

矢印表現による効率的キーフレーム補間手法の一検討
A study on effective keyframe interpolation method by arrow input

今野 瞳[†] 青木 輝勝^{†‡} 沼澤 潤二[‡]

Hitomi Konno[†] Terumasa Aoki^{†‡} Junji Numazawa[‡]

1. はじめに

従来のアニメーションの制作手法は、膨大な費用や手間を必要とし、個人や少人数が短時間で制作することは容易ではなかった。特に3次元コンピュータグラフィックスの分野でその傾向が顕著であり、制作はプロダクションレベルの設備を必要とするため、個人レベルでのアマチュアへの普及がほとんどなされて来なかった。

3次元コンピュータグラフィックスによる動作生成を個人レベルで行うためには、より効率的、直感的、そしてインタラクティブに作業のできる環境が要求される。個人レベルでの動作生成の代表的な手法は線形補間によるキーフレーム法であるが、これは多くの作業工程を必要とし、また直感的な作業ができず、大変非効率的であると言える。本研究では、キャラクターの自発的な動作を、ユーザーに対して視覚的に分かりやすく、かつインタラクティブに生成することを目的とした、新たな補間方法を提案する。

2. 関連研究

線形補間を用いたキーフレーム法のより効率的な動作生成について、これまで幾つかの研究が行われてきた。近藤らの研究では、弾性体に対してキーフレーム法に物理シミュレーションを合わせた手法を提案している^[1]。五十嵐らの研究では、ユーザーによる操作をそのまま実時間で記録して動作生成するための手法として、空間的キーフレーム法を提案している^[2]。これは、キーフレームに加えてユーザーが動かすハンドルによりキャラクターの姿勢を補間する手法をとる。しかし、[1]はキャラクターの動作生成に適しておらず、また[2]はボーン(骨)を使用しないため、人体を模したキャラクターの動作生成を対象としていない。

3. 矢印動作生成手法の提案

3.1 オブジェクトの前提条件

本手法では、オブジェクトの前提条件として、

3次元でモデリング・レンダリングされ、さらに、スケルトン構造をとり、ボーンに駆動可能な関節角が設定されていることを想定する。ユーザーはこのオブジェクトの骨をドラッグ操作で動かすことでポーズを作成し、2つのキーフレームを設定する。これらキーフレームの間を計算機が線形補間して矢印を自動生成し、ユーザーがその矢印を修正という形を取ることで、効率的に非線形的な動作を生成することができる。動作として生成したデータは、1フレーム後コマ画像として再生できる。

ユーザーは作業の都度、部位の骨を選択し(またはあらかじめ選択され)その後その部位に対応する矢印を修正する。視覚的、直感的に分かりやすいよう、線形補間によって自動生成される矢印に以下のような制限を課す。

3.2 矢印に関する規則

- 選択されている部位は、色を変えて表示する。その際ボーンは最前面に不透明で表示され、マッスル部分は半透明の色を乗せて表示される(図1)。
- 矢印の描く動作経路は、選択された部位のボーンの頂点(体の中心から最も遠い点)とする。
- 線形補間は、2つのキーフレームのポーズを設定すると、その差分の、角度による線形補間が自動で計算・表示される。計算は全ての関節において、フォワードキネマティクスが用いられる。
- 線形補間をユーザーが修正する際、曲線表示のためにパラメトリックスプラインによる補間を用いる。制御点には、線形補間時の始点と終点に加えてユーザーが任意に指定する。それらの点により、計算機がスプライン補間を提示し、それが修正曲線になる。修正はx-y平面・y-z平面・z-x平面からの平面的操作のみ可能であり、3つのパラメータを同時に変化させることはできない。

[†] Hitomi Konno · RIEC, Tohoku Univ

[‡] Terumasa Aoki · RIEC, Tohoku Univ

[‡] Junji Numazawa · RIEC, Tohoku Univ

- 矢印の速さを矢印の太さで表現する。速い部分は細く表現する。よって、急に速度を変化させることはできない。ユーザーが矢印をクリックすることで、その部分の速度が変化する。この操作において、加速度は一定とする。
- 速度はユーザーの操作のみで制御され、計算機が作成する線形補間の段階の矢印は速度一定とする。
- 軸回転の場合は、あらかじめ対象となるボーンを選択した後、ボーンの外周部分に弧状の矢印を書くことで表現する。
- 全身移動は、全身を選択するか、またはキャラクターが立っている面 (x-z 平面) に太く平面的な矢印を書くことで表現する。矢印が描く軌道は、スケルトン階層の最上位階層にあたる部位であり、オブジェクトに複数の最上位階層が存在する場合は、もともと y 軸の値が小さい部位とする。

3.3 矢印を作成するための周辺環境

ユーザーが矢印を修正操作する場合の、付加機能について以下に記す。

- 時間軸は、別ウインドウに表示する。フレームを横軸とし、部位ごとに矢印が描かれる。ユーザーがこの矢印の長さを修正することで、動作速度を制御することもできるが、一つの矢印中の速度変化は上記の太さでのみ制御可能で、時間軸ウインドウでは制御できない。
- 視点は、x-y 平面・y-z 平面、z-x 平面から、それぞれの平面について正負からの、6 面からの調整をする。また、ズームイン・ズームアウト機能により、細部の複雑な動作や全身移動などに対応する。

3.4 エラー処理

矢印が実現不可能な位置に書かれた時は、エラーであることを警告し、オブジェクトにあらかじめ付随したボーンの情報に基づき、実現可能な範囲の位置に修正する。修正する点は、ボーンの駆動可能領域のうち、最も修正前の点の近い点である。軸回転できない部位に軸回転矢印が描かれた場合は、エラーの告知のみで矢印は表示しない。修正は、ユーザーによる矢印の修正が終わった段階で、1 フレームごと全部位について確認が行われる。

4. 矢印動作生成手法の評価

提案手法を用いることにより、キーフレームの枚数がどの程度削減できるかの検証を行った。

線形補間による動作生成は、ユーザー側が幾つ

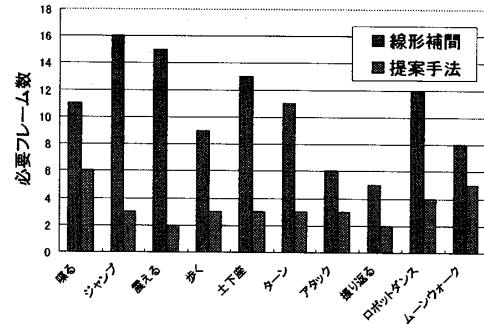


図1: 線形補間と提案手法の必要キーフレームの枚数

かのフレームを作成し、その間を計算機によって自動線形補間される手法を取る。フレームレートは 30fps に設定した。提案手法の方には、線形補間で作成したフレームの内から最低限必要なフレームを選び、そのフレームに矢印を書き込んだ。生成した動作はグラフ中の 10 種類である。

今回作成した全ての動作において、提案手法は線形補間に比べて必要なフレーム数が少ないという結果が得られた。特に、全身運動や円を伴う動作、複数の部位が時間差で動く場合について、この傾向が顕著であった。また、動作時間が長くなるほど、部位が多くなるほど、提案手法の方がより必要なフレーム数が少なくなっている。ターンを例にとると、線形補間では 5 枚のキーフレームを必要とするのに対し、提案手法の場合は、2 枚のキーフレームと矢印による補正のみで生成できるため、線形補間より 3 枚少ないフレーム数で同様の動作を作成できた。

しかし、「喋る」など部位を高速で細かく動かす動作では、線形補間と提案手法に大きな差は見られなかった。

5. 結論

本稿では、キーフレーム法を用いた、効率的、直感的、インタラクティブに動作生成するための新たな補間方法として、矢印による動作生成手法を提案した。実際に動作生成をして必要フレーム数を比較することで、提案手法の効率性を示すことができた。

今後の課題としては、この手法に読み込むオブジェクトを容易に生成するための、自動骨作成技術の研究をしていく必要がある。

参考文献

- [1] 近藤亮、金井崇、安生健一、"物理指向補間による弾性体アニメーションの制御手法"、グラフィクスと CAD / Visual Computing 合同シンポジウム 2005
- [2] 五十嵐健夫、"空間的キーフレーム法によるキャラクターアニメーション"、インタラクティブシステムとソフトウェアに関するワークショップ X (日本ソフトウェア科学会 WISS 2002)