

群行動とビデオ・シンセシスを組み合わせた
アニメーション生成法

Animation method based flocking behavior and video synthesis

岡部 龍太郎[†] 千種 康民[†] 服部 泰造[‡] 富田 光博^{††}

Ryotaro Okabe Yasutami Chigusa Taizoh Hattori Tomita Mitsuhiro

1. はじめに

近年、映画やゲームなどの多くの作品に CG アニメーションが用いられている。特に、現実では撮影が困難な、多くの動物や人が群れをなしている映像を作成する際に用いられる。そして、それらの CG アニメーションの多くは、物体の動きが物理シミュレーションを用いて作成されており、数学的なアルゴリズムに基づいている。しかし、数学的なアルゴリズムでは実際の物体の動きと異なる場合が多い。

そこで、本研究では既存の手法[1]を拡張し、Boids アルゴリズム[2][3]と、既存の手法であるテクスチャ・シンセシス[4]を拡張したビデオ・シンセシス、この2つの技術を統合し、実写動画に基づくアニメーションの自動生成法を提案する。

これにより、実写映像の動物の動きに基づいたアニメーションを作成することで、よりリアルに物体が群行動を行うアニメーションを作成することを実現した。

2. 提案手法

まず、既存の研究[1]では入力動画に前処理を行っていなかったのに対し、本研究では、入力された実写動画から背景差分を用いて移動物体を抽出し、抽出した移動物体の重心を中心とする小領域の動画を作成する。これにより、文献[1]に比べ高速化および遷移可能な点の数の増大を実現した。そして移動物体を抽出した動画の各フレームの移動物体の方向を分析し、記録する。ここで、方向とは移動物体の移動方向ではなく、移動物体の向いている方向を表す。

そして、抽出した動画の数フレームからなるビデオクリップに分割する。そして、同一動画の他ビデオクリップとフレーム毎に RGB の各輝度値の差の絶対値を計算し、それを合計したものをビデオクリップの類似度とする。そして、類似度（形状と色）が高く（誤差の合計が小さい）、方向も近いビデオクリップの対を選出する。ここで選出されたビデオクリップ対は、類似度が高く、方向も近いため相互に入れ替えを行っても違和感小さい。よって、相互に遷移可能となる。そして、遷移可能なビデオクリップを始点と終点とするフレームの集合を、一つのビデオパッチとする。そして、ビデオパッチの方向の遷移を記録する。

次に、Boids アルゴリズムに基づいて k 個の個体による群行動パターンを生成する。Boids アルゴリズムによって生成されるパラメータは、個体の位置、方向、速度があるが、ここでは方向に注目する。各個体に対して、方向の時間遷移と、先ほど抽出したビデオパッチの方向の時間遷移

が最も近くなるようなビデオパッチの組み合わせを算出する。

最後に、算出したビデオパッチの組み合わせを元に、背景画像に移動物体をビデオ・シンセシスを用いて合成する。

その結果、移動経路は Boids アルゴリズムによって生成されたものを用いて、動物の動きは実写動画のものを用いたアニメーションを作成することができる。

以上の方法について、移動物体の抽出と類似度と方向の分析を行う動画分析、群行動パターンの生成を行う Boids アルゴリズム、そして、各ビデオパッチをつなぎ合わせてアニメーションの自動生成を行うビデオ・シンセシスの3つに分けて解説を行う。

3. 動画分析

ここでは、前処理として、実写動画から移動物体のみを抽出し、各フレームについて物体の重心が中心となるような動画の生成を行う。そして、ビデオ・シンセシスを行うのに適している遷移可能なビデオクリップの候補を Distance Matrix を作成して算出し、移動物体の方向からビデオパッチの選出を行う。

3.1 前処理

まず、入力された 1 匹の実写の個体が存在する実写動画 V から移動する個体のみを背景差分法を用いて抽出する。そして各フレームについて移動物体の重心を中心とする動画 U (図 1)を作成する。



図 1 移動物体の重心を中心とした図

既存の研究では、前処理を行っていなかったが前処理を行うことで次の Distance Matrix の生成を行う際に、余分な部分の計算を行わないことで類似度の精度が上がり、計算量を大幅に減らすことが可能となった。

3.2 Distance Matrix の生成

まず、 U を 1 フレームずつずらした f フレーム毎の C_i に分割する。それをビデオクリップとし、 $C_1, C_2, \dots, C_{N-f+1}$ ($C_1=U_1 \dots U_f, C_2=U_2 \dots U_{f+1}$) と表す。

次に、2つの異なるビデオクリップ C_i と C_j がどれくらい似ているかを示すために、距離の概念を導入する。

C_i と C_j の距離が近いときに遷移可能なフレームの候補となり、 C_i と C_j 間がビデオパッチとなる。

そこで、 C_i と C_j の距離を D_{ij} とし (式 1), ビデオクリップのフレームそれぞれについて、RGB の輝度値の差の絶対値を計算したものを求める。

[†] 東京工科大学 Tokyo University of Technology

[‡] 東京国際大学 Tokyo International University

^{††} 日本ダイナミックシステム株式会社

$$D_{ij} = |C_i - C_j| \quad \dots (1)$$

式1の右辺は、式2のようにあらわすことができる。

$$|C_i - C_j| = \sum_{k=0}^{f-1} |U_{i+k} - U_{j+k}| \quad \dots (2)$$

但し、 $|U_{i+k} - U_{j+k}|$ については、画素単位で計算すると式3のようにして求める。

$$|U_{i+k} - U_{j+k}| = \sum_{n=0}^{w-1} \sum_{m=0}^{h-1} \left\{ \sum_{p=0}^2 |U_{i+k}(n, m, p) - U_{j+k}(n, m, p)| \right\} \quad \dots (3)$$

ここで、 w, h はそれぞれ U の幅と高さを表し p は RGB プレーンを表す。

すべての D_{ij} を計算し、距離の近い部分を黒く、遠い部分を白くしたものを Distance Matrix として図2に示す。また、 $D_{ij} < \epsilon$ となる C_i と C_j を遷移可能なビデオクリップとし、そのビデオクリップを黒く表現したものを図3に示す。

これらのビデオクリップ間が、ビデオパッチの候補となる。

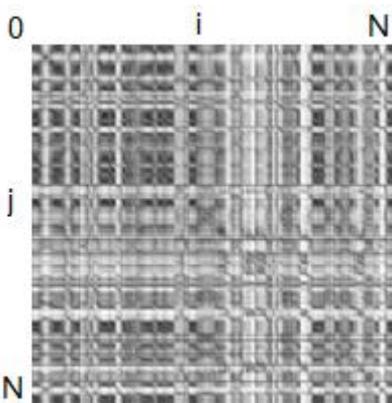


図2 Distance Matrix

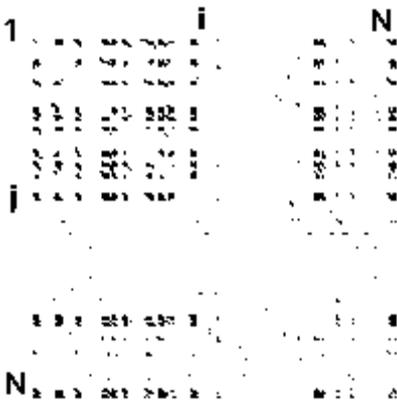


図3 遷移可能なビデオクリップの候補

3.3 角度の算出

3.1 で生成した動画 U の各フレームについて図4の α のように、抽出したオブジェクトの2次中心モーメントをもつ楕円の主軸と、抽出したオブジェクトの重心を通る水平

線との角度を移動物体の方向とする。また、移動物体が、主軸のどちらを向いているかについては前のフレームを参考にする。

そして、その角度の時間遷移を横軸を時間軸（フレーム数）とし、縦軸を角度として図式化したものが図5である。図中の縦に長く線が引かれているところは、 2π から 0 に変化していることを表しているの、瞬時に大きく角度が変化した、ということを表しているのではない。

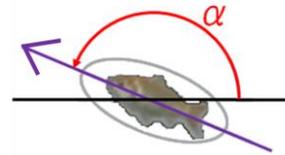


図4 移動物体の方向の算出方法

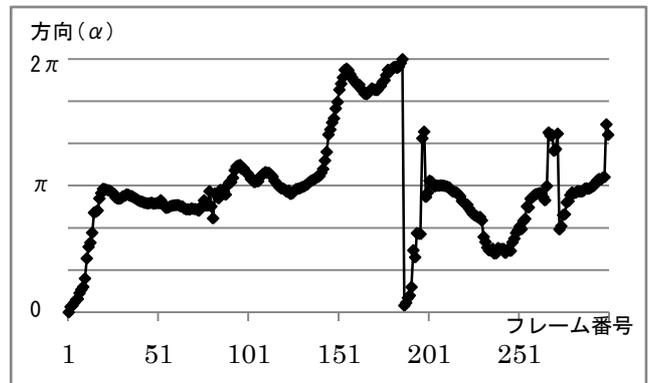


図5 方向 α の時間遷移図

そして、Distance Matrix の遷移可能なビデオクリップの候補について、方向が近く類似度も近いものが遷移可能なビデオクリップとなり、そのビデオクリップ間がビデオパッチとなる。

これらにより、ビデオ・シンセシスを行っても違和感のないビデオクリップが、始点と終点となるビデオパッチが抽出される。

ここで抽出されたビデオクリップは、相互に遷移可能である。よって、相互に遷移可能なビデオクリップに挟まれたビデオパッチは、どのように組み合わせても遷移が不可能になることはなく、無限に遷移を行うことができ、任意の長さの自然な動画を自動生成することが可能になる。

また、ビデオパッチの終点となるビデオクリップが動画 U の最後の遷移可能なビデオクリップでないとき、遷移せずにフレームを進めることも可能であり、ビデオパッチの長さ(フレーム数)と個数を増やすことができる。後述するビデオ・シンセシスを行う際には様々な方向の遷移を持ったビデオパッチが候補にあることが重要である。

4. Boids アルゴリズム

Boids アルゴリズムとは、任意の数の各個体に対して、以下の3つのルールを規定することで鳥や魚の群れをシミュレートするというものである[2][3]。

各個体は、各ルールの適応範囲であり、物体の視野となるスコープをもっており、そのスコープ内の個体から影響

を受ける。そして、各ルールを組み合わせた結果、方向と速度が決定され、その値にしたがって個体が移動する。

それぞれのルールを図 6.1~6.3 に示す。図中の円と矢印はそれぞれ、注目している個体のスコープ、方向と速度を表している。

ルール A 引き離し(Separation)

個体に近づきすぎたらぶつからないようにするルールで、周囲の個体と方向を合わせ、他の個体の前にいれば速度を落とし、後ろにいれば速度を上げる。

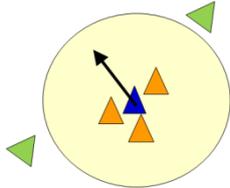


図 6.1 引き離し

ルール B 整列 (Alignment)

スコープ内の他の個体と速度を合わせようとするルールで、周囲の個体と方向と速度を合わせようとする力が働く。

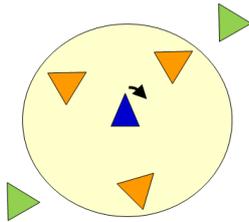


図 6.2 整列

ルール C 結合 (Cohesion)

スコープ内の個体が多くいる方向へ移動するルールで、スコープ内の個体の重心方向へ加速する。結果、群れとしてまとまる力となる。

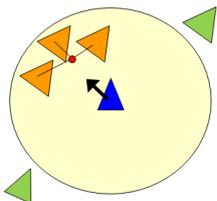


図 6.3 結合

これらの 3 つのルールを用いて各個体の速度を算出する。また、これら 3 つのルールの影響度を変えることにより、様々な種類の群れを作りだすことができる。

ここでは、 n 個の個体での群れを作りだし、各個体について方向の遷移と、速度の遷移を記録する。

そして、次のビデオ・シンセシスでビデオパッチを合成する際に、ここで生成した各個体の移動経路を用いる。

5. ビデオ・シンセシス

ここでは、今まで求めてきたパラメータを利用し、Boids アルゴリズムによって生成された経路を、実写動画の移動物体が違和感なく移動をする動画の生成を行う。

より違和感のない動画を生成するために、1) Boids アルゴリズムによって生成された方向の遷移を元に、ビデオパッチの遷移と比較する。2) 移動経路を Boids アルゴリズムにより生成する。3) 最も近い遷移を行うビデオパッチの組み合わせを選出する。

以上の処理により、より多くのビデオパッチを利用可能にすることで、Boids アルゴリズムと実写動画とのマッチングを図っている。

5.1 ビデオパッチの選出

まず、4 の Boids アルゴリズムで生成された各個体について、方向の遷移を抽出する。そして、生成された各個体について、3.3 で抽出されたビデオパッチの方向の時間遷移をさまざまに組み合わせていき、時間ごとの方向の差の合計が、最も小さくなるようなビデオパッチの組み合わせ順序を算出する。

次に、各個体について最初のビデオパッチのフレーム番号を順番に記録し、ビデオパッチの終点のビデオクリップになったときに、遷移先のビデオパッチのフレーム番号を記録していく。これを、最後のビデオパッチまで繰り返す。

例えば、物体の大きさの変わらない実写動画から図 7.1 のようにビデオパッチ $a \sim h$ の方向の時間変化を抽出した場合について考える。

ビデオパッチの終点の値が、同じ始点となるビデオパッチへ遷移が可能となる。つまり、次のようなことが言える。 a からは b, d, f へ、 b からは c, g へ、 c からは b, d へ、 d からは a, e へ、 e からは b, f へ、 f からは c, g へ、 g からは h へ、 h からは a, e へ、それぞれ遷移可能である。図 7.2 は上記を模式図化したもので、水平方向の矢印がフレームの遷移を表す。その上側の矢印が、過去から未来へ、下側の矢印は、未来から過去への遷移が可能であることを表す。

Boids アルゴリズムによって生成された個体の方向の時間変化が、図 7.3 のようになったとき、図 7.4 のようにビデオパッチを $d \rightarrow e \rightarrow b \rightarrow g \rightarrow h \rightarrow a \rightarrow h$ というように組み合わせることで再現することができる。この際に方向の変化が、完全に一致する必要はなく、抽出されたビデオパッチの組み合わせのうち最も変化が似ているものを選択する。例えば、図 7.5 のような方向の時間変化が生成された場合でも、図 7.6 のように $c \rightarrow f \rightarrow c \rightarrow d \rightarrow a \rightarrow f \rightarrow g$ というビデオパッチの選択をする。このように完全一致を必要としないのは、群行動において、1 個体の影響力があまり大きくない、という理由からである。特に本研究では、群行動パターンを生成した後に各個体の実際の動きを生成しているため、群行動への影響がより少ないものとなる。

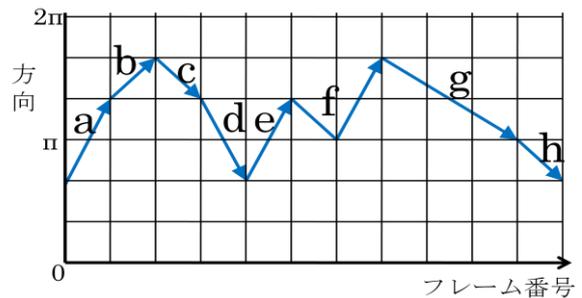


図 7.1 抽出されたビデオパッチ

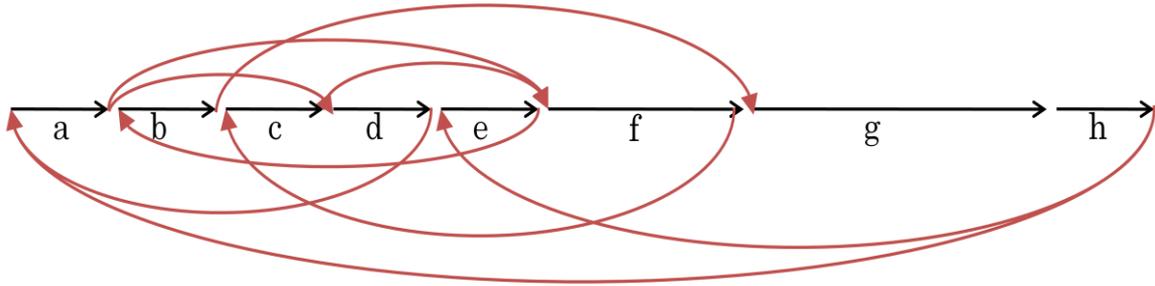


図 7.2 ビデオパッチの遷移図

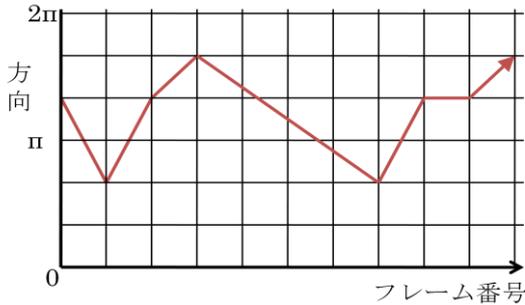


図 7.3 生成された動き(1)

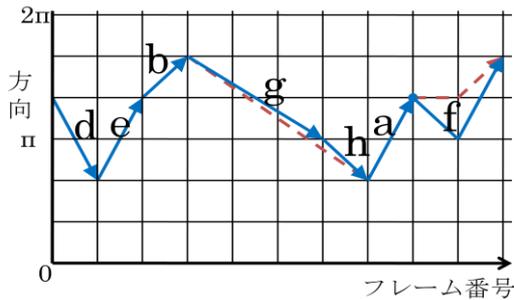


図 7.4 動きに近いビデオパッチの選択(1)

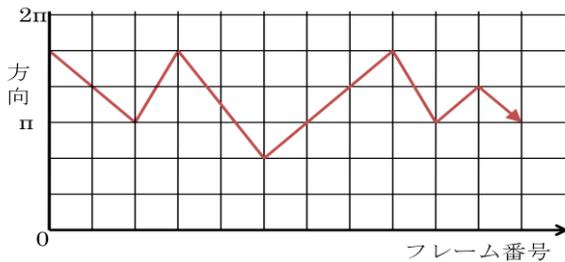


図 7.5 生成された動き(2)

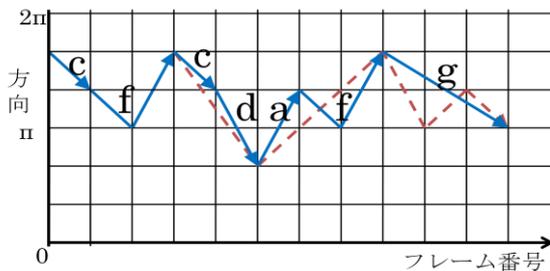


図 7.6 動きに近いビデオパッチの選択(2)

このように、限られた個数のビデオパッチを適切に組み合わせることで、違和感のない遷移が可能になる。さらに、すべてのビデオパッチが遷移可能なビデオクリップで終了しているため、ビデオ・シンセシスを行うことで元の動画よりも長時間の動画を生成することが可能になり、任意のフレーム数の動画を作成することができる。例として、図 7.3 の生成された動画はさらに遷移可能であるので、元の実写動画と異なる動きをする動画をさらに生成することができる。

既存の研究では、移動物体の位置と、Boids アルゴリズムで生成された個体の位置を対応付けている部分があったが、本研究では、位置ではなく方向の遷移を利用することで、出力される動画に合成される個体の連続性が保たれることとなった。

5.2 移動距離を考慮した動画の生成

次に、各個体について記録されたフレームの遷移を元に、移動距離を考慮した動画の生成を行う。

Boids アルゴリズムによって生成されるパラメータは、1 フレームのうちあまり大きく変わらない。そこで、移動距離については実写動画の移動距離を利用するのではなく、Boids アルゴリズムによって生成された 1 フレームの移動距離を用いる。

まず、ビデオクリップを合成するとき、Boids アルゴリズムで生成された速度と、3.3 において算出された方向を利用して、移動距離を算出する。そして、その移動距離を 1 フレーム前の個体の位置に加え、その時の個体の位置を更新する。ここで、方向は実写動画の方向を元としているが、Boids アルゴリズムによって生成された個体の方向の遷移と類似度が最も高く、Boids アルゴリズムによって生成された速度を利用しているため、Boids アルゴリズムによって生成された群行動に近い経路を進むことになる。しかし、個体の動き（魚の場合は泳ぎ方）は実写動画に基づいているため、動きは実写動画と一致する。

そして、5.1 で得られたフレーム番号の個体を、ここで算出した、移動距離を現在のフレームの個体の位置に加えたものに合成していく。これを繰り返すことで違和感のない遷移と合成が可能となる。最後に、生成した動画と背景画像を合成する。

6. 実行結果

水槽の中で金魚が泳いでいる実写動画 1200 フレーム (図 8) を入力動画とし、背景画像を図 9 として本システムを適用した。

まず、動画分析を行うために移動物体のみを抽出し、抽出した物体の重心が中心となるように動画を生成した(図

1). そして、その動画を元に **Distance Matrix** を作成し、遷移可能なビデオクリップの候補を抽出した。(図 3)

そこから、方向の近いものを選び出し、最終的に遷移可能なビデオクリップとして選ばれたものは 72 個となった。

そして、**Boids** アルゴリズムを用いて、20 の個体からなる群行動を 400 フレーム生成した。次に、各個体の方向の時間遷移が、最も近くなるようなビデオパッチの組み合わせを算出し、**Boids** アルゴリズムによって得られた速度の遷移を考慮して、背景画像に合成を行った。

また、図 10 は、計算の対象となる個体がどのようなパラメータをもっているかを確認するために個体数を 4 に減らして実験を行った画像である。図 10 の魚を囲む長方形は 3.1 において抽出した移動物体の領域である。そして、その移動物体を背景上に合成する。ただし、その長方形の中の黒い点が移動物体の重心で、そこから伸びる緑色の線が移動物体の方向を表している。

これらの結果、個体数を 20 と指定した時に出力された動画の 1 フレームを抜き出したものが図 11 と図 12 である。

7. まとめ

本研究では、現実で行うには困難な個体が群行動を行う動画を、**Boids** アルゴリズムによって群行動を生成し、個体の動きは実写動画を基にすることで違和感のない動きを再現した。さらに、ビデオパッチをさまざまに組み合わせることで長時間のアニメーションとして生成する方法を提案し、実例を示した。

これにより、1) 1 個体の細かい実写動画から長時間のリアリティのある動画を生成することを実現、2) 1 個体の動画から群れる多数の個体からなるリアリティの高い動画を生成することを実現した

今後は、方向の誤検出が生じることがあるので、それを減らすために奥行き方向への移動についての対応を考察した画像処理を実現する。そして、出力動画の品質を上げるために、オブジェクト抽出の高品質化、個体の拡大、縮小による遷移可能なビデオクリップの数を増大させる方法を検討していく。

参考文献

- [1] "New video synthesis based on flocking behavior simulation"
G. Colqui, M. Tomita, T. Hattori and Y. Chigusa,
Communications, Control and Signal Processing, 2008. ISCCSP 2008.
3rd International Symposium on Volume , Issue , 12-14 March 2008
pp.936 - 941
- [2] Craig W. Reynolds "Flocks, Herds, and Schools: A Distributed Behavioral Model" In SIGGRAPH 1987 (Computer Graphics 21(4), July 1987, pages 25-34)
- [3] Hartman, Christopher, Benes, Bedrich. "Autonomous boids",
Computer Animation and Virtual Worlds, Volume 17, Numbers 3-4,
July 2006, pp. 199-206.
- [4] A. A. Efros, T. K. Leung. "Texture synthesis by non-parametric sampling." In International Conference on Computer Vision, pages 1033-1038, Corfu, Greece, September 1999.



図 8 入力動画の第 1 フレーム



図 9 背景画像

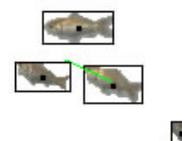


図 10 移動物体の領域と方向



図 11 出力動画(1)



図 12 出力動画(2)