

北海道大学大学院 ○岩館 健司, 米陀 佳祐, 柴田 勝也, 鈴木 育男, 山本 雅人, 古川 正志

現在の三次元コンピュータグラフィクスアニメーションは物体のモーション製作に多くの時間を浪費する。そこで本研究では、モデリングした剛性体に対し、物理法則に従った動作を自動的に生成する物理モデリングソフトウェアの開発を目的とする。本稿では、製作したソフトウェアの構成、物理モデル、データベース形式、モーション生成方について述べる。

1. 緒言

近年、コンピュータグラフィクスアニメーション技術は工学分野、映画産業をはじめとする様々な分野で必要不可欠な技術となっている。しかし、現在の三次元コンピュータグラフィクスアニメーションは、モーションキャプチャや、キーフレーム、キーポイントを用いて作られることが多く、物体のモーション製作に多くの時間を浪費してしまうのが現状である。

そこで本研究では、モデリングした剛性体に対し、物理法則に従った動作を自動的に生成する物理モデリングソフトウェアの開発とこれを用いたシミュレーションの実施を目的とする。

2. ソフトウェアの構成

本節では、本研究で製作したソフトウェアの大まかな流れについて述べる。Fig.1 にソフトウェアの構成を示す。

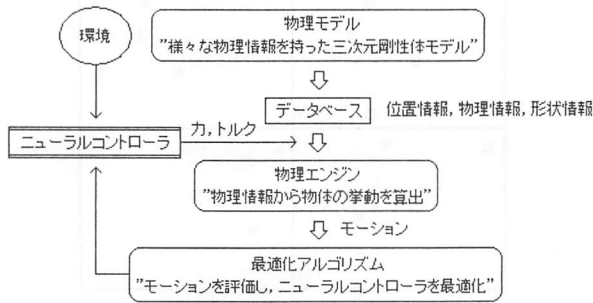


Fig.1 Software architecture

本ソフトウェアでは、生成した物理モデルから、位置情報、物理情報、形状情報を抜き出したデータベースを作成し、これにニューラルコントローラから与えられた力、トルクを加え、物理エンジンに渡す。物理エンジンによって計算されたモーションを最適化アルゴリズムによって評価し、ニューラルコントローラを最適化することでより効率的なモーションを獲得する。

3. 物理モデル

本節では、開発中のソフトウェアに実装した物理モデルについて述べる。

3.1 構成要素

物理モデルを作成するにあたって、本ソフトウェアでは以下の3つの要素から物理モデルを構成している。以下に各要素について詳細を述べる。

① プリミティブ (基本立体)

- ・球
- ・直方体

② ジョイント

- ・接合ジョイント
- ・関節ジョイント

③ センサ

- ・光センサ

3.1.1 プリミティブ

プリミティブとは、物理モデル製作時に使用する基本的な形状の立体である。本ソフトウェアでは球と直方体を実装した。

3.1.2 ジョイント

ジョイントは、複数のプリミティブ同士を接合する役割を担っており、本ソフトウェアでは、複数のプリミティブ同士を接合して一つの物体とするための接合ジョイントと、複数のプリミティブの間を結ぶ稼動可能な関節となる関節ジョイントの二種類を実装した。

3.1.3 センサ

センサは、作成した物理モデルの五感を担っており、本ソフトウェアでは目に相当する光センサを実装した。

3.2 ソフトウェア概観

物理モデル作成時の本ソフトウェアの概観は、一般的な三次元コンピュータグラフィクス作成ソフトを参考に製作しており、ユーザは、三次元コンピュータグラフィクスを製作する要領で物理モデルを作成することが可能である。

4. データベース

3 節で作成された物理モデルは、以下に示すデータ構造で保存され、さまざまな解析、シミュレーションに対して適応することが可能である。データベースの内容を Table1 に示す。

Table1 Data base

| ① 剛性体データベース   |                          |
|---------------|--------------------------|
| 形状情報          |                          |
| ・プリミティブタイプ    | (sphere, box)            |
| ・サイズ          | (rx, ry, rz)             |
| 位置情報          |                          |
| ・位置           | (x, y, z)                |
| ・回転角度         | (alpha, beta, gamma)     |
| 材料定数          |                          |
| ・密度           | (density)                |
| ・ヤング率         | (E)                      |
| ・ポアソン比        | (μ)                      |
| ② ジョイントデータベース |                          |
| 形状情報          |                          |
| ・ジョイントタイプ     | (fixed, revolute)        |
| ・接合対象         | (object1, object2)       |
| ・アンカーポイント     | (jx, jy, jz)             |
| ・自由度          | (tx, ty, tz, rx, ry, rz) |
| 材料定数          |                          |
| ・バネ係数         | (k)                      |
| ・ダンパ係数        | (c)                      |
| ③ センサデータベース   |                          |
| 形状情報          |                          |
| ・センサタイプ       | (light)                  |
| ・感度           | (r, theta)               |
| 位置情報          |                          |
| ・位置           | (x, y, z)                |
| ・回転角度         | (alpha, beta, gamma)     |

5. 物理エンジン

物理エンジンとは、与えられた形状情報、物理情報を元に、物理法則に従って各時間における物体の挙動をシミュレートするライブラリである。本ソフトウェアでは、AGEIA 社が開発した PhysX ライブラリを使用している。PhysX ライブラリは、物体の衝突検出や、剛体力学、柔体力学、流体力学に則ったシミュレーションをサポートしている。

6. ニューラルコントローラ

本研究では、人工ニューラルネットワーク (Artificial Neural Network) をコントローラとして採用し、物体のモーションを生成する。このニューラルコントローラは、3 節で述べた物理モデルと連動しており、物理モデルに光センサが取り付けられると、そのセンサからの情報がニューラルネットワークに入力信号として伝えられる。センサからの入力情報 *Input* は光の強度 *light*、光源とセンサ位置の差  $L_x, L_y, L_z$  として式 (1) で計算される。

ニューラルネットワークは Fig.2 に示すような構造を持っており、出力された信号は物理モデルの各関節へ伝わり、モーションを生成する。各関節はサイン波形状に振動すると仮定し、この運動に要するトルクは振幅  $R_i$ 、角速度  $\omega_i$ 、位相  $\phi_i$  として式 (2) で計算される。

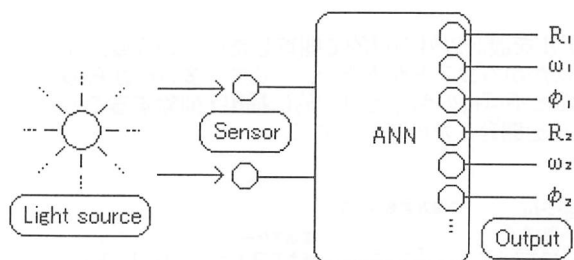


Fig.2 Structure of neural controller

$$\text{Input} = \text{light} \cos \{ \arctan (\sqrt{L_y^2 + L_z^2} / L_x) \} \quad (1)$$

$$\text{Torque}_i = R_i \sin (\omega_i t + \phi_i) \quad (2)$$

7. 最適化アルゴリズム

本研究では、Genetic Algorithm (GA) を用いてニューラルコントローラの各ニューロンを繋ぐ神経の太さ  $W_{ij}$  を最適化する。GA に用いた定数を Table2 に、評価関数を式 (3) に示す。

Table2 Constants for GA

|              |                           |
|--------------|---------------------------|
| 個体の遺伝子表現     | 重みを持った隣接行列 $W$            |
| 個体の形質表現      | 物体と光源の距離 $d$              |
| 個体数          | $n = 30$                  |
| 交叉率          | $\text{cross over} = 0.7$ |
| 突然変異率        | $\text{mutation} = 0.3$   |
| シミュレーションステップ | $\text{step} = 5000$      |
| 初期座標         | (0, 0, 0)                 |
| 光源座標         | (100, 0, 100)             |

$$\text{評価関数 } f(d) = -d \quad (3)$$

$W$  は、行および列に節番号を取った行列であり、各節点を結ぶ神経の太さが格納されている。任意の節点間に経路が存在しない場合は要素が 0 となる。本研究では、シミュレーション終了時に物理モデルが持つセンサと光源の距離が最小となるように学習を行うため、評価関数としてセンサと光源の距離  $d$  に -1 を掛けた値を採用した。神経の太さの幅は、-1 から 1 までの実数で表され、突然変異によって無段階に変化する。また、本実験では、個体の淘汰に際して、優秀な個体をそのまま次の世代へ引き継ぐエリート保存方法を採用した。

8. シミュレーション

任意の物理モデルを作成し、これに対して物理シミュレーションを行った。以下にシミュレーション条件と結果を示す。

8.1 センサを持たない物理モデルのシミュレーション

はじめに、二つの直方体と一つの球を作成し、センサを実装しない状態で物体の挙動をシミュレートした。

Fig.3 (a) に初期状態を、(b)~(d) にシミュレーションの様子を示す。坂に沿ってボールが転がる様子を確認した。

8.2 センサを持った物理モデルのシミュレーション

次に、五つの球を関節ジョイントで接合し、一つの光センサを作成し、ニューラルコントローラによる物体の挙動をシミュレートした。

Fig.4 (a) に初期状態を、(b) ~ (d) にシミュレーションの様子を示す。Fig.4 (a) の左端に接合した小さな球体が光センサである。シミュレーションの結果、芋虫のように地面を這う様子を確認した。

9. 結言

本研究ではモデリングした剛性体に対し、物理法則に従った動作を自動的に生成する物理モデリングソフトウェアを開発し、これを用いてシミュレーションを実施した。

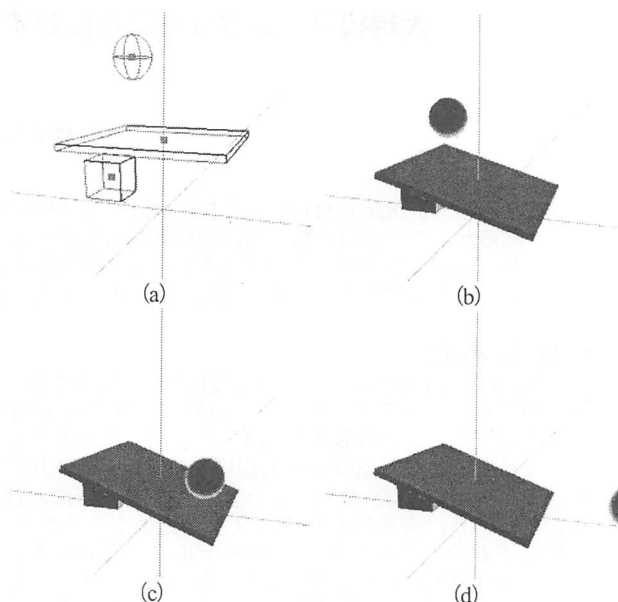


Fig.3 Physical model simulation without sensor

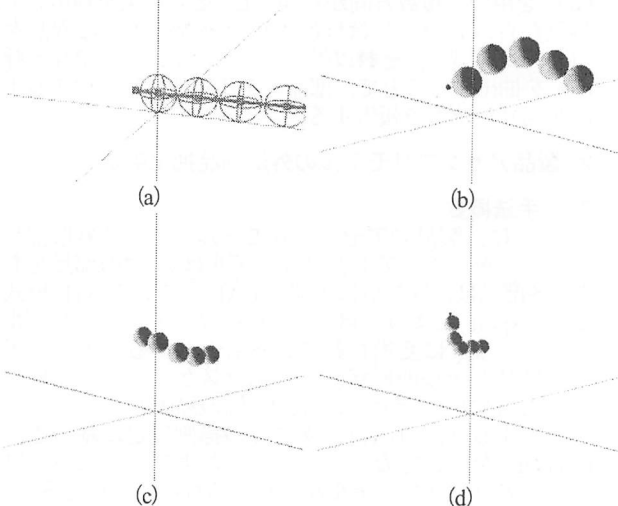


Fig.4 Physical model simulation with sensor

今後の課題として、プリミティブ、ジョイント、センサ種類の拡張、メニューやボタンの配置によるユーザビリティの向上、他のソフトウェアで製作した形状データのインポート、他のソフトウェアへのエクスポート機能の実装が挙げられる。

次に、現在は GA の評価関数として光源との距離のみを考慮しているが、より現実的なシミュレーションを行うためには、他の要素を加える必要があると考えられる。

また、関節の挙動としてサイン波を仮定しているが、この各パラメータの設定には考察の余地があると考えられる。カオスニューラルネットワークの利用によって、サイン波を用いずに任意のモーションを生成することも可能であると考えられる。

参考文献

1) D. Terzopoulos, X. Tu, and R. Grzeszczuk. Artificial Fishes: Autonomous Locomotion, Perception, Behavior, and Learning in a Simulated Physical World. *Artificial Life*, 1(4):327-351, 1994.

2) N. Chaumont, R. Egli, C. Adami. Evolving Virtual Creatures and Catapults. *Artificial Life*, 13(2):139-57, 2007