I-052

モーションキャプチャデータの LOD に関する一手法
Level-of-Detail of Motion on Crowd Scene

長谷川 新悟† 西尾 孝治† 小堀 研一†
Shingo Hasegawa Koji Nishio Ken-ichi Kobori

1. はじめに

近年，群集アニメーションはモーションキャプチャデータを用いることによりその品質が向上し，広く利用されるようになってきた．しかし，モーションキャプチャデータは高品質になるとデータ量が増え，実時間でのアニメーション表示が困難となる．そこで本研究ではモーションキャプチャデータを用いた群集アニメーションの高速化手法を提案する．提案手法では関節の運動エネルギーに着目し，エネルギーの小さな関節から段階的に簡略化したモーションキャプチャデータを生成する．また，視点距離に応じて簡略化したモーションキャプチャデータを切り替えて使用する．

2. 関連研究

2.1 アニメーション生成の処理

モーションキャプチャデータを用いて 3DCG キャラクタのアニメーションを行う場合，フレームごとに「姿勢の決定」，「スキニング」，「レンダリング」を行う．「レンダリング」についてはこれまでに様々な方法によって高速化が実現されており，LOD はその代表である．どんなに詳細なモデルでも，視点から遠ざかると詳細部は認識できなくなり，簡略化したモデルに置き換えても違いはほとんど認識できない．LOD は視点距離に応じて詳細なモデルを段階的に簡略化していき，レンダリングコストを削減する技術である．また，近年の GPU の発達によりレンダリングの高速表示が可能となった．以上のように，近年のレンダリングにかかる計算コストは大幅に改善されている．一方,「姿勢の決定」に対しての高速化はほとんど研究されていないのが現状である．しかし，高品質なモーションキャプチャデータの利用によって計算コストは増加傾向にあるため，今後注目される分野であると考えられる．

2.2 姿勢の生成に対する高速化

現在，発表されている姿勢の決定に対しての高速化手法には LOD の考え方が取り入れられており，視点からの距離に応じて利用するモーションキャプチャデータを切り替える．Junghyun らは類似した姿勢が連続するフレームを段階的に統合していき，統合したフレーム間で共通する姿勢データを使用することでデータの呼び出し時間を短縮し，計算コストを削減している[1]．しかし，周期動作を対象にしているため，非周期的で不規則な動作では効果が少ない.

†大阪工業大学

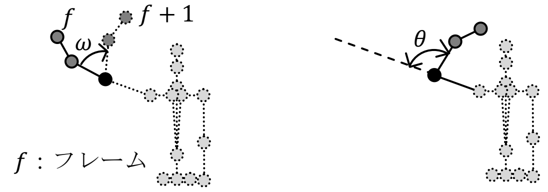
3. 提案手法

3.1 関節の削除

提案手法では既存手法と同様に LOD の考え方を利用し，目立たない動作から簡略化した複数のモーションキャプチャデータを視点距離に応じて入れ替えることで，計算コストを削減し，高速な群集アニメーションを実現する．また，既存手法での問題点である非周期動作にも対応させる．

ここで，モーションキャプチャデータのデータ構造に着目すると，モーションキャプチャデータはツリー構造になっており，ある時点での姿勢を決定するには上位ノードから順に位置と方向を決定していくことがわかる．よって，関節を 1 つでも削除すれば，その関節の位置と方向を決定する処理が不要になり，計算コストが削減できる．

そこで問題となるのが，どの関節から優先的に削除するかである．提案手法では，各関節の動作から関節エネルギーを算出し，エネルギーの低い関節から順に削除していく．関節を大きく回転させると姿勢が大きく変化し，この傾向は対象の関節が親関節に近いほど顕著である．そこで，関節エネルギーの算出には運動エネルギーの法則を利用し，図 1(a)に示すようにフレーム間で関節が大きく回転しているとエネルギーが高くなるようにする．しかし，同図(b)に示すように，フレーム間での関節の回転量が小さくても屈折した状態を維持している関節は大きな動作を行っているといえる．そのため，提案手法では関節の屈折度が閾値よりも大きい場合，運動エネルギーを高くするような重み値を設定する．



(a) フレーム間の回転量　(b) 関節の屈折度

図 1 運動エネルギー

以上のことから，ある関節jの運動エネルギーE_jは式(1), (2), (3)によって求める．ここで，Fは総フレーム数，$I_{j,f}$は慣性モーメント，$\omega_{j,f}$は角速度，$\theta_{j,f}$は屈折度である．また，aは閾値であり，前実験により$a = 0.5\pi$としている．

$$E_j = \frac{1}{F}\sum_{f=0}^{F} e_{j,f} \tag{1}$$

$$e_{j,f} = \frac{1}{2} I_{j,f}\,\omega_{j,f}{}^2 \times W_{j,f} \tag{2}$$

$$W_{j,f} = \frac{a}{\theta_{j,f}} \tag{3}$$

ここで，最小のエネルギーを持つ関節を削除したモーションキャプチャデータを LOD データと定義する．提案手法では LOD データの生成を削除する関節がなくなるまで繰り返す．なお，ルート関節と末端関節は削除の対象外とする．また LOD データには，生成した順にレベル付けしておく．

3.2 LOD データの切り替え

関節エネルギーが低いとき，その関節を削除した LOD データは視点に近い距離でも元のモーションキャプチャデータとほとんど違いのない動作といえる．そこで，削除した関節の関節エネルギーを LOD データに保持させ，この関節エネルギーと視点距離が線形に対応するようにする．

また，視点距離が遠くなるにつれて表示されるキャラクタは小さくなり，ある距離以降はどの動作も認識できなくなる．そこで，視点距離がある距離Lより大きくなる場合，削除可能な関節を全て削除した LOD データを用いることにする．Lの値はキャラクタの大きさと画面の解像度によって決定される．

4. 実験と考察

4.1 実験

提案手法の有効性を検証するために実験を行った．実験には，mocapdata.com のモーションキャプチャデータを用いた．このモーションキャプチャデータは関節数が 24 個あるので，削除対象の関節は末端関節(両手足と頭部)5 個とルート関節を除く 18 個あり，LOD データは Level 18 まで生成されることになる．実験は歩く動作のモーションキャプチャデータで行った．図 2 に実験に用いたモーションキャプチャデータの代表的な姿勢例を示す．また，図 3 に生成した LOD データの一部を示す．なお，実験に使用した PC の CPU は Intel Core i5 2.67GHz，メモリは 4.00GB である．

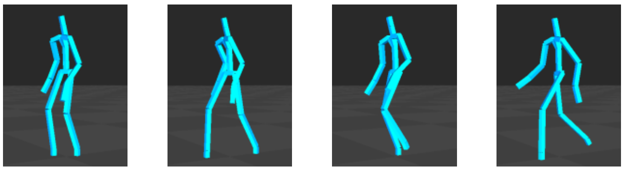
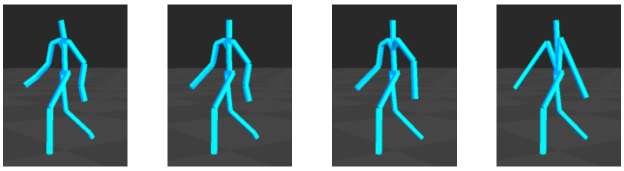


図 2　姿勢例



Level 0　Level 4　Level 8　Level 12

図 3　LOD データ

次に，図 4 に示すように隊列を組んだ群集アニメーションに対して，本手法を適応する前と適用した後での処理速度を比較した．処理速度を表 1 に示す．なお，用いたキャラクタの数は 10,000 体であり，処理速度は姿勢の決定に要する速度のみを示す．

図 4　群集アニメーション

表 1　平均処理速度(ms)

適用前	26.1
適用後	13.2

4.2 考察

図 3 より，レベルが大きくなるにつれて姿勢の変化が大きくなっているが，歩く動作において特徴的な動作を行う脚部は変化していないことから，特徴を残しつつ動作の小さな関節から削除されているといえる．

表 1 より，図 4 に示す群集アニメーションの姿勢の決定に本手法を適応すると，処理速度が約 50%高速化した．このことから，本手法によるアニメーションの高速化が有効であることが確認できた．

しかし，視点の移動によって LOD データが切り替わる瞬間に着目すると，関節エネルギーの低い関節でも変化が認識できてしまうことがある．これは，屈折していた関節が次のレベルで削除され，直線に変わってしまうことが原因であると考えられる．そのため，レベル間で動作の補間を行い，滑らかに LOD データが切り替わるようにする必要があると考えられる．

5. おわりに

本研究ではモーションキャプチャデータを用いた群集アニメーションの高速化手法を提案した．提案手法では，モーションキャプチャデータから各関節のエネルギーを算出し，エネルギーの低い関節から優先的に削除していくことで，目立たない動作から簡略化された複数の LOD データを生成した．実験により，生成した LOD データは特徴を残しつつ動作の細かな関節から簡略化されており，また，高速化を実現出来ていることが確認できた．

今後の課題として，レベル間での動作の補間があげられる．

謝辞

本研究で用いたモーションキャプチャデータは mocapdata.com, Eyes, JAPAN Co. Ltd. により「クリエイティブ・コモンズ表示 2.1 日本ライセンス」の下でライセンスされています。ここに感謝いたします.

参考文献

[1] Junghyun Ahn, Kwangyun Wohn, “Motion level of detail:A simplification method on crowd scene”, Computer Animation and Social Agents, pp.129-137 (2004).