

## ■2 群 (画像・音・言語) -3 編 (コンピュータグラフィックス)

---

# 2 章 コンピュータグラフィックスアニメーション

(執筆者: 張 英夏・中嶋正之) [2012 年 10 月 受領]

### ■概要■

3 次元コンピュータグラフィックスを用いる利点の一つとして、アニメーションが容易に生成できるという点がある。本章では、アニメーション生成のための基本的な技術について説明する。また、自動的にアニメーションを作成する技術についても説明する。

### 【本章の構成】

2-1 節では、アニメーション生成のための基本的なアルゴリズムについて説明する。2-2 節では、ユーザが動きを指定せず、コンピュータ側で自動的に動きを生成する手法について説明する。

## ■2群 - 3編 - 2章

### 2-1 アニメーションの基礎理論

(執筆者：張 英夏・中嶋正之) [2012年10月 受領]

人間には欠けて見えた部分を無意識に補完する機能が備わっている。例えば、画像 A のある位置にあった物体を少し右にずらした画像 B を作り、A と B を高速に交互に見せると、あたかも A と B が連続的に動いているように見える。このような主観的な運動現象を仮現運動 (Apparent Movement) と呼んでいる。更に、図形を見た後はしばらくその形状情報が目に残る。これを残像効果という。この仮現運動と残像効果を利用したのがアニメーション (Animation) である。つまり、物体の位置や形状、色、透明度、テクスチャなどに変化をつけた複数の画像を時系列に並べ、高速に表示を切り替えることで、あたかも時間経過とともに滑らかに変化しているように見せることが可能である。

従来は画像を 1 枚ずつ作成していた。例えば、粘土の形状を少しずつ変えながら撮影するクレイアニメーションなどがそれである。この方式をコマ撮り (Single Picture Crank) もしくはストップモーションアニメーション (Stop Motion Animation) と呼ぶ。しかしながら、このような手法は当然ながら莫大な時間や労力を要する。そこで登場したのがセルアニメーション (Cel Animation) である。これは、セル板と呼ばれる透明なシートの上に不透明絵の具で彩色することで画像を作成していく。このとき、シーンを物体ごとに、更には物体も顔や手などパーツに分割して別々の透明セルに描く。そうすることによって、物体もしくは物体のパーツに動きが生じたとき、その動いた部分のセルのみを新たに描けばよく、動かない背景などは使い回せばよい。このような仕組みにより、アニメーション制作における労力を減らすことができたが、すべてを手作業で行うのは依然として莫大な労力を要する。そこで近年、作業の一部をコンピュータに行わせ、作業労力を劇的に減らす方法が多く提案されている。本節ではそれらについて述べる。

#### 2-1-1 フレーム間の補間によるアニメーション生成手法

アニメーションの際に表示する画像 1 枚 1 枚をフレーム (Frame) と呼ぶ。特に、フレームの中で動きを表す代表的な画像をキーフレーム (Key Frame) と呼ぶ。コンピュータを用いた補間を行う場合には、このキーフレームは制作者が用意し、キーフレームの中割り画像をコンピュータが作成するという方法が考えられる。これをキーフレーム法 (Key Framing) と呼ぶ。

キーフレーム法の中でも最も簡単な方法が線形キーフレーム法 (Linear Key Framing) である。これは、二つのキーフレームの対応する特徴点の動きを均等に分割する方法である。しかし、少ないキーフレームで特徴点を線形補間すると、動きの軌跡が直線で結ばれるため、不自然なアニメーションとなってしまう。このような理由から非線形なキーフレーム法、例えば曲線を用いた中割り画像の生成が広く用いられている。キーフレーム法はその簡便さから広く利用されているが、複雑な形状の物体の場合、物体内部の特徴点の数が多くなり、特徴点同士の対応点指定作業が煩雑になるという問題がある。

スケルトン法 (Skeleton Method) は物体の形状データに対しスケルトン (Skeleton : 骨格) データを与える。中割りは、スケルトンデータを用いて作成する。最後に各々の中割り画像

において、スケルトンの形状から元物体の形状を復元する。このような方法は、人間のような骨格構造をもつ物体の動きを表すのに適している。

### 2-1-3 運動学を用いたアニメーション生成手法

フォワードキネマティクス (Forward Kinematics : 順運動学) とは、肩やひざなどの各関節の回転角度を直接入力することによってキャラクタを動かす方法である。例えば、手のある位置まで動かしたい場合、肩、肘、手首を順に回転させて望む位置にくるように設定する方法である。ただし、関節ごとの動きの相関などを考慮しながら設定しなければ、全体としてどのような姿勢になるか予測しづらいため、動きの設定が困難であるといえる。

インバースキネマティクス (Inverse Kinematics : 逆運動学) とは、関節の角度ではなく、例えば手先などの末端位置を指定することによって姿勢を指定する手法である。ユーザの指定した末端の位置によって、コンピュータ内部の計算で他の関節の回転角度が計算される。このため、フォワードキネマティクスより直感的に動きを入力することができる。

### 2-1-4 物理シミュレーションに基づくアニメーション生成手法

物理シミュレーションとは、物体のもつ物理的な制約を用いて振る舞いを予測 (シミュレーション) しようという方法である。例えば、アニメータによる制御が困難な流体のアニメーションなどに用いられている。

物理シミュレーションに基づく方法には、大きく分けて格子法と粒子法がある。格子法 (Grid Based Method) は、シミュレーションしたいターゲットを格子状に分割する。そして、各々の格子において対象物体の速度などを計算する方法である。広く利用されている方法として、有限要素法 (FEM : Finite Element Method)、有限体積法 (FVM : Finite Volume Method) などがある。粒子法 (Particle Based Method) はシミュレーションしたいもののある大きさの粒子の集まりで表す。粒子法には SPH (Smoothed Particle Hydrodynamics) 法<sup>1)</sup> と MPS (Moving Particle Semi-implicit) 法<sup>2)</sup> がある。また、格子と粒子を併用する PIC (Particle In Cell) 法<sup>3)</sup> や FLIP (Fluid-Implicit Particle) 法<sup>4)</sup> など存在する。

### 2-1-5 計測されたモーションデータを利用するアニメーション生成手法

キャラクタに対して動きを付けるためには、熟練した感覚と多くの労力を要する。そこで動作計測装置を使って人間の動作データを取得し、利用することでアニメーションを作成することができる。更には、それらのデータをデータベース化して再利用しようという、動作データベースに関する研究が盛んに行われている。有名な動作データベースとしては CMU のモーションキャプチャデータベース<sup>5)</sup> がある。

データベースを利用するための技術として、動作データをキャラクタに違和感なく対応付けするための研究<sup>6)</sup> や、動作と動作の接続部分を滑らかにするための研究<sup>7)</sup> などが広く行われている。

## ■2群 - 3編 - 2章

### 2-2 リアルタイムアニメーション

(執筆著：張 英夏・中嶋正之) [2012年10月 受領]

本節では、実時間でアニメーションを生成する手法について述べる。ゲームなどのインタラクションを前提とした分野では、ユーザの反応に応じてリアルタイムでキャラクターを動かす必要がある。そのために、実時間アニメーション生成に関する研究が数多くなされている<sup>8)~11)</sup>。これらの研究は、GPU (Graphics Processing Unit) の出現とともに一気に加速し、髪の毛<sup>12)</sup>、流体<sup>13)</sup>、水墨画<sup>14)</sup>、群衆シミュレーション<sup>15)</sup>など様々な分野で盛んに研究が行われている。なお、SIGGRAPHでは、2006年からこうしたリアルタイムアニメーションに関するコースを毎年開催しているので、より詳しい内容はそちらを参考にされたい<sup>16)</sup>。

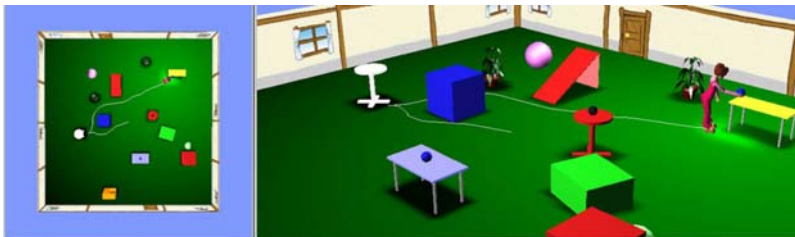
本節では、こうした流れから少し離れ、キャラクターの動きを計算機側でインタラクティブかつ自動的に設定する手法を紹介する。



(a) 初期位置



(b) ボールを取りに行くためのパスとアニメーション



(c) ボールを置きに行くためのパスとアニメーション

図 2・1 ユーザの命令に沿ってキャラクターが自動的に動いている様子

まずは、自律エージェントについて説明する。ここでは、自律エージェントをソフトウェアエージェントに限定しており、より具体的には3次元キャラクタを意味する。船津らはキャラクタエージェントに自然言語で命令を出し、その命令に沿った動きをキャラクタが行うための手法を提案している<sup>17)</sup>。例えば、図2・1(a)に示すようなシーンがあったとする。このとき、ユーザが「白いテーブルの上の青いボールを黄色いテーブルの上に置いて」と命令した場合、システムではキャラクタの現在の位置から白いテーブルまでの最適なパスを計算し、そのパスに沿ってキャラクタを動作させ、ボールを取らせる(図2・1(b))。次に、白いテーブルから黄色いテーブルまでの最適なパスを計算し、そのパスに沿ってキャラクタを歩かせ、ボールを置かせる(図2・1(c))というアニメーションを自動的に生成してくれる。これを可能にするために、テーブルの「前」や「そば」など、自然言語のもつ曖昧さを可能性場として数値化する手法を提案している。

そのほかに、台本を記述することでアニメーションを作ることができるTVML(TV Program Making Language)が提案されている<sup>18)</sup>。これはNHKの林らによって開発されたもので、実際のテレビ番組制作現場で用いられている番組台本の記述方法に似た方法でテレビ番組を作成できる。スクリプトの中にはカメラやライト、撮影環境に関する設定のほかに、3次元キャラクタの発話、動作に関する記述もある。1行がある一つのイベントに対応しているため、時系列順にやらせたい動作、例えば、bow, talk, turn, walkなどを記述していくことでキャラクタを動作させることができる。また現在は、ユーザとのインタラクションが可能になっている<sup>19)</sup>。現在はT2Vという名前の下、HTMLを自動的にTVML台本に変換したり、中国語や韓国語などにも対応するなど、様々な機能が付加されつつあり、無料で配布されている<sup>20)</sup>。

図2・2に例を示す。左側が実際に出来上がったアニメーション、右側がこのアニメーションのためのスクリプトを示している。



図2・2 TVMLの実行情例 (Gotland大学 林 正樹氏提供)

■参考文献

- 1) L. B. Lucy, "Numerical approach to testing the fission hypothesis," *Astronomical Journal*, vol.82, pp.1013-1024, 1977.
- 2) S. Koshizuka and Y. Oka, "Moving particle semi-implicit method for fragmentation of incompressible fluid," *Nuclear Science and Engineering*, vol.123, pp.421-434, 1996.
- 3) F. H. Harlow, "The particle-in-cell method for numerical solution of problems in fluid dynamics," *Methods in Computational Physics*, pp.319-343, 1964.
- 4) J. U. Brackbill and H. M. Ruppel, "FLIP: A method for adaptively zoned, particle-in-cell calculations in two dimensions," *Journal of Computational Physics*, vol.65, no.2, pp.314-343, 1986.
- 5) CMU Graphics Lab. [Online]. <http://mocap.cs.cmu.edu>
- 6) M. Gleicher, "Retargetting motion to new characters," *SIGGRAPH*, pp.33-42, 1998.
- 7) L. Kovar, M. Gleicher, and F. Pighin, "Motion graphs," *SIGGRAPH*, pp.473-482, 2002.
- 8) G. Sannier, S. Balcisoy, N. Thalmann, and D. Thalmann, "VHD: A system for directing real-time virtual actors," *The Visual Computer*, vol.15, pp.320-329, 1999.
- 9) S. R. Musse, "Hierarchical model for real time simulation of virtual human crowds," *Transactions on Visualization and Computer Graphics*, vol.7, no.2, pp.152-164, 2001.
- 10) N. Thalmann, L. Moccozet, G. Sannier, A. Aubel, and D. Thalmann, "Real-time animation of realistic virtual humans," *Computer Graphics and Applications*, vol.18, no.5, pp.42-56, 2002.
- 11) F. Cordier and N. Thalmann, "Real-time animation of dressed virtual humans," *Eurographics*, vol.21, no.3, pp.327-335, 2002.
- 12) S. Tariq and L. Bavoil, "Real time hair simulation and rendering on the GPU," *SIGGRAPH, Talks*, 2008.
- 13) Y. Liu, X. Liu, and E. Wu, "Real-time 3D fluid simulation on GPU with complex obstacles," *Computer Graphics and Applications*, pp.247-256, 2004.
- 14) N. Chu and C. L. Tai, "MoXi: Real-time ink dispersion in absorbent paper," *Transactions on Graphics*, vol.24, no.3, pp.504-511, 2005.
- 15) J. Shopf, C. Oat, and J. Barczak, "GPU crowd simulation," *Siggraph Asia, Technical Sketch*, 2008.
- 16) *Advances in real-time rendering in 3D graphics and games*. [Online]. <http://advances.realtimerendering.com>
- 17) S. Funatsu, T. Koyama, S. Saito, T. Tokunaga, and M. Nakajima, "Action generation from natural language," in *Advances in multimedia information processing*, pp.15-22, 2004.
- 18) M. Hayashi, "TVML (TV Program Making Language)," *SIGGRAPH, Sketches*, 1998.
- 19) 牧野英二, 道家守, 林 正樹, 相川恭寛, "TVML プレイヤーによるインタラクティブアプリケーション," *映像情報メディア学会年次大会*, pp.17-5, 2000.
- 20) T2V. [Online]. <http://t2vlab.jp>