

## 魚の群れの捕食-被食シーンにおける動作のリアルな表現

佐藤大輔\* 吉田典正\*\*

\*日本大学 生産工学研究科 \*\*日本大学 生産工学部

実際の魚の群れの映像を参考にすると、群れ内の各個体は頻繁に移動し群れの形は常に変化する。また、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。本研究では、実際の映像を参考にして魚の群れの動作アルゴリズムを開発し、ユーザが求める振り舞いや実際の魚のような振り舞いをする魚の群れを表現する。群れの形成には Boid アルゴリズムを利用し、群れ内の個体の移動、捕食行為と捕食回避行為、被食者が形成する渦を考慮した魚の動作アルゴリズムの開発を行う。また、実際の映像のような膨大な個体数の魚の群れを表現するために、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。

### Realistic Behavior in the Preying Scene of a Shoal of Fish

Daisuke Sato\* and Norimasa Yoshida\*\*

\*Graduate School of Industrial Technology, Nihon University

\*\* College of Industrial Technology, Nihon University

Each individual in a shoal moves frequently and the shape of the shoal continuously changes. When referring to the preying scene of actual fish, the preys are cornered under the surface of the sea by a number of predators, and preys form a whirlpool. This paper presents an algorithm for realistic behavior for the preying scene of a shoal of fish. We develop boid-based algorithms, such as for the predators cornering the preys under the surface of the sea, and for the preys forming the whirlpool. For expressing a shoal composed of a lot of fish, we propose a fractal-based boid algorithm.

#### 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に捕食-被食関係によって生態系が成り立っている。実際の魚の捕食-被食場面（例えば、映画 DEEP BLUE (2003) など）を参考にすると、ある程度の水深がある領域で捕食者（イルカやカツオなど）が被食者（イワシなど）の群れに単純に突入して捕食行為を行う場面は見受けられず、複数の捕食者が被食者の群れを海面下に追い詰めることで捕食行為が行われる。これは、海中では群れの利点のひとつである攪乱効果[5]（ひとつの個体に狙いを定めにくい効果）が最大限に発揮されるため捕食行為は困難となるが、海面下では海面が壁となり攪乱効果が最大限に発揮されず捕食行為が容易になるためだと考えられる。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し、群れ内の各個体は、決まった定位置を維持せず群れ内を頻繁に移動し、群れの形は常に変化する。

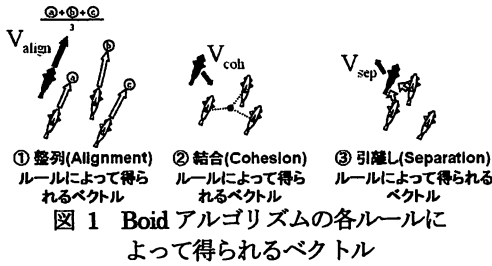
本研究では、実際の魚の群れの映像を参考にして魚の動作アルゴリズムを開発し、各個体に様々な特性を与えることで、ユーザが求める振り舞い

や実際の魚のような振り舞いをする魚の群れを表現する。

群れの形成には Boid アルゴリズム[1,2,3]を利用し、各個体に群れ内を移動するための固有のベクトルを持たせることで群れ内における個体の移動を表現する。これは、既存の研究[4]に比べて単純な手法である。捕食者には被食者の群れを海面下に追い詰める特性を、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とで実際の映像のような迫力ある捕食-被食シーンを表現する。衝突回避の必要がない個体に対して、周辺の個体とベクトルを合わせる処理を行わず群れの中央へ向かう処理だけを行うことで渦の形成を行う。また、従来の研究[1,2]の魚の群れの表現では、個体数 10000 の群れを実時間に近いフレームレートで表現することは処理時間が膨大になり困難であった。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の計算で、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。Boid アルゴリズムは、整列、結合、引離しの3つのルールを用いる。整列ルールでは仲間の進む方向とスピードに合わせる処理を行い、結合ルールでは群れの中心へ向かう処理を行い、引離しルールでは一定距離以上仲間に近づかない処理を行う。

図1は、黒色の個体があるステップでの処理対象とし、白色の個体をその仲間の個体としたときに、Boid アルゴリズムの各ルールによって得られるベクトルを示したものである。整列ルールでは仲間の個体の平均ベクトルである  $V_{align}$  が得られ、結合ルールでは黒色の個体から仲間の平均位置までの単位ベクトルをスカラー倍したベクトル  $V_{coh}$  が得られ、引離しルールではある一定距離内に存在する仲間から黒色の個体までの平均ベクトルの単位ベクトルをスカラー倍したベクトル  $V_{sep}$  が得られる。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する。すべての個体の移動ベクトルを求め、このベクトルで各個体を移動させる。

### 3. 本研究の概要

実際の魚の群れの映像を参考にすると次の4つの点に気づく。本研究ではこれら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。

- ① 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。
- ② 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から捕食行為が行われる。
- ③ 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。
- ④ 群れの個体数は膨大である。

### 4. 群れ内における個体の移動

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  $R_{large}$ 、 $R_{middle}$ 、 $R_{small}$  の3つの距離をあらかじめ定め、個体を中心としてそれぞれの距離を半径とした球の領域をそれぞれ Large-Area、Middle-Area、Small-Area とする。そして、Large-Area 内に存在する仲間の個体を Boid アルゴリズムの結合ルールの処理対象、Middle-Area 内の仲間の個体を整列ルールの処理対象、Small-Area 内の仲間の個体を引離しルールの処理対象とする。また、図3に示すように、各個体を群れ内で移動させるためのベクトル  $V_{self}$  を与える。

図4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4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

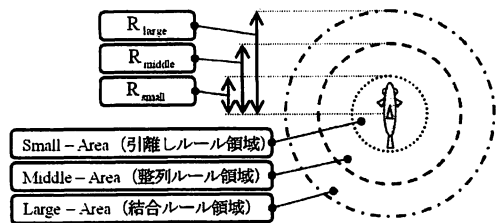


図 2 個体を中心として定めた3つの各領域を各ルールの処理対象とする

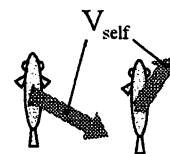


図 3 各個体にベクトル  $V_{self}$  を与える

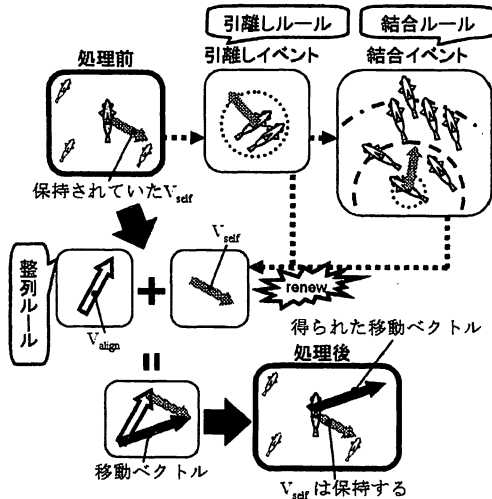


図 4 灰色の個体の移動ベクトル (方向・速度) の計算手法

ールが実行されず、尚且つ Middle-Area 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。

$N$  の値を大きくすると個体間が密集した群れを表現でき、小さくすると個体のすき間が大きい群れを表現できる。 $R_{small}$  の大小によって個体間の最接近距離を決定することができる。 $R_{middle}$  を大きくすると群れ内での各個体の角度や速度において安定した群れを表現でき、小さくすると群れ内で各個体の角度や速度が不均一な群れを表現できる。 $R_{large}$  を大きくすると分裂しにくい群れを表現でき、小さくすると分裂しやすい群れを表現できる。

また、プログラムを実行して一番初めのステップにおける各個体の  $V_{self}$  の与え方は、ランダムなベクトルを与えるか、ゼロベクトルを与えて最初のステップで必ず分離イベントや結合イベントが実行されるような個体の配置を行う。

### 5. 捕食-被食行為の表現

図 5 に示すように、捕食者の個体を中心とした半径  $R_{predator}$  の球の領域内を 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

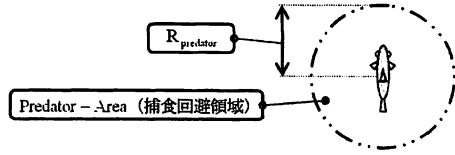


図 5 個体を中心とする半径  $R_{predator}$  の球の領域内を Predator-Area とする

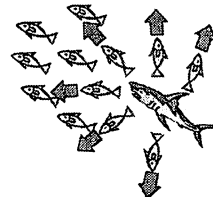


図 6 Predator-Area 内に捕食者が存在する場合、群れの動作アルゴリズムの処理を行わず、捕食回避行為を行う

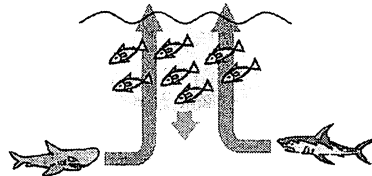


図 7 捕食者には被食者の群れを海面下に追い詰める特性を与え、被食者にはひとつの群れで海中を維持する特性を与える

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 6 に捕食回避行為を行う被食者の群れの様子を示す。

また、図 7 に示すように、捕食者には被食者の群れの下から上の方向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 : ひとつの個体に狙いを定めてくじ攪乱効果[5]を効果的に発揮して捕食回避行為を行うために、捕食者に囲まれて群れ

が行き場を失っても各個体は速い速度を保つ

要因3： 密集した群れ内の各個体において、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

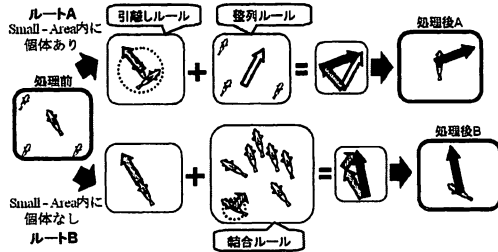


図8 渦の形成を考慮した群れの動作アルゴリズム

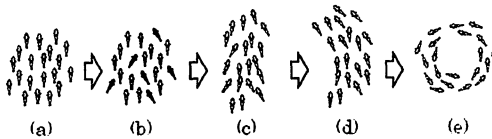


図9 渦が形成される様子

本研究における渦の形成を考慮した群れの表現では、最初に要因1と要因2に示したように個体間距離を縮めて速度を一定速度とする群れを形成する。各個体の動作アルゴリズムは図8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つのベクトルを足したベクトルを個体の移動ベクトルとする。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。この場合は、整列ルールが適用されないで Middle-Area 内の仲間の個体とベクトルを合わせずに、群れの中央へ向かう処理となる。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9示す。図9 (b)の白色の個体は図8におけるルートAの処理を、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

本研究の群れアルゴリズムにおけるひとつの群

れを対象として、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10に示す。図10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、別々の群れだった2つの群れが出会い、群れの結合が行われる場合を考えると、群れどうしが互いに影響を及ぼすか否かの判定を行うために、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自分以外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。本研究の群れアルゴリズムにおいて個体数100の群れを互いに影響を及ぼさない場所に配置する場合において、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11に示す。図11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ことがわかる。

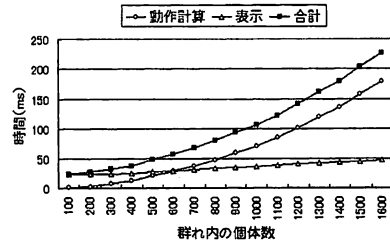


図10 ひとつの群れに関して群れ内の個体数の増加に対するワンステップの処理時間のグラフ

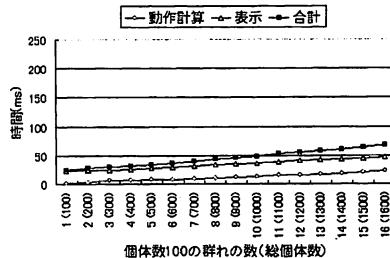


図11 個体数100の群れを互いに影響を及ぼさない場所に配置する場合において、群れの増加に対するワンステップの処理時間のグラフ

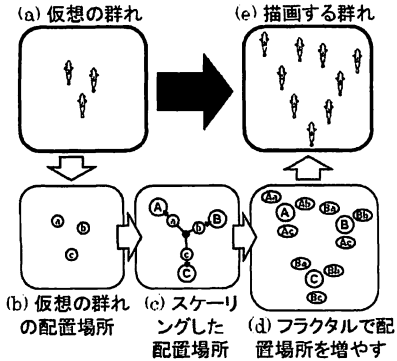


図 12 フラクタルを利用した群れの表現手法

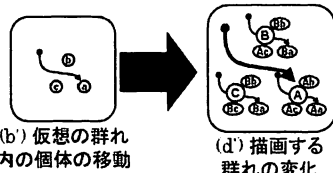


図 13 仮想内の群れにおけるひとつの個体の移動が実際に描画する群れに及ぼす影響

以上のように、動作計算時間は Boid アルゴリズムの処理によって群れの数よりも群れ内における個体数に強く依存し、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を表現する手法を提案する。図 1 2 (a) のような仮想の群れを仮定したとき、その群れ内の各個体の配置場所を  $a \cdot b \cdot c$  とし図 1 2 (b) に示す。また、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 1 2 (c) に示す。次に、A の配置場所に  $a \cdot b \cdot c$  を配置し、配置した場所をそれぞれ  $Aa \cdot Ab \cdot Ac$  とする。これを B と C についても行い、最終的に作成された 9 つの配置場所に個体を描画する。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、仮想する群れ内の個体の移動が、実際に描画する群れに及ぼす影響を図 1 3 に示す。図 1 3 (b) のような仮想内の個体の動きに対し、実際の個体の動きは図 1 3 (d) 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 1 4 に示す。図 1 4 において、(a) は Boid アルゴリズムにおける群れ、(b) は本研究における群れのあるステップとあるステップから 20 ステップ後の様子を示す。(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を、(d) は海面下で捕食される様子を示した。被食者の群れの渦が形成される過程を (e) に示し、(f) は渦を斜め上から見た様子を示す。(g) にはフラクタルを利用することで、個体数 100 の動作計算で個体数 10000 の群れを表現した様子を示す。また、フラクタルの群れに関して群れ内の個体数の増加に対するワンステップの処理時間のグラフを (h) に示す。これより、フラクタルを利用した群れにおけるワンステップの処理時間は、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により膨大な数の群れの表示が期待される。

## 9. まとめ

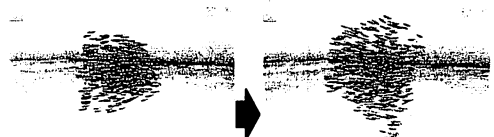
実際の魚の群れの映像を参考にして、群れの動作アルゴリズムを開発した。群れ内の個体の移動手法には既存の研究より単純な手法を用い、動きのある生き物らしい群れを表現した。捕食者と被食者に実際の映像の捕食行為と捕食回避行為を参考にして特性を与えることで迫力ある捕食・被食シーンを表現した。また、渦を形成する要因となる被食者の行動を仮定し、渦の形成を試みた結果、渦は形成されたが実際の渦に似た形は形成されず、改良の余地が残されている。また、フラクタルを利用して動作計算時間を大幅に短縮した群れの表現手法を提案した。この手法の群れの動作は通常の群れと比較するとやや不自然であるが、カメラから遠く離れた群れや、群れ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の様子



(i) あるステップ (ii) 20 ステップ後  
(b) 本研究の群れの様子



(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



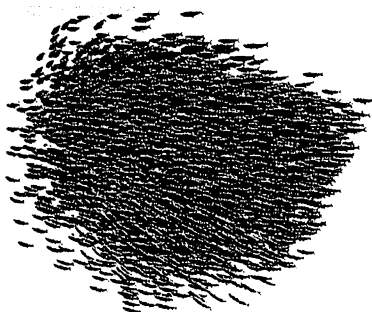
(d) 海面下で捕食行為が行われる様子



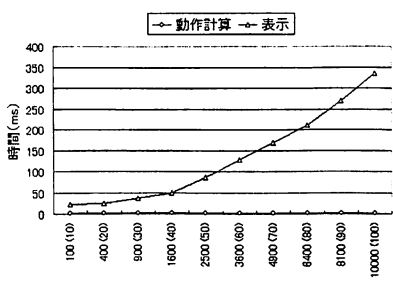
(e) 被食者の群れが渦を形成する様子



(f) 渦を斜め上から見た様子



(g) フラクタルを利用することで個体数 100 の群れの動作計算で個体数 10000 の群れを表現した様子



(h) フラクタルを利用した場合の処理時間

図 14 実行画面のキャプチャーとフラクタルの群れにおける処理時間のグラフ