
Right Leg	5

図 1: サンプルにおけるヒューマノイドアバタの Pose とその関節グループ。

3.3 ユーザの情報の管理と Pose への反映

ユーザの情報は多くのバーチャルリアリティ (VR) システムにおいて、トラックの位置・姿勢の情報を保持する点群の集合 (カメラリグ) によって表現されるが、その階層構造は VR システムによって異なる。特定の VR システムへの依存を防ぐため、PoseSynth は点群に対して独自のバインドを設定し、バインドを介して点群を管理する。そのため、PoseSynth は実験設定に応じて複数種類の VR システムの中から使用するものを選択できる。

バインドを介して取得したユーザの実身体の位置・姿勢などの情報は PoseMapper と呼ばれる変換器によって Pose に適用される (図 2)。PoseMapper はアバタ全身の姿勢の決定に用いる逆運動学 (IK) などの外部システムを PoseSynth に適応させたものである。外部システムを PoseSynth の仕組みである PoseMapper として運用することで、PoseSynth と外部システムとの疎な結合が実現される。PoseMapper は PoseSynth のユーザが新しい処理を追加で実装できる。

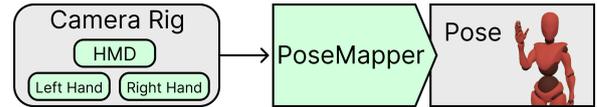


図 2: バインドと PoseMapper を利用してアバタの姿勢を決定する流れ。図ではカメラリグにヘッドマウントディスプレイ (HMD) と両手の位置・姿勢を示す点群の情報が存在し、それぞれ「HMD」「Left Hand」「Right Hand」のバインドが割り当てられている。PoseMapper はバインドから取得した HMD と両手の位置・姿勢の情報を元に Pose に値を代入し、アバタの姿勢を決定する。

3.4 任意数の Pose の重ね合わせ

Pose は各種の変換器によって値を変更できる。PoseMixer は任意数の Pose を引数として、関節グループごとに決められた重みを用いて重み付き平均を取ることで、重ね合わせた値を他の Pose に代入する変換器である。重み付き平均はクォータニオンの球面線型補間 (Spherical linear interpolation, Slerp) を利用する。一般に 2 個のクォータニオンは、2 個目のクォータニオンの寄与率を示す媒介変数 $t \in [0, 1]$ を用いた以下の式で補間できることが知られている。

$$q = \text{Slerp}(q_1, q_2, t) = \frac{\sin(1-t)\phi}{\sin\phi} q_1 + \frac{\sin t\phi}{\sin\phi} q_2 \quad (1)$$

PoseMixer の変換処理はこの球面線型補間を任意数のクォータニオンの場合に拡張したものをを用いる。N 個のクォータニオン q_i ($i = 1, 2, \dots, N$) に対してそれぞれ重み w_i を定め、重みを正規化した値を寄与率として補間したクォータニオンを計算することを考える。補間は q_1 と q_2 から順に行う。まず q_1 と q_2 を球面線型補間して得られたクォータニオン $q_{12} = \text{Slerp}(q_1, q_2, w_2/(w_1 + w_2))$ に対して重み $w_{12} = w_1 + w_2$ を置く。 q_{12} と q_3 を球面線型補間すると $q_{123} = \text{Slerp}(q_{12}, q_3, w_3/(w_{12} + w_3))$ を得る。 q_{123} における q_1 , q_2 , q_3 それぞれの寄与率を考える。まず q_3 については $w_3/(w_1 + w_2 + w_3)$ と計算できる。次に q_2 は q_{123} における q_{12} の寄与率に q_{12} における q_2 の寄与率をかけることで計算でき、その寄与率は $w_2/(w_1 + w_2 + w_3)$ となる。同様に q_1 の寄与率は $w_1/(w_1 + w_2 + w_3)$ となる。したがって、 q_{12} の重みを $w_{12} = w_1 + w_2$ と置くことで、重みを正規化した値を寄与率として補間することができる。 q_4 以降との補間についても q_{123} の重みを $w_{123} = w_1 + w_2 + w_3$ とするなどして同様に補間できる。

重みは各 Pose の関節グループごとに用意されており、PoseSynth のユーザは API や図 4 に示すようなグラフィカルユーザインタフェース (GUI) を用いて重みの値を参照・変更できる。引数の Pose の関節グループのラベルのいずれかが代入先の Pose に含まれない場合は、その関節グループに関する計算を行わない。

なお、1 個の Pose のみを PoseMixer の引数とした場合は恒等変換になり、どのように重みを設定しても引数の Pose の値をそのまま出力して別の Pose に代入する。

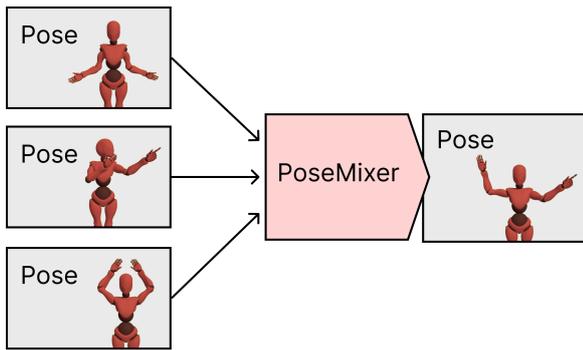


図 3: PoseMixer による任意数の Pose の重ね合わせ。

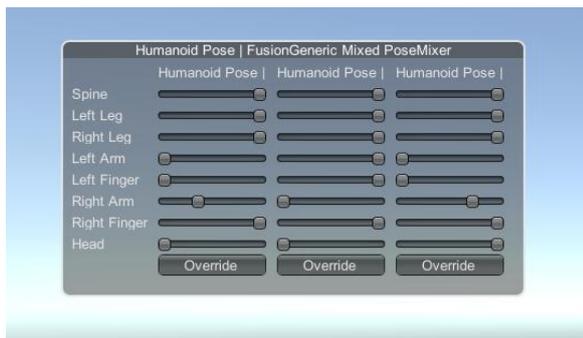


図 4: PoseMixer における重み調整用の GUI。図中では表 3.2 に示す関節グループを持つヒューマノイドアバタ 3 体の Pose を重ね合わせている。

3.5 Pose の値の変更

1 個の Pose を引数として、その値に所定の変更を加えた値を別の Pose に代入する処理 (図 5) を行う変換器を PoseRemapper と呼ぶことにする。変更の処理は具体的な PoseRemapper の種類により異なる。PoseRemapper は PoseSynth のユーザが独自に変更の処理を実装することを想定している。ここではサンプルとして予め実装されている PoseRemapper についてそれぞれ説明する。

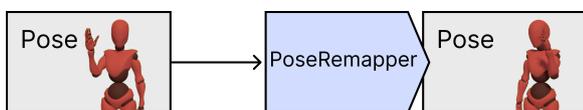


図 5: PoseRemapper を用いた Pose の値の変更の流れ。PoseRemapper の例として PoseMirror を用いて Pose の姿勢を左右反転させている。

PoseRemapper の一種である PoseMirror は Pose の各関節の姿勢を左右で反転させた値を別の Pose に代入する。反転処理は関節グループごとに行うが、左右で対になる関節グループがあればそれも考慮する。対になる関節グループの有無の判断はラベルに左右を示す特定のキーワード (デフォルトでは「Left」および「Right」) が含まれ、かつそれ以外の部分が両者で共通していることである。反転処理は、各関節の x 軸、y 軸、z 軸周りの回転量のいずれかに -1 をかけることで行う。反転処理を行う軸は PoseSynth のユー

ザが選択できる。例えばヒューマノイドアバタは y 軸と z 軸周りの回転量に関して反転を行うことで全身の姿勢を反転できる。

PoseRemapper の一種である PoseDelayer は Pose の値を決められたフレーム数だけ保持し、最も古い値を別の Pose に代入することで動作を遅延させる。

3.6 マルチユーザへの対応

融合身体のように複数のユーザからなる実験系への対応のため、PoseSynth には Unity 公式プラグインである Netcode を用いたマルチユーザ機能が用意されている。Netcode はサーバ・クライアント方式による通信を採用しており、クライアントの入力に応じてサーバが系全体の計算処理を行い、その結果をクライアントに反映させる。PoseSynth におけるクライアントの入力はバインド付きのカメラリグである。クライアントがサーバに接続すると、サーバ側にクライアントのカメラリグと同期するカメラリグが生成される。サーバはそのカメラリグを元にシングルユーザの場合と同様に系全体の計算処理を行う。計算処理の結果として得られたアバタの Pose はクライアントに同期され、クライアントは HMD を通してアバタの様子を確認できる。

4. PoseSynth によるアバタの作成例

前章で説明した Pose と各種の変換器を組み合わせると、様々な構造・動作を取るアバタを作成することができる。本章ではその例を示す。なお、章中の各図左側にある Pose は全て図 2 のように PoseMapper を使ってユーザの実体の位置・姿勢の情報が反映されているものとし、当該 Pose が複数存在する場合はマルチユーザであるとする。

融合身体: 2 体のアバタの Pose を PoseMixer で重ね合わせることで作成できる。それぞれの Pose の重みの調整の仕方によって加重平均型融合身体 (図 6) と部位選択型融合身体 (図 7) のそれぞれを作成できる。

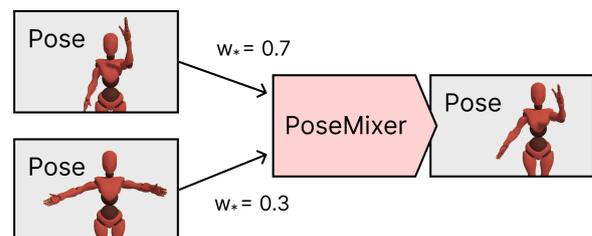


図 6: PoseSynth による加重平均型融合身体の実例。左上のアバタを 70 %、左下のアバタを 30 % だけ融合して右のアバタに反映させている。

分身: アバタの Pose の値を別のアバタの Pose にそのまま代入した後、その Pose の足元の位置・姿勢を変更することで作成できる (図 8)。

ノンヒューマノイドアバタ: Pose の関節グループの取り方は PoseSynth のユーザが任意に決定できるため、ノンヒューマノイドアバタにも問題なく設定できる。特に追加肢を持つヒューマノイドアバタは、ヒューマノイドアバタ

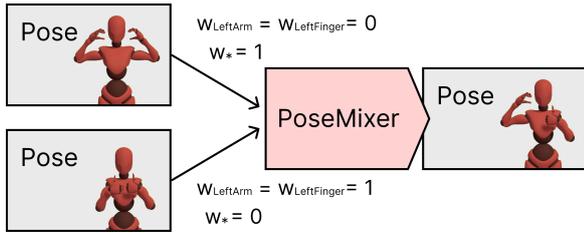


図 7: PoseSynth による部位選択型融合身体の作成例。左上のアバタに左下のアバタの左腕を融合して右のアバタに反映させている。

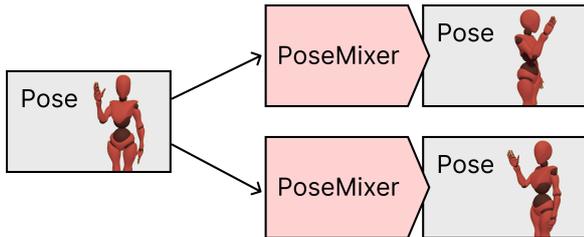


図 8: PoseSynth による 2 体のアバタへの分身の作成例。

と追加肢それぞれに Pose を設定した後、それらを別々の変換器で動作させることで作成できる (図 10)。サンプルでは左腕または右腕のみのアバタが用意されており、簡単に追加肢をヒューマノイドアバタに追加できる。左腕のみのアバタの関節グループを図 9 に示す。腕のみのアバタはヒューマノイドアバタの腕部分と同じラベル・関節数の関節グループが設定されているため、必要に応じて PoseMixer による恒等変換を用いることでヒューマノイドアバタの腕の姿勢を適用することができる (図 10 上)。

	ラベル	関節数
	Left Arm	4
	Left Finger	20

図 9: 左腕のみのアバタの Pose とその関節グループ。

5. 結論

本研究では多様な構造のアバタの多様な動作を効率良く実装可能にするツールキット「PoseSynth」を提案し、その仕組みとアバタの作成例を説明した。PoseSynth は既存のアバタを容易に再現できることに加え、アニメーションや AI との連携による新しい形式のアバタを作成できる可能性がある。また、Pose や 変換器は拡張を前提としているため、ユーザが新しい Pose や変換器を作成することで様々な構造・動作のアバタに対応できる。今後は具体的なアバタ研究に実際に活用していくことで研究の効率化に貢献するとともに、研究者からのフィードバックを踏まえた改良を施すことで、実用性を向上していきたい。

謝辞 本研究の一部は、JST ムーンショット型研究開発事業 (JPMJMS2013) の支援を受けて行われた。

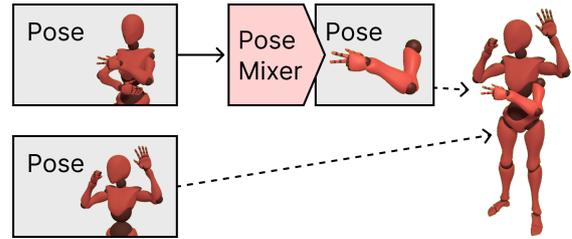


図 10: 追加肢を持つヒューマノイドアバタの作成例。

参考文献

- [1] R. Fribourg, N. Ogawa, L. Hoyet, F. Argelaguet, T. Narumi, M. Hirose, and A. Lécuyer, “Virtual Co-Embodiment: Evaluation of the Sense of Agency While Sharing the Control of a Virtual Body Among Two Individuals,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 27, no. 10, pp. 4023–4038, 2021.
- [2] R. Miura, S. Kasahara, M. Kitazaki, A. Verhulst, M. Inami, and M. Sugimoto, “Multisoma: Motor and Gaze Analysis on Distributed Embodiment With Synchronized Behavior and Perception,” *Frontiers in Computer Science*, vol. 4, 2022.
- [3] D. Kodama, T. Mizuho, Y. Hatada, T. Narumi, and M. Hirose, “Effects of Collaborative Training Using Virtual Co-embodiment on Motor Skill Learning,” *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 5, pp. 2304–2314, 2023.
- [4] T. Hagiwara, G. Ganesh, M. Sugimoto, M. Inami, and M. Kitazaki, “Individuals Prioritize the Reach Straightness and Hand Jerk of a Shared Avatar over Their Own,” *iScience*, vol. 23, no. 12, p. 101732, 2020.
- [5] H. Hapuarachchi and M. Kitazaki, “Knowing the intention behind limb movements of a partner increases embodiment towards the limb of joint avatar,” *Scientific Reports*, vol. 12, p. 11453, Jul 2022.
- [6] N. Yamamura, D. Uriu, M. Muramatsu, Y. Kamiyama, Z. Kashino, S. Sakamoto, N. Tanaka, T. Tanigawa, A. Onishi, S. Yoshida, S. Yamanaka, and M. Inami, “Social Digital Cyborgs: The Collaborative Design Process of JIZAI ARMS,” in *Proceedings of the 2023 CHI Conference on Human Factors in Computing Systems*, CHI '23, (New York, NY, USA), Association for Computing Machinery, 2023.
- [7] Y. Seol, C. O’Sullivan, and J. Lee, “Creature Features: Online Motion Puppetry for Non-Human Characters,” in *Proceedings of the 12th ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA '13, (New York, NY, USA), p. 213–221, Association for Computing Machinery, 2013.