

魚の群れの捕食-被食シーンにおける動作の表現に関する研究

日大生産工(院) 佐藤大輔 日大生産工 吉田典正

1. はじめに

ライオンがシマウマの群れを襲うように、海の中でもカツオがイワシの群れを襲うというような捕食-被食関係によって生態系が成り立っている。実際の魚の群れの映像(例えば、映画 DEEP BLUE(2003)など)を参考にすると、被食者の群れは複数の捕食者によって海面下に追い詰められて捕食され、その過程で被食者の群れは渦を形成する。また、群れ内の各個体は群れ内を頻りに移動し、群れの形は常に変化する。本研究では、これらの点を考慮した魚の動作アルゴリズムを開発し、コンピュータグラフィクスで魚の群れを表現する。また、フラクタルを利用して少ない個体数の動作計算で多くの個体数の群れを表現する手法を提案する。

2. Boid アルゴリズム

Boid アルゴリズム¹⁾²⁾³⁾は群れを成す個体すべてに整列、結合、引離しという共通の単純な3つのルールを与え、各ルールが個体間で相互作用することで群れを形成させるアルゴリズムである。図1は、黒色の個体について Boid アルゴリズムの各ルールの処理を示したものである。整列ルールで仲間の個体の平均ベクトル V_{align} が得られ、結合ルールで仲間の平均位置までのベクトル V_{coh} が得られ、引離しルールで一定距離内に存在する仲間と逆のベクトル V_{sep} が得られ、これら3つのベクトルの和を黒色の個体の移動ベクトルとし、すべての個体において移動ベクトルを求め、各個体を移動させる。本研究の群れの形成において、この Boid アルゴリズムを利用する。

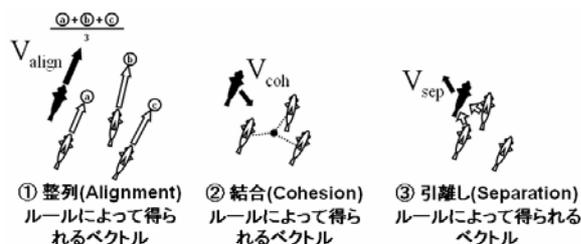


図1 Boid アルゴリズムの各ルールの処理

3. 本研究の概要

本研究では実際の魚の群れの映像を参考に、次の4つの点を考慮した魚の群れの動作アルゴリズムを開発する。

群れ内のどの個体も決まった定位置を維持せず、群れ内を自由に移動し、各個体の移動によって群れの形は常に変化する。イワシなどの被食者の群れは、イルカやサメなどの複数の捕食者によって海面下に追い詰められて捕食される。被食者の群れは海面下に追い詰められる過程で渦を形成する。膨大な個体数の群れを扱う。

4. 群れ内で個体が移動する群れの表現

Boid アルゴリズムで表現する群れでは、群れが一度安定した状態になると群れ内における個体の移動や群れの形が一定となってしまう、実際の映像と比較すると不自然である。

本研究では、図2に示すように各個体に Large-Area, Middle-Area, Small-Area の3つの領域をあらかじめ定め、各領域内の仲間の個体を Boid アルゴリズムの各ルールの処理対象とする。また、各個体に群れ内で移動させるためのベクトル V_{self} を与える。

図3に灰色で示した個体のワンステップの移動ベクトルの計算手法を示す。図3において、灰色の個体は最初に Middle-Area 内の仲間の個体を対象として Boid アルゴリズムの整列ルールを適用し、白色のブロック矢印で示したベクトル V_{align} を得る。次に、各個体が保持しているベクトル V_{self} とベクトル V_{align} を足して黒色のブロック矢印で示したベクトルが得られ、このベクトルを灰色の個体のワンステップでの移動ベクトルとする。ただし、Small-Area 内に個体が存在する場合は引離しイベントが実行され、Small-Area 内の仲間の個体を対象として Boid アルゴリズムの引離しルールが適用され、得られたベクトルが新しい V_{self} として再定義される。また、引離しルールが実行されず、尚且つ Middle-Area

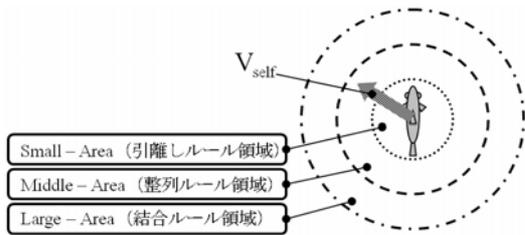


図 2 個体を中心とする3つの領域をBoidアルゴリズムの各ルールの処理対象とする

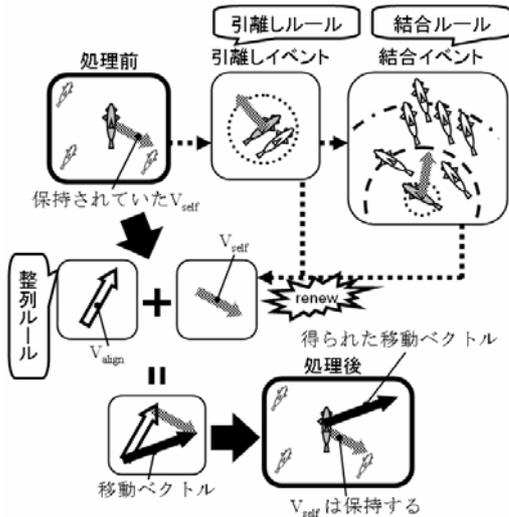


図 3 個体の移動ベクトルの計算手法

内の仲間の個体数が一定値 N を下回る場合には結合イベントが実行され、Large-Area 内の仲間の個体を対象として Boid アルゴリズムの結合ルールが適用され、得られたベクトルが新しい V_{self} として再定義される。

N の値の調整で個体間が密集した群れや、すき間が大きい群れを表現できる。Small-Area の調整で個体間の最接近距離を決定することができる。Middle-Area の調整で各個体の角度や速度が安定した群れや不均一な群れを表現できる。Large-Area の調整で分裂しにくい群れや分裂しやすい群れを表現できる。また、本研究での群れ内で個体が移動する群れの表現は、既存の研究⁴⁾に比べ単純である。

5. 捕食-被食行為の表現

図 4 に示すように、被食者の個体を中心としたある領域内を Predator-Area とし、この領域内に捕食者が存在する場合は捕食回避行為が実行される。捕食回避行為は、群れの動作アルゴリズムの処理を行わず、捕食者の方向と逆の方向に自己の最大スピードで移動させることで表現する。

図 5 に示すように、捕食者には被食

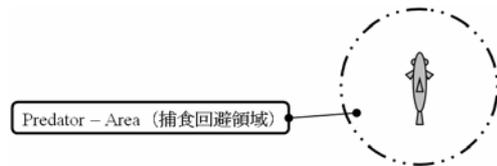


図 4 個体を中心として定めたある領域内に敵が存在すると捕食回避行為を行う



図 5 捕食者には被食者の群れを海面下に追い詰める特性を与え、被食者にはひとつの群れで海中を維持する特性を与える

者の群れの下から上の方で緩やかな突入を繰り返すことで被食者の群れを海面下に追い詰める特性を与え、被食者にはひとつの群れで海中を維持する特性を与える。こうすることで実際の映像のような捕食者と被食者の迫力ある争いを表現する。

6. 被食者の群れの渦の表現

被食者の群れが渦を形成する詳細な理由は明らかになっていないが、本研究では渦が形成される要因を次の3つと仮定する。

- 要因 1: 仲間の動きを素早く察知するために個体間距離を縮める
- 要因 2: 捕食回避行為のための攪乱効果⁵⁾ (ひとつの個体に狙いを定めにくい) を効果的に発揮するために、捕食者に群れが囲まれて行き場を失っても各個体は早い速度を保つ
- 要因 3: なるべく捕食される確率が低く仲間の動きを察知しやすい群れの中央へ向かう

本研究では、個体間距離を縮めて一定速度を保った群れを形成し、各個体の動作アルゴリズムは図 6 に示すように Small-Area 内の仲間の個体の有無によって異なり、仲間の個体が存在する場合は引離しルールと整列ルールを適用して得られた2つのベクトルの和を個体の移動ベクトルとする。また、仲間の個体が存在しない場合は結合ルールを適用して得られたベクトルに自己が持っていたワンステップ前の移動ベクトルを足したベクトルを個体の移動ベクトルとする。この動作アルゴリズムによって渦が形成される様子を図 7 に示す。図 7 (b)の白色の個体は図 6 におけるル

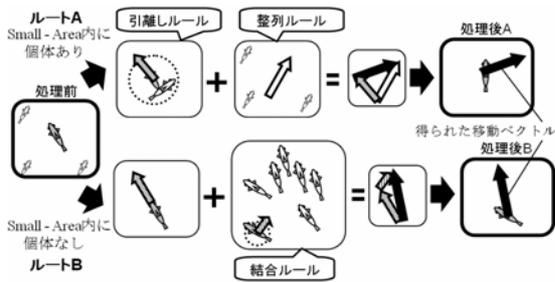


図 6 渦を考慮した群れの動作アルゴリズム

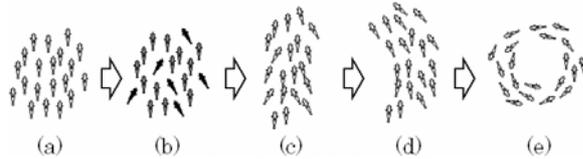


図 7 渦が形成される様子

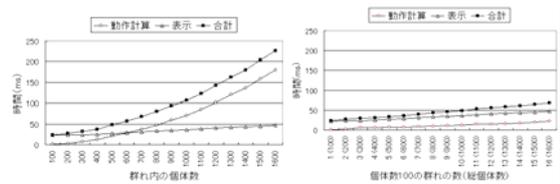
ート A の処理, 黒色の個体はルート B の処理が行われたことを示し, 各個体の動作の繰り返しによって群れが横に潰れた様子を(c)に示し, この潰れた群れが各個体のカオス的な動きによって左右どちらかに傾き, その結果渦が形成される様子を(d)と(e)に示す.

7. 群れ単位の処理の簡略化とフラクタルを利用した群れの表現

別々の群れだった2つの群れが出会い, 群れの結合を考慮する場合を考えると, 各個体は現在における自己の群れ以外の個体の位置についても把握しなければならない. 本研究では, 各個体がアクセスする個体を自己の群れ内の個体と自己の群れと接触可能性のある群れ内の個体だけに限定することで処理の簡略化を行う.

本研究の群れにおいて, ひとつの群れの群れ内の個体数を100から1600まで変化させたときのワンステップの処理時間の変化を図8(a)に示し, 個体数100の群れを互いに影響を及ぼさない場所に複数配置する場合において, 配置する群れの数を1(総個体数100)から16(総個体数1600)まで変化させたときのワンステップの処理時間の変化を図8(b)に示す. 図8(a)より, ひとつの群れ内の個体数を増加させると動作計算時間は個体数の2乗に比例して増加する. これは, 群れ内の各個体が仲間の情報を得るために自分以外の各個体にアクセスするというBoidアルゴリズムの処理によるものである. また, 図8(b)より, 群れの数を増加させると処理時間は群れの数に比例して線形に増加する.

以上のように, 本研究における群れの動作計算時間は群れの数よりも群れ内の個体数に



(a) 群れ内の個体数を増加させたとき (b) 個体数100の群れを増加させたとき
図 8 ワンステップの処理時間のグラフ

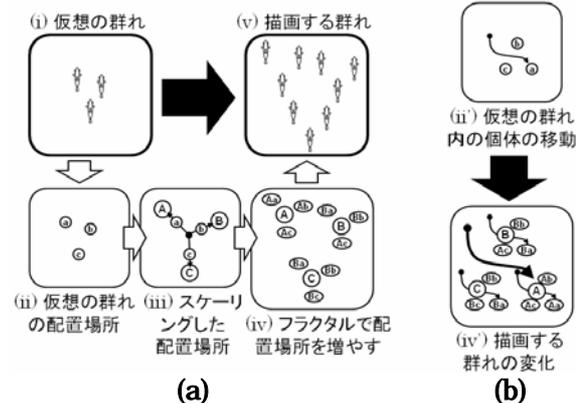


図 9 フラクタルを利用した群れの表現

強く依存し, その個体数の2乗に比例した処理時間が必要となる. そこで, 本研究ではフラクタルを利用して少ない個体数の群れの動作計算で多くの個体数の群れを表現する手法を提案する. 図9(a)の(i)のような仮定の群れを仮定し, その群れ内の各個体の配置場所を $a \cdot b \cdot c$ とし(図9(a)の(ii)), その配置場所をスカラー倍にスケールした配置場所を $A \cdot B \cdot C$ とし(図9(a)の(iii)), $A \cdot B \cdot C$ のそれぞれの配置場所に $a \cdot b \cdot c$ を配置し, 最終的に作成された9つの配置場所(図9(a)の(iv))に個体を描画する. このようにして, 個体数3の群れの動作計算で個体数9の群れを表現する手法を提案する.

また, 仮定する群れ内のある個体の移動が, 実際に描画する群れに及ぼす影響を図9(b)に示す. 図9(b)の(ii')のような仮定内の個体の動きに対し, 実際の個体の動きは図9(b)の(iv')のように, 小さな個体群の移動と各個体群内のひとつの個体の移動という2つの種類の移動を引き起こす.

8. 実行結果

プログラムは Visual Studio .NET 2005 C++を使用した. 実行画面をキャプチャーした画像を図10(a)から(g)に示す. 図10(a)(b)はBoidアルゴリズムの群れと本研究の群れにおける20ステップ後の変化を示し, (c)は複数の捕食者が被食者の群れを海面下に追

い詰める様子，(d)は海面下での捕食行為の様子を示し，(e)は渦が形成される様子を，(f)は渦を斜め上から見た様子を，(g)はフラクタルを利用して個体数 10000 の群れを表現した様子を示す．また，フラクタルの群れのワンステップの処理時間のグラフを図 10 (h) に示す．これより，フラクタルを利用した群れにおける処理時間は，動作計算時間より表示時間が大きくなり，今後，表示を高速化することで膨大な数の群れの表示が期待される．

9. まとめ

実際の魚の群れの映像を参考にして，群れの動作アルゴリズムを開発した．群れ内の個体の移動手法には既存の研究より単純な手法を用い，動きのある生き物らしい群れを表現した．捕食者と被食者に実際の映像の捕食行為と捕食回避行為を参考にして特性を与えることで迫力ある捕食-被食シーンを表現した．また，渦を形成する要因となる被食者の行動を仮定し，渦の形成を試みた結果，実際の渦に似た形は形成されず，改良の余地が残されている．また，フラクタルを利用して動作計

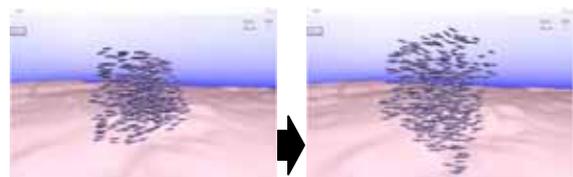
算時間を大幅に短縮した群れの表現手法を提案した．この手法の群れの動作は通常の群れと比較するとやや不自然であるが，カメラから遠く離れた群れや，群れ内におけるカメラから遠い一部分において活用することが可能ではないかと考えられる．

参考文献

- 1) 佐藤，吉田，“捕食者-被食者生態系に基づき個体数の変動する群れアニメーションの作成”，情報処理学会全国大会，67，(2005)，pp.241-242.
- 2) 佐藤，吉田，“魚の群れを対象とした捕食-被食シーンのリアルな表現”，画像電子学会第34回年次大会，(2006)，pp.71-72.
- 3) Craig W. Reynolds，“Flocks, Herds, and Schools: A Distributed Behavioral Model”，Computer Graphics (Proc. SIGGRAPH)，21(4)，Jul. (1987)，pp.25-34.
- 4) 松延，水森，蔡，“自己組織化理論を用いた群れのアニメーション作成”，情報処理学会グラフィクスとCAD研究会，112，(2003)，pp.59-64.
- 5) 佐原，細見，“メダカとヨシ-水辺の健康度をはかる生き物-”，岩波書店，東京，(2003).



(i) あるステップ (ii) 20 ステップ後
(a) Boid アルゴリズムの群れ



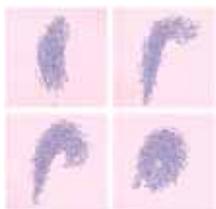
(i) あるステップ (ii) 20 ステップ後
(b) 本研究の群れ



(c) 複数の捕食者が被食者の群れを海面下に追い詰める様子



(d) 海面下で捕食行為が行われる様子



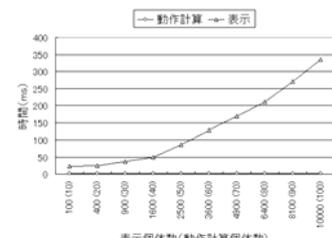
(e) 渦が形成される様子



(f) 渦を斜め上から見た様子



(g) フラクタルを利用して個体数 10000 の群れを表現した様子



(h) フラクタルの群れにおける処理時間のグラフ

図 10 実行画面をキャプチャーした画像