

大規模な動作データに基づくシミュレーションミドルウェア

栗山 繁 向井 智彦 入野 祐輔 按田 和幸 花村 真孝

豊橋技術科学大学 情報工学系

1. はじめに

オートメーション化された組立工場での作業効率の検証に3次元CGで構築されたデジタルヒューマン(以後、仮想人間)の技術が利用されている^[1]。しかし、動作のシミュレーションにはロボットのモデルを用いているので、人間としての動作の自然さに欠けることが問題点として指摘されている。また、CGアニメーションツールを用いた手作業での動作入力には、多くの手間と熟練作業を必要とする。一方、近年のモーションキャプチャ技術の進歩により、3次元計測した動作データを用いて人間の自然な動作アニメーションを製作する方法が普及しており、動作データの変形・合成に関しても多くの手法^[2]が提案されている。

本研究では、作業環境のシミュレーションに特化した大規模な動作データベースシステムとそのデータの知的な処理機構をミドルウェアとして提供する。これにより、自然な動作生成に基づく高品質なシミュレーションシステムが、低コストで構築されるようになる。

2. システム概要

ミドルウェア群はXMLとWeb3Dの技術を基盤として開発され、それらはView-Control-Model(VCM)の三層の設計パターンを構成する(図1参照)。すなわち、シミュレーションに使用するデータベースの層とそれを制御する層、および制御結果を可視化する層に大別され、各層が独立したパッケージとしても機能するように構成される。

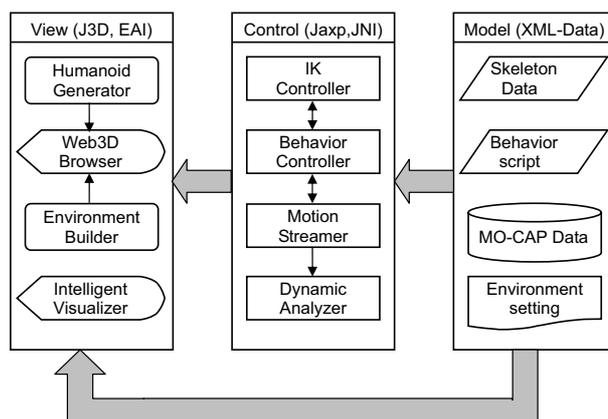


図1. ミドルウェア群構成図

◆ View: 動作表示層(Web3D ブラウザ)

無償配布されているVRMLブラウザでの表示を可能とするために、Javaを用いたExternal Authoring Interface(EAI)を用いて仮想人間の表示を制御する。具体的には、骨格データからWeb3Dコンソーシアムで提案されているヒューマノイドアニメーション(H-Anim)形式^[3]のデータを自動生成する機能(Humanoid Generator)と、環境設定データからVRMLデータを自動生成する機能(Environment Builder)をミドルウェアとして実装する。また、大規模動作データ群の分布パターンや運動学的または動力的な特徴等を直観的かつ知的に可視化する機能(Intelligent Visualizer)を、Java3D(J3D)を用いて実装する。

◆ Control: 動作制御層(データの遷移、変形)

動作データはシミュレーションの対象となる環境や体型に適合するように知的に変形・合成する^[4]必要がある。ゆえに動作制御層では、逆運動学の手法を拡張して手先や足先の到達位置を修正する機能(IK-Controller)を提供する。また、

行動規則の記述に基づき、複数個の単純な動作データを遷移させて連続的な動作を自動生成する機能(Behavior Controller)や、生成された動作姿勢を動的可操作性に基づき検証する機能(Dynamic Analyzer)もミドルウェア化する。ただし、逆運動学や可操作性の計算等多くの演算量を要するものに関しては、実行時の応答性能を考慮してネイティブコードで実装したものをJNI(Java Native Interface)を用いて呼出す。

◆ **Model:** データ層(動作、骨格、行動、環境)

動作データの様式は複数種類が存在する^[5]が、使用するデータの汎用性と拡張性を考慮して、データ構造の取り扱いが容易な XML 形式に変換する。また、人体骨格のデータ、行動規則、および環境の設定も XML 文章として保存し、動作制御層の各ミドルウェアは、それらの XML 文書を JAXP(Java API for XML Processing)を用いて読み込む。

3. 大規模動作データベースの構築

本研究は作業環境での行動シミュレーションを目的とするので、年齢層と体型の異なる複数の被験者に対して、日常的または作業的な動作に特化して動作計測を実施した。したがって、身振りや挨拶などの表現手段としての動作や、舞踏、格闘技、体操、およびスポーツ等の競技としての動作は計測の対象外とした。

計測は Vicon8 Motion Capture System (光学式計測) を用い、被験者はプロの俳優を含む、男性 7 名(痩せ型、肥満型、筋肉質、および平均的体型の成人と高齢者)と女性 5 名(長身、小柄、筋肉質、および平均的体型の成人と年少者)を選定した。その結果取得された動作データは、1 名につき 30 分程度であり総計約 6 時間となった。データベースは 5 章で述べる情報可視化の機能を用いた検索を可能とする。

表 1 に計測した動作データの種別とその内容を示す。

動作種別	計測した動作内容
移動系	クロソイド曲線に沿った歩行と走行(速度、歩幅、向きの変更) 荷物を持った歩行(両手、片手、2人共同) 障害物の回避(扉の開閉を含む) 昇降動作(階段、斜面、梯子) 複数人間の衝突回避
操作系	手掴み(片手と両手で12ヶ所、座位、立位) 荷物移動(片手と両手で6ヶ所の台位置) 押し引き動作(3箇所)、長棒の操作 組立て作業の基本動作
休息系	座り(椅子2種)、もたれ、寝転び
その他	関節の曲げ限界測定(腕部、脚部、腰背骨部)

表 1. 計測動作項目

4. 行動規則のXML文書化とその制御

行動シミュレーションのために CG 仮想人間の動作を制御する規則を XML 文書の形式で記述し、その内容に基づいて動作データを知的に処理する機構を開発した。行動を制御するために XML のタグとして定義される要素を以下に列挙する。

- ◆ **Human:** 人体骨格のデータ、初期行動、初期位置、および感知器等を指定
- ◆ **Resource:** 動作データを含むファイルと、前処理を行う変換クラスを指定
- ◆ **Motion:** 動作データの再生範囲と遷移状態における動作の補間期間等を指定
- ◆ **Behavior:** 再生する動作データを順に列挙し、仮想人間の行動様式をモジュール化
- ◆ **Sensor:** 感知器とその反応行動、物体操作に関するクラスを指定
- ◆ **Channel, Manipulator:** Sensor から送られる情報に基づく行動選択と物体操作

図 2 にこれらの要素を用いて記述された、2 名の仮想人間の簡単な行動制御の例を示す。

仮想人間のアニメーションは、動作データを部分的に抜き出して加工したものを逐次遷移させて生成する。このとき、動作の滑らかな状態遷移を実現するために、人体中心の位置と向きの変換情報を調節する機構と、関節角度を補間

する機構を実装する。また、与えられた環境に対して仮想人間を自律的に行動させるために、感知器－反応器モデルを導入する。このモデルは VRML 形式で与えられる環境データと仮想人間の状態との幾何学的な関係を計算する機能と、その値に応じて動作を自動的に選択・変形させる機能を備える。また、仮想人間の物体操作を制御する操作器モデルも導入し、この操作器から逆運動学に基づく手先や足先位置の調整機構 (IK Controller) が呼び出される。

```
<?xml version="1.0" encoding="UTF-8" ?>
<World title="SimpleExample">

<Human name="Taro" skeleton="TaroBone"
behavior="gait1" cond="1. 0. 0." >
  <Sensor class="CollisionSensor" id="Cols" data="Jiro">
    <Channel state="near" target="gait2" />
  </Sensor>
</Human>
<Human name="Jiro" skeleton="JiroBone"
behavior="gait2" cond="-1. 0. 2." >
  <Sensor class="ObjSensor" id="PickUp" data="Object">
    <Manipulator class="handManip" state="near"
operation="take" />
  </Sensor>
</Human>

<Resource file="gait" id="walk" segment="100 400" />
<Resource file="gait" id="walkMirror" segment="100 400"
converter="Mirror" />
<Resource file="jog" id="run" segment="100 400" />

<Behavior id="gait1" loop="true">
  <Motion resource="walkMirror" frame="0 100 25" />
  <Motion resource="run" frame="0 150 25" />
</Behavior>
<Behavior id="gait2" loop="true">
  <Motion resource="walk" frame="0 200 25" />
  <Motion resource="run" frame="0 75 25" />
</Behavior>

</World>
```

図 2. 行動制御の XML 記述例

シミュレーションシステムを柔軟に構築するために、Sensor や Manipulator 等の動作規則は XML 記述とはせずに、Java の基底クラスを拡張する方式を採った。しかし、これらの機能の設計パターンに共通性が認められれば、行動規則と同様に XML 文書としての記述方式を検討する。

5. 動作データベースの可視化

一般に、動作データを確認する手段としては、仮想的な人体モデルを用いた CG アニメーション表示が考えられる。しかしながら、大規模な動作データベースから必要な動作部分のみを抽出したり、動作の個人間での相違を発見したりするには、通常の人体表示では多くの試行錯誤を要することが予想される。したがって、大量のデータの分布が一覧表示され、その特徴や固有のパターン、および類似性等が容易に把握できるような可視化方法が望まれる。本研究では、人間の動作データを時間軸に沿って変動する多次元情報として扱い、そのような情報に対する可視化機構をミドルウェアとして実装する。

現段階では、動作データを人体の姿勢を効率よく配置した地図上の曲線群として表示するために、自己組織化マップ SOM^[6] (Self-Organizing Maps) を用いた可視化機構を実装している。さらに階層的な表示方法を開発して、大規模データの可視化技術に必要な Focus + Context の機能を実現する。

図 3 に、実装した動作データの可視化例を示す。

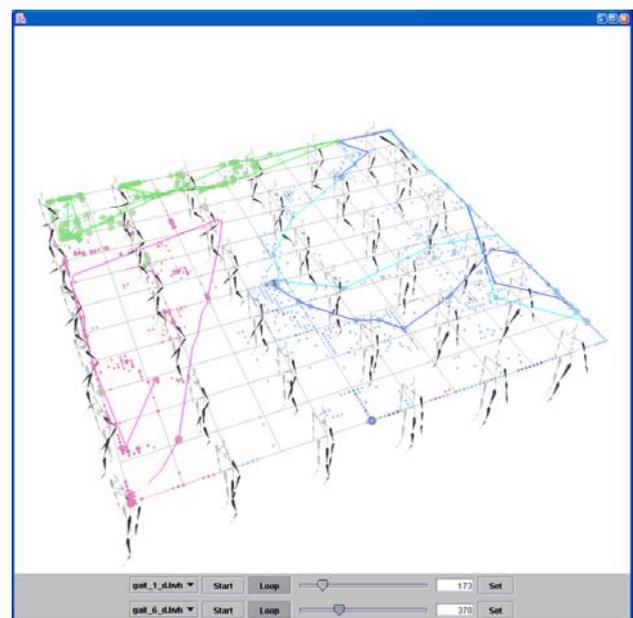


図 3. SOM を用いた動作データの可視化例

6. 適用事例

本研究で開発されたミドルウェアは、自動車組立工場の設備配置設計支援システムの開発に用いる予定である。また、不慣れな作業従事者の教育に用いるアニメーション・マニュアルの開発も計画している。図 4 に、自動車組立作業のシミュレーション結果の表示例を示す。



図 4. 作業シミュレーション結果の表示例

7. おわりに

人間工学に基づくシミュレーションを実現するには、広範な種類の動作を環境や体型の差異を考慮して高精度に生成することが必須となる。我々の研究チームでは、多くの被験者に対して計測した多様な動作データから高品質な大規模データベースシステムを構築し、信頼性の高い作業動作の自動生成機構を、シミュレーションミドルウェアとして開発した。本システムは、商業店舗やオフィス等の工場以外の設備環境でのシミュレーションも視野に入れて開発されているが、今回構築した動作データベースをそのような環境に対してどれだけ有効に用いることができるかを今後検証していく必要がある。

現時点では無償配布されている VRML ブラウザと、Java3D によって開発されたブラウザを併用しているが、実用的にはブラウザ環境の統合を検討する必要がある。これにはオープンソース

の Xj3D^[7] (Java で開発されている X3D のブラウザ) を用いてカスタマイズの可能なブラウザを開発することが考えられる。

今後は動作のシミュレーション機構に対して行動規則の文書例を多く作成し、その動作結果によりシステム全体の性能評価を行う。また、動作や行動の制御に関する記述形式の妥当性を検証し、シミュレーションミドルウェアとしての有用性、再利用容易性、および拡張性などを評価していく。さらには、行動規則を自動的に生成する、より高次の記述形式または入力インタフェースについても検討していく予定である。

謝辞

本研究は情報処理振興事業協会 (IPA) における、平成 14 年度未踏ソフトウェア創造事業によるものである。また、本研究の一部は SUZUKI(株)とのプロジェクト研究による資金と栢森情報科学振興財団・第 6 回助成事業による援助を受けた。

参考文献

- [1] N. I. Badler, C. A. Erignac, and Y. Liu, "Virtual Humans for Validating Maintenance Procedures", *Communications of the ACM*, Vol. 45, No. 7, pp. 57-63 (2002-7)
- [2] M. Gleicher, "Comparing Constraint-Based Motion Editing Methods", *Graphical Models*, Vol. 63, No. 2, pp. 107-134 (2001-3)
- [3] <http://h-anim.org/>
- [4] 栗山繁, 産業技術総合研究所ライフサイエンス関連研究講演会「Digital Human Modeling - Motion Generation -」"モーションキャプチャデータの知的再利用技術", pp. 39-42 (2002-3)
- [5] A. Menache, *Understanding Motion Capture for Computer Animation and Video Games: Chap. 4*, Morgan Kaufmann, pp. 121-142 (2000)
- [6] T. Kohonen, *Self-Organizing Maps*, Springer (1994)
- [7] <http://www.web3d.org/TaskGroups/source/xj3d.html>