J-34

3Dアニメーション定義における拡張されたLOD（モーションLOD）について

Motion LOD: Extending Level-of-Detail Concept to Computer Character Animation

宮沢 篤† 増山 隆司† 奥澤 和則† 苗村 久美子‡
MIYAZAWA Atsushi MASUYAMA Takashi OKUZAWA Kazunori NAEMURA Kumiko

1．まえがき

人々を，いつの間にかその世界に没入させてしまうコンピュータゲームの映像は，映像表現力，モーション再現，演技力，感情表現の4つを，その構成要件として満たしているといわれる．中でも「モーション再現」は，世界初のポリゴン格闘ゲーム『バーチャファイター』が，（株）セガから1993年12月に発売されて以来，CGによるキャラクターに息吹を与えるゲームデザインの最先端として，プレイヤーからは注目を浴びつづけている．ゲームキャラクターたちのアニメーションには，モーションキャプチャーの技術が欠かせないが，このデータはプログラムで表現するには“非常に密”であり，派生的なモーションに関しては“非常に疎”である，といった問題点を抱えている．これらを解決するには，「時間軸に沿った解像度（resolution along the timeline）」に基づいて，アニメーションの複雑度を任意に変化させる手段が必要不可欠である．

本稿では，3Dアニメーション定義における多重解像度表現を得るための，「モーションLOD」という新しい概念について説明し，また自動的なキーフレーム削減アルゴリズムへの応用などについて述べる．

2．モーションキャプチャーのいくつかの問題

一般のグラフィックス処理において，3D幾何モデリング（3D geometric modeling）に適用される3Dアニメーション定義（3D animation definition）は，

(1) 主にキーフレーミングによるモーションデザイン（motion design），
(2) モーションキャプチャー（motion capture），
(3) 物理シミュレーションなどによるモーション計算（motion computation）

から成り立っていて，特にゲームキャラクターたちのアニメーションには，比較的簡単に人の動きを記録できる，モーションキャプチャーの技術が欠かせない．人間のような複雑な動きは通常，今考えられる最高性能のマシンを使ったとしても，そのまま実時間レートで計算結果をレンダリングすることは不可能であり，あらかじめモーションデータをキャプチャーし，それを必要に応じて再生することで，マシンの負荷を減らしている[1]．

ここでいうモーションデータとは，一定の時間間隔（通常は1/60秒）で与えられたキャラクターの関節の位置や角度情報のことであるが，このデータはプログラムで表現するには“非常に密（very dense）”であり，動きの微妙な点を失わない限りにおいて，データ量を減らす必要がある．また（ほぼ自動的に動きが作り出されることから），派生的なモーションに関しては“非常に疎（very sparse）”であるため，キーフレーミングと比べて細かい調整を行う余地があまりない．リアルで迫力満点のモーションを作るには，訓練されたアーティストの気配りが必要，といった問題点を抱えている．前者については，データ量を減らすために，関数の当てはめやアニメーションカーブのフィルタリングを行うことがあり，これによってより少ないキーフレーム数でもとのアニメーションをできるだけ忠実に再現する．モーションデータを圧縮・伸張することの見方を変えると，モーションデザインの立場からは，従来の手作業によるキーフレームアニメーション技法を，キャプチャーされたモーションデータにも同じように適用するための，数学的な仕組みであるということができる．

3．最適キーフレームの自動決定技術（ATKINS）について

前述のとおり，モーションデータの圧縮とは，与えられたモーションデータからいくつかのキーフレームを選定し，より少ないキーフレーム数でもとのアニメーションをできるだけ忠実に再現することである．このときのキーフレーム選定のための方法論は確立されておらず，従来は実験によって確認していく試行錯誤の繰り返しによる方法（遺伝的アルゴリズムやシミュレーテッド・アニーリング法などのヒューリスティクスを含む）に頼るほかない状況であった．その際，複数のモーションデータを同じパラメータで圧縮するのであるが，望んだ結果が得られなかった場合はその都度パラメータを変更し，すべてのモーションデータの圧縮結果を再確認しなければならず，モーションデータの数が多くなるほどに作業は困難なものとなる．この問題は，本質的には m 次スプライン関数の最適な節点を求める方法であり，以下に述べる最適キーフレームの自動決定技術（ATKINS: AuTomatic Keyframe IdentificatioN Strategy）は，解析的なアプローチによる圧縮方法である．

関数の連続性に関する説明のために，実軸上の片側 $\boldsymbol{R}_+$ で値をもつベキ関数，

$$x_+^m = \begin{cases} x^m, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

を考えることにする．この“右”ベキ関数を使うと，n+1個の節点(x_j, y_j), $j = 0, \ldots, n$をもつ一般の m 次（つまり C^{m-1} 級）スプライン関数は，

† （株）ナムコ インキュベーションセンター 事業開発グループ, NAMCO LIMITED

‡ （株）ナムコ CTカンパニー CT技術環境グループ

$$s(x)=s_{-1}(x)+\sum_{j=0}^{n}c_j(x-x_j)_+^m$$

と，単一の式で表現できることになる [2]．ただし，s_{-1} は区間$[-\infty, x_0]$上の多項式である．圧縮されたモーションデータは，m 次（C^{m-1} 級）スプライン関数を用いて補間する．そこで，補間に用いる m 次スプライン関数の m 階導関数が階段関数（step function）であることに着目する．階段関数とは，各小区間内で値の変動がない関数のことであり，その小区間を単一のスプライン関数によって補間するのである．ただし，モーションデータは離散データであるから，導関数の代わりに差分商を利用し，m+1 階差分商を階段関数で近似することを考える．近似の方法はいくつかありえるが，閾値の設定による，時間的にシーケンシャルな（したがって処理速度が速い）方法を以下に述べる．

(1) 最初のフレームをキーフレームとする，
(2) もとのモーションデータの差分商を計算して，一つ前のキーフレームにおける値との差が，あらかじめ設定された閾値 ε を超えた点を次のキーフレームとする，
(3) 以降，その点を基準にして，データがなくなるまで（2）を繰り返す．ただし，最小の時間間隔で値の変化が ε を超えることがあるので，階段近似する区間の幅の下限 δ をあらかじめ決めておき，その区間を超えない間はキーフレームを選ばないようにする

実用上頻繁に使われるスプライン関数の次数はたかだか 3 次であるから，この場合はモーションデータの 3 階差分商をとり，これを階段近似することによって C^2 級のキーフレームを決定する．さらに 2 階と 1 階の差分商を計算すれば，C^1 級と C^0 級のキーフレームを，分割された小区間について再帰的に決定することができる．

4. むすび

前述の最適キーフレームの自動決定技術（ATKINS）によれば，アニメーションカーブの数学的な連続性に基づいて，3D アニメーションにおける「時間軸に沿った解像度（resolution along the timeline）」を無段階に定義することができ，3D モデルのみを対象にしていた従来からの LOD の概念を，コンピュータキャラクターアニメーションへ拡張することが考えられる（オブジェクトの解像度を選ぶための判定基準には，「視点からの距離」や「画面上のサイズ」以外にも，たとえばオブジェクト間の「相対運動」などが考えられる）．また，もとのモーションデータの特徴を損なわずに，アニメーションの複雑度を任意に変化させる手段は，自動的なキーフレーム削減アルゴリズムへの応用が可能であり，現在アニメーションソフトウェアのプラグインツールとしての実装を通して，モーションデザインなど実際のゲーム制作現場での性能評価を中心に進めている（下図）．

参考文献

1) 宮沢 篤，武田政樹，柳原孝安：コンピュータゲームのテクノロジー【岩波科学ライブラリー】（岩波書店，1999）．
2) 桜井 明，『スプライン関数入門』，東京電機大学出版局，1981.
3) David Luebke et al.: Advanced Issues in Level of Detail, SIGGRAPH 2001 Course Notes 45, August 2001.

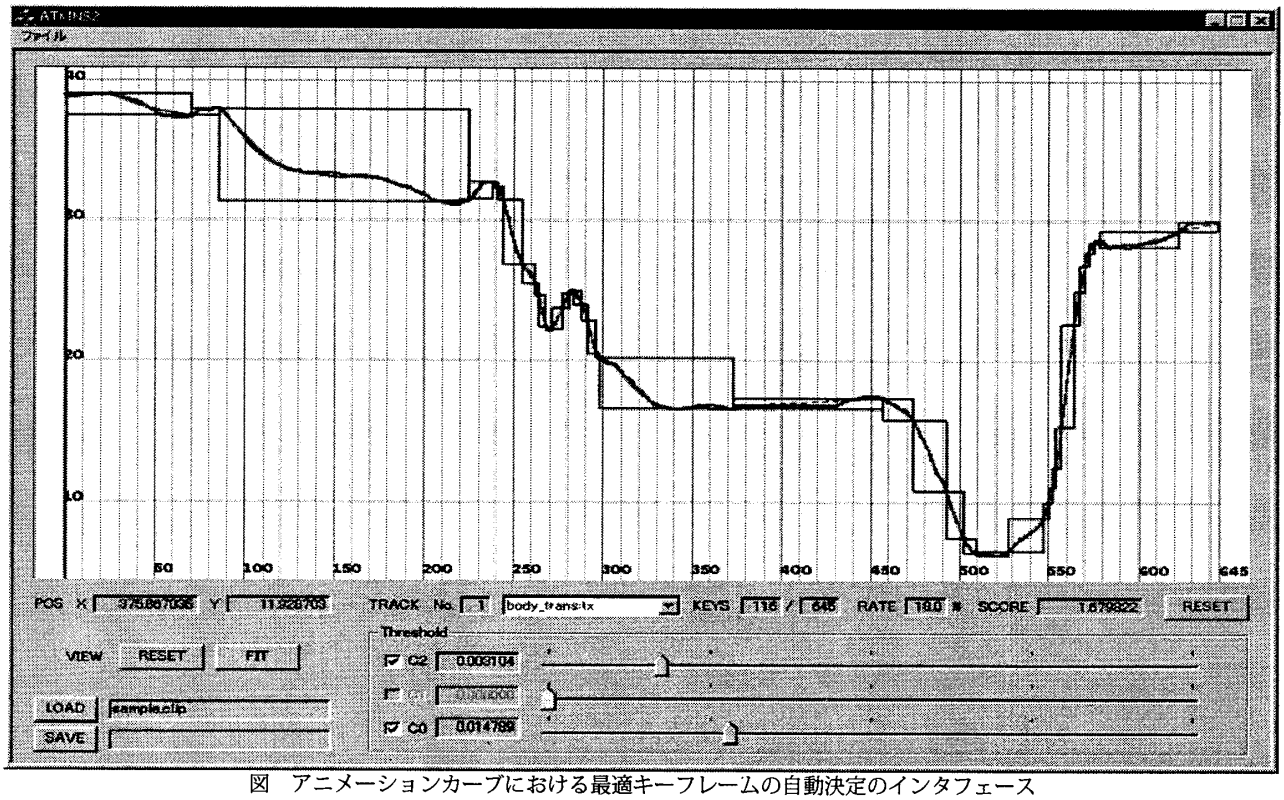

図 アニメーションカーブにおける最適キーフレームの自動決定のインタフェース