論 文

論文特集 ■ 最先端画像映像生成技法

ＮＵＲＢＳ, 逆運動学, 協調的運動モデルによる自然感の高い魚の実時間ＣＧ動画像生成

河村貴弘†, 土肥 浩†, 正会員 石塚 満†

Realtime CG Animation of Realistic Fishes using NURBS, Inverse Kinematics and a Co-operative Motion Model

Takahiro Kawamura †, Hiroshi Dohi†, and Mitsuru Ishizuka†

Abstract In this paper, we present a realistic CG(Computer Graphics) animation method to move fishes. These have highly realistic appearance and movement. NURBS(Non-Uniform Rational B-Spline) is employed for both modeling and deformation of CG fishes. This makes the surface data of the fishes very simple to obtain. We can create the very smooth shape of the fishes(one side) by using about 8×4 control points of NURBS. Realistic deformation of the fish is also achieved by forming the control points into several groups and moving them. Inverse kinematics are also incorporated into this deformation. The fishes can swim naturally using a combination of 3 deformation patterns and 3 locomotion patterns. Autonomy of the fishes is realized by a mental model employing random numbers. Both schooling behaviors and collision avoidances are obtained by an algorithm employing detection vector and direction vector. Since the computing costs for this method are not high, we can run the program using general-purpose graphics workstations in realtime.

1. まえがき

ハードウェアレンダリングを高速で行える計算機が開発されたことによって, ＶＲ (Virtual Reality, 仮想現実と訳される) が可能になってきた. これはソフトウェアレンダリングによって得られた画像をテープなどの媒体に記録する, 従来のアニメーション手法では実現できないものである. しかし, これまでのＶＲの多くは, 景観シミュレーションに見られるような静止物体 (建物など) を扱ったものや, 動く物体でも剛体 (車など) を扱ったものであった.

本研究は魚という生物を扱った点で, 従来のＶＲとは一線を画する. オブジェクトには形状のデータと共に, 固有の数学関数を用いた運動モデルを持たせなければならない. ひと足早く, Tu らによる同様な趣旨の論文[1)]が提出されている. Tu らは魚の筋肉のみならず水の粘性をも考慮に入れて, できる限り現実の世界に忠実に魚の動きを実現しようとしている. 本研究では, 必ずしも「現実の世界に則した」魚の動きを実現するというアプローチではなく, 実時間動画像生成が可能な「現実のように思える」魚の動きの実現を図る. ここでは Tu らの手法に比べ簡単な手法を用いて自然感の

キーワード : realtime animation, CG animal, CG fish, co-operative motion model, B-spline
1995 年 3 月 2 日受付, 1995 年 6 月 12 日再受付
† 東京大学工学部電子情報工学科 (〒 113 東京都文京区本郷 7-3-1, TEL 03-3812-2111)
† Dept. of Information and Communication Engineering, Faculty of Engineering, University of Tokyo (7-3-1 Hongo, Bunkyo-ku, Tokyo 113, Japan)

高い魚の動きを実現しており, 実時間ＣＧ(Computer Graphics)において最も重要な計算コストが著しく軽減されている. また写真のテクスチャマップ(モデルに 2 次元画像を貼り付ける手法)を用いて魚に現実感を与えており, アニメータの力を借りることなく短時間で多数の自然感の高い魚を作成することができる.

これまでの著者らによる魚の CG 動画像の実時間生成のシステム[2)~6)] に比較すると, 本システムは魚の形状モデルと動きのモデルの両面で, 自然感の大幅な向上が図られている.

魚のモデリングには NURBS (Non-Uniform Rational B-Spline)を用いている. このことによって魚の形状データは大幅に簡素化され, ウミヘビなどの体長の長い魚を除けば, 8×4 程度の制御点で滑らかな魚の形状(片面)が生成できる. ただし, 本手法で対象とする魚は表面が滑らかなものであり, ハリセンボンやハナミノカサゴのように表面に凹凸がある魚は考慮していない. NURBS などの自由曲面表現形式を用いずに, パッチ曲面(三角形や四角形の集合として表される曲面)によって滑らかな魚の形状を作成しようとすれば, 数百のポリゴンが必要となり, すべての頂点座標を数値入力によって与えることは困難となる. 市販のモデリングシステムを利用すれば頂点の 3 次元配置を視覚的に決定することができるが, パッチ曲面のモデリングにはかなりの時間が必要になるにもかかわらず, 出来上がった形状は滑らかとはいえない場合が多い. また, 形状を表すデータ量も多量かつ複雑になり, デフォメーション(形状変化, 魚の場合はひねり)を容易に実現することは困難となる.

魚のデフォメーションは NURBS の制御点をグループ化した制御点群を, 固有の数学関数に基づいた回転角で魚に垂直な軸に関して回転させることにより実現している. この際, 魚の体長が伸びないように単関節の逆運動学を多段化した手法を適用した. ひねりは自然ひねり, 左ひねり, 右ひねりの３種類を基本とし, ひねりの振幅やピッチはパラメータで調節できる. 魚のグローバルな動きは直進, 左ターン, 右ターンの３種類の泳ぎを基本とし, ひねりと同期させることによって, 自在な魚の動きが実現できるようになっている.

本研究では, 水生生物の間に見られる最も普通の運動様式は, 進行波のように頭部で始まり後方に向かうにつれて振幅を増し波長を減じていく波動形であり, 一連の曲線が体長にそって通るように体形を変えて振り動かす[7)]という説に基づき動きのモデル化を行った. 本手法はサメ, タラのような紡錘形, チョウチョウウオ, カワハギのような側面圧縮形, ウツボ, ウミヘビのような細身形の魚には適用可能であると思われるが, エイやヒラメのような背腹圧縮形の魚やハコフグのように体の硬い魚は, 前述の一般的な運動様式があてはまらないため, 本手法をそのまま適用することはできない. トビウオやハゼのような特殊な泳ぎ方をする魚は対象としていない. デフォメーションの手法では, 複雑な図形(主にパッチ曲面で表される)に対して単純で変形の基準となるスケルトン(骨格)を定義して, 図形をこの骨格との位置関係で定めるスケルトン法を利用し, 多関節構造体の変形として扱うことが多い. しかし, この手法はロボットや節足動物などの表現には向いているが, 筋肉の動きによって表面形状が変化する脊椎動物や軟体動物の表現には向いていない.

Sederberg らによって 3 次元物体を滑らかに変形させる手法である自由形状変形(FFD:Free Form Deformation)が考案されている[8)]. この手法は, 変形させたい 3 次元物体を平行六面体で囲み, 各陵を適当な間隔で分割して出来る格子点を移動して, この格子点で囲まれた空間を変形させて内部にある物体を変形させるものである. しかし, この手法はパッチ曲面や自由曲面で表される任意の図形に対して適用できるが, 現在のハードウェア性能では実時間ＣＧに適用できない. 本手法によるデフォメーションは与えられた形状を変形させるのではなく, モデルの形状とデフォメーションを考慮に入れて制御点の配置し, その制御点 (群) を動かすことにより NURBS 曲面を発生させている点で FFD と異なるが, FFD のようなゴムのような柔らかい変形が実現できる. 本研究におけるデフォメーション手法は必ずしもスケルトン法のように骨格を必要としていないため, 人間の顔の表情など筋肉に基づく形状変化にも適用できると考えられ, 現在研究を進めている.

魚のロコモーション(空間移動)は直進, 左ターン, 右ターンの 3 種類の泳ぎを基本とし, 3 種類のひねりと同期させることによって自在な魚の動きが実現できるようになっている. 現在は一様乱数を利用したメンタルモデルに基づいて魚の自律性(autonomy)を実現しているが, 今後はユーザとのインタラクションを付加する予定である.

本研究では NURBS, テクスチャマップを含めて 3 次元画像生成のためにシリコングラフィックス社(Silicon Graphics)のグラフィックライブラリ(Graphics Library:GL)を利用している.

群れのシミュレーションに関しては Reynolds の論文[9)]が先駆的である. Reynolds の方法は特に魚の群れに限定されたものではなく, 鳥やチョウにも適用でき

る反面, 魚特有の群れ形成をシミュレートしているわけではない. 魚の群れは, 視覚と体側を走る側線という感覚器よって形成されていると考えられている[10]. 魚はこの側線によって, 他の魚が発生する低周波振動や自分が発生する振動の反響を感じとり近接関係を維持しているらしい. 本研究では, この側線に注目し, 魚の向きを表す進行方向ベクトルを一致させ, 魚どうしの間隔を均等にすることによって魚の群れをシミュレートしている.

2. NURBS (Non-Uniform Rational B-Spline, 非一様有理 B スプライン)

自由曲線 (曲面) の数学的表現形式の代表的なものとして, 欧州で発達した Bézier 曲線と米国で発達した B スプライン曲線がある.

Bézier 曲線は「制御」性には優れているが,「接続」には問題があると考えられる. Bézier 曲線セグメントが与えられたすべての制御点位置ベクトルによる凸結合であるのに対し, B スプライン曲線セグメントはその近傍の局所的ないくつかの制御点位置ベクトルによる凸結合で表される点が異なる. このことから, B スプライン曲線は与えられたポリゴン形状に対して, Bézier 曲線に比べてポリゴンの形状変化の特徴がより一層はっきり現れた形状をなしている. したがって, 曲線形状を制御するために, 1 つの制御点位置を変更した場合, その曲線形状への影響は局所に限定される. この性質はモデリングにおいて極めて重要な性質である. 曲線の形状記述として, B スプライン曲線は「制御」と「接続」の点で, ほぼ理想的な特性を持っていると考えることができる. なお, B スプライン曲線セグメントにおいて, ノットベクトルを特別なものに指定すると Bézier 曲線セグメントに一致する.

M階 ($M-1$ 次) の B スプライン曲線上の点を$\vec{P}(t)$, j番目の制御点を$\vec{Q}_j$とすると, 一般に次式で表現できる.

$$\vec{P}(t) = \sum_{j=0}^{n} N_{j,M}(t)\vec{Q}_j \qquad (1)$$

ここで, $N_{j,M}$は j番目の制御点を係数ベクトルとする混ぜ合わせ関数 (B スプライン関数) である. B スプライン関数には「次数」があり, (次数$+1$) の値を「階数」と呼ぶ.

B スプライン曲線は複数のセグメントから構成される. 2 つのセグメントが共有する点を接続点と呼ぶ. B スプライン曲線は実数直線上の部分空間に対して定義される. その空間を定義する点をノットと呼ぶ. ノットに対応する B スプライン曲線上の点は接続点になる. パラメータの節目といえるノットを数値の列として与えるものが「ノットベクトル」である. j番目の制御点$\vec{Q}_j$は混ぜ合わせ関数 $N_{j,M}(t)$ を持つ. $N_{j,M}(t)$ はノット t_jから M区間にまたがる B スプライン関数である. したがって, 制御点の数に階数を加えたものがノットの数になる.

B スプライン曲線において, ノットの値を等間隔に指定して発生する曲線をユニフォーム (uniform) な B スプライン曲線, 不等間隔なノット値を指定して発生する曲線をノンユニフォーム (non-uniform) な B スプライン曲線という.

ユニフォームな B スプライン曲線は接線ベクトルの大きさが一定と考えられるから, 曲率の大きな所ではなかなか曲がり込むことができない. 一方, ノンユニフォームな B スプライン曲線はノット間隔の比で接線ベクトルの大きさを補正できるので, 曲線の形状をコントロールすることが容易である. 同じ制御点からノットベクトルを変更することでいろいろな形状を得られることは, 曲線を作る上で有用である.

Bézier 曲線, B スプライン曲線はいずれも, パラメータに関する通常の多項式として表現される. しかし, 通常の多項式によっては, 工業上非常に重要な曲線である円錐曲線を厳密に表現することができない. 円錐曲線の厳密な数式表現は, パラメータに関する有理多項式によって可能である. そこで開発されたものが NURBS である.

このような汎用性を有する NURBS は専用回路でボード化し易く, 多くのグラフィックスライブラリが NURBS で処理されている. また, 異なる CAD/CAM システム間での形状データの変換が NURBS という形式で行われつつある. NURBS は機能が高い反面, 計算コストが悪いという欠点があるが, 本研究で使用したグラフィックスライブラリでは自由曲線曲面処理に NURBS が採用されており, その処理は高度に最適化されている. また, 将来的には実時間 CG の必要性と NURBS の汎用性から, 多くのグラフィックワークステーション, パーソナルコンピュータで NURBS の処理がボード化され, この欠点は問題とならなくなるであろう. 詳しくは文献[11][12]を参照されたい.

3. 魚のモデリング

NURBS の制御点メッシュを意図する魚の形状となるように配置した (図 1). すなわち, 写真のテクスチャにぴたりと整合する $M \times N (M \geq 4, N \geq 4)$ の制御点を設け, NURBS を用いて滑らかな曲面を生成

し, 魚の片面を作り上げる (図 2). この際, 魚の厚み (Z方向の値) は図鑑等で他方向から撮影した写真像などを参考にして, 上方から見た魚の像をスケッチするなどして常識的な判断の元で決定する. 同様な方法で表と面対称となるように魚の裏側の面も作り, 一匹の魚が出来上がる (図 3). この際, 制御点メッシュの端点である $2M + 2N - 4$ 個の制御点を自由曲面を通るように, メッシュの u, v 方向のノットベクトルはノットの重複を最初と最後の点で NURBS の階数と等しくする. その結果, 片面ずつ生成した魚の形状は隙間をあけることなく接合される. 例えば, 8×4 の制御点メッシュで NURBS の階数が 4 のとき各ノットベクトルは, 以下のようになる.

$$\vec{u} = [0.0, 0.0, 0.0, 0.0, 0.2, 0.4, 0.6, 0.8, 1.0, 1.0, 1.0, 1.0] \quad (2)$$

$$\vec{v} = [0.0, 0.0, 0.0, 0.0, 1.0, 1.0, 1.0, 1.0] \quad (3)$$

この際に制御点の配置には注意が必要である. 滑らかな曲面を生成するためには, 制御点の間隔ができるだけ均等であることが望ましい. また, モデルのデフォメーションを考慮して, 魚を輪切りにするような形で制御点を配置し M個の制御点群を作る. 現実の魚は非常に多くの脊椎からなる背骨を持ち, 筋肉はその１つ１つの脊椎に対応するように胴体を輪切りにするような形でセグメント化されている. そのため, 魚の表面の動きも胴体を輪切りにするような形で変化していると思われる. それは, 人間が腹筋を曲げた際に胴体を輪切りにするような形でしわができることからも容易に想像できよう. このような理由から, 制御点群は魚を輪切りにするような形で配置するのが良いと考えた. また, 計算コストを軽減し自然なデフォメーションを実現するために, 制御点群はできるだけ等間隔に, 魚の形状の特徴的な箇所を選ぶ. モデリング座標の原点は, 頭から体長の３分の１程度のところが良い. これは魚の後方に向かうにつれて振幅を増す進行波を実現するための工夫である.

また,「胸びれ」などは別オブジェクトとして生成し, NURBS で表現されたボディに結合するのが良い. その際, 魚の動きに同期させて結合角を変えれば「胸びれ」の制御は可能である. 口などのように形状に鋭い角度を持たせたい場合には, NURBS のノットベクトルを調節する. 口の開閉も制御点を移動することにより制御できる.

図 2 では三角形のポリゴンの切り方が一定でないが, 三角形ポリゴンは NURBS 曲線の曲率に基づいて分割数が変化し, 画面上の魚の大きさやモデルの変形

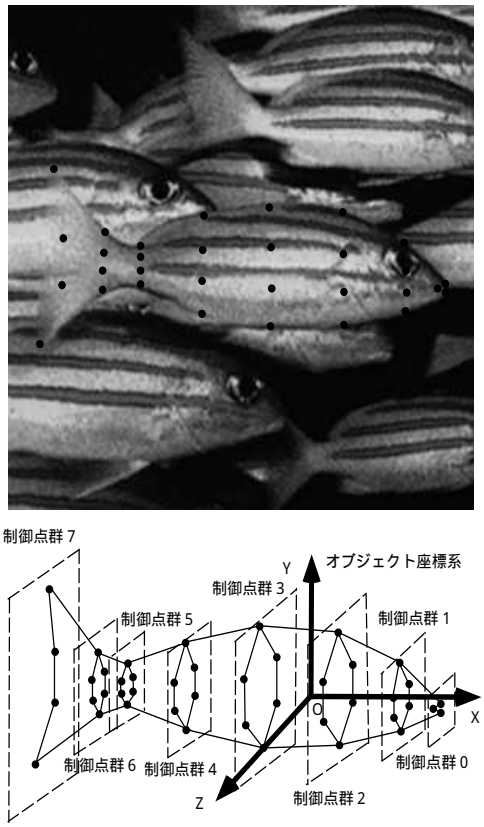

図 1　制御点の取り方の例 (ヨスジフエダイ)
The way of setting control points. (Lutjanus kasumira)

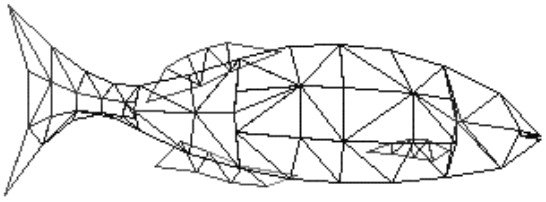

図 2　ヨスジフエダイの形状モデル (片面)
Surface model of Lutjanus kasumira. (one side)

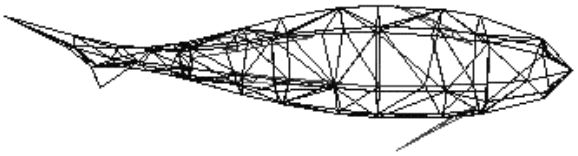

図 3　ヨスジフエダイの形状モデル (両面)
Surface model of Lutjanus kasumira. (both side)

に応じて分割が細かくなったり, 荒くなったりして, 最適化される. 曲面の分割で生成されるポリゴンや曲線を分割 (三角形への分解) して生成される線セグメントのエッジの最大長 (ピクセル単位) を指定できるようになっている. 図 3　での最大長は 50.0 である. これらの機能はグラフィックライブラリで提供されている. この制御点データはデフォメーションの際の制御点でもあり, 形状を規定する制御点でもある. この合理化されたデータの使い方は計算コストの節約に役立っている.

4.　写真のテクスチャマップ

オブジェクトの模様がＣＧ像の現実感, 実在感に与える影響は非常に大きい. 本研究では魚のＣＧ像にはすべて写真のテクスチャマップを施し, 現実感を高めている. これはあたかも 2 次元の魚の写真に厚みを加え, 3 次元の魚を復元するような処理である. また写真のテクスチャマップを行うことで, アニメータの力を借りることなく, 短時間で多数のリアルな魚を作成することができる.

テクスチャマップされた魚のＣＧ像には, 光源からの光に基づいた陰影をつけて立体感を与え, テクスチャの透過率に基づいてひれなどを半透明にしている. 具体的には, NURBS を利用してモデリングされた魚のＣＧ像には白を基調としたスムーズシェード * を施し, そのピクセル値のカラーや透過率とテクスチャの各要素 (テクセルと呼んで前者と区別されることがある) のカラーや透過率を混ぜ合わせることで, 陰影と透明感を有する魚のＣＧ像を生成している.

図 4 はテクスチャマップされた魚の例である. これらの魚はすべて後述するデフォメーション, ロコモーションを行ない, リアルにディスプレイ内で泳ぎ回る.

5.　魚のデフォメーション

自然感の高い魚の動きを実現するためには, デフォメーションとロコモーションが重要な鍵となる. ここでは前者について説明する.

魚の体を走る進行波を実現するためには, 曲線を表す関数を与え, その曲線に魚の背骨が乗るように制御点群を移動しなければならない. 制御点群の間隔が変化しないようにするには曲線の長さを求めなければならないが, 積分方程式を解かなければならず原始関数が求まるのは稀であり, 数値解析では実時間の制約を満たさない. また, 多関節構造の骨格モデルを曲線に載せる方法も, 関節の回転角を求めるのは容易ではない. そこで, 魚のデフォメーションは制御点群を Y 軸に関して回転させることで魚の体を走る近似的な進行波を実現した.

図 4　テクスチャマップされたヨスジフエダイ
Texture mapped CG fish (Lutjanus kasumira)

NURBS では区分的な多項式の組合せで曲線を表現しているため, 局所的に形状を変化させることができる. また, 魚の体長が伸びることのないように単関節の逆運動学 * を多段化する方法を考えた. 本研究では魚のデフォメーションを魚が直進する際の推進力となる自然なひねりを表す「自然ひねり」, 左方向に反転する際のひねりを表す「左ひねり」, 右方向に反転する際のひねりを表す「右ひねり」の 3 種類に分けた.

5.1　単関節逆運動学の多段化

デフォメーションに適用した単関節逆運動学の多段化の原理について記す. 図 5 の三角形 OAB に関して以下の 2 式が成立する.

$$\theta_2' = \arcsin\left\{\frac{r_1}{r_2}\sin(\theta_2-\theta_1)\right\} \qquad (4)$$

$$r_2' = r_1\cos(\theta_2-\theta_1)+r_2\cos\theta_2' \qquad (5)$$

同様にして, 三角形 OBC に関して以下に 2 式が成立する.

$$\theta_3' = \arcsin\left\{\frac{r_2'}{r_3}\sin(\theta_3-\theta_2)\right\} \qquad (6)$$

$$r_3' = r_2'\cos(\theta_3-\theta_2)+r_3\cos\theta_3' \qquad (7)$$

一般に i を正の整数として, i 番目のアームの長さを r_i, アームの先端の X 軸からの回転角 (左回りが正) を θ_i, アームの先端と原点を結んだ線がアームとなす

* 多角形 (主として三角形) 単位で与えられた輝度の情報をもとにして, 陰影の滑らかな変化を表現する各画素点の輝度を推定する方法

* 運動学は, リンク構造を持つ系の関節や先端の位置と関節の回転角との関係を明らかにするものである. 順運動学は, 関節の回転角を使ってアーム先端位置を表す問題である. 逆に, アームの先端位置が与えられたとき, 各関節の回転角を求める問題を逆運動学と呼んでいる.

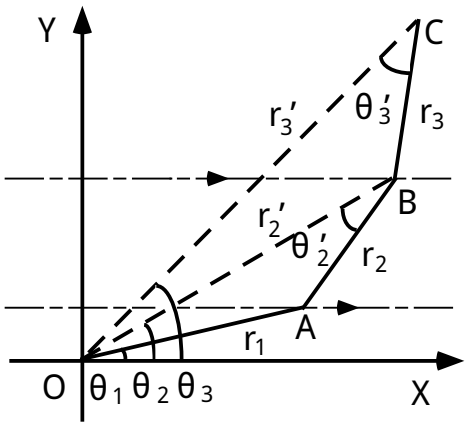

図 5　単関節逆運動学の多段化
Multisatage one-joint inverse kinematics.

角をθ_i'とすると, 以下の漸化式が成立する.

$$\theta_1' = 0,\ r_1' = r_1 \text{とすると}$$

$$\theta_{i+1}' = \arcsin\left\{\frac{r_i'}{r_{i+1}}\sin(\theta_{i+1}-\theta_i)\right\} \quad (8)$$

$$r_{i+1}' = r_i'\cos(\theta_{i+1}-\theta_i) + r_{i+1}\cos\theta_{i+1}' \quad (9)$$

i 番目のアームと $i+1$ 番目のアームの関節の座標を (x_i, y_i) とすると, (x_{i+1}, y_{i+1}) は以下のように表される.

$$x_{i+1} = r_{i+1}'\cos\theta_{i+1} \quad (10)$$

$$= r_{i+1}\cos(\theta_{i+1}+\theta_{i+1}') + r_i'\cos\theta_i \quad (11)$$

$$= r_{i+1}\cos(\theta_{i+1}+\theta_{i+1}') + x_i \quad (12)$$

$$y_{i+1} = r_{i+1}'\sin\theta_{i+1} \quad (13)$$

$$= r_{i+1}\sin(\theta_{i+1}+\theta_{i+1}') + r_i'\sin\theta_i \quad (14)$$

$$= r_{i+1}\sin(\theta_{i+1}+\theta_{i+1}') + y_i \quad (15)$$

これは各変数の正負に影響はされない. だだし, アームが届く範囲でθ_iを指定しなければならないので, θ_i には以下のような制約を設ける.

$$|\theta_{i+1}-\theta_i| \leqq \arctan(\frac{r_{i+1}}{r_i}) \quad (16)$$

この原理は 3 次元にも応用できるが, 本研究ではこの原理を 2 次元に限って適用した. つまり, $i+1$ 番目のアームに付随する制御点群を $(x_i,\ y_i)$ を中心にして, $\theta_{i+1}+\theta_{i+1}'$の角度で回転させることで, 各制御点の座標を求めた.

5.2　自然ひねり

図 6 は魚のモデルをオブジェクト座標系の Y 軸の上方から見たものである. この図が示しているように, 各制御点群をオブジェクト座標系の Y 軸に関して回転 (Y 軸の正方向から見て反時計回りが正方向) させることで, 「自然ひねり」が実現される.

「自然ひねり」を実現する数学関数の例として, ここでは以下の関数を用いた. 制御点群 i の Y 軸に関する回転角をθ_i, ひねりの大きさを制御するパラメータを a, ひねりの速さを制御するパラメータである角速度をω, 時間を t, 制御点群 0 からの位相のずれを d_i して次式から求める.

$$\theta_i = \frac{\pi}{8}a\cdot\sin(\omega t - d_i) \quad (17)$$

ヨスジフエダイの例 (図 6) では, 制御群 i の X 座標の初期値を x_i, 制御点間の間隔を r_i とすると定数は表 1 のようにまとめられる. なお, 制御群の座標はテクスチャ画像の幅が単位長となるように正規化されており, 角速度ωは$\pi/4$, 時間 t はフレームの番号 n として$\pi/4\times n$ で表現した (SGI indy R4000SC 上で 4 frame/sec の割合で表示される場合).　魚のデフォメーションと魚のロコモーションは (17) 式の 2 つパラメータを介して結び付けられる. 自然ひねりの極端な例である図 7 は, 単関節逆運動学の多段化アルゴリズムの有用性を示している.

表 1　自然ひねりにおける定数例
Data example of natural twist.

i	x_i	r_i	d_i
0	0.2466	0.0269	$\pi/8$
1	0.2197	0.1345	$\pi/8$
2	0.0852	0.0852	$\pi/8\times2$
3	-0.0763	-0.0763	$\pi/8\times3$
4	-0.2376	-0.1613	$\pi/8\times4$
5	-0.3990	-0.1614	$\pi/8\times5$
6	-0.5066	-0.1076	$\pi/8\times6$
7	-0.6142	-0.1076	$\pi/8\times7$

図 6　自然ひねり ($a=1$)
Natural twist.

5.3　左ひねりと右ひねり

「左ひねり」も原理は自然ひねりと同じである.「左ひねり」を実現する数学関数として以下の関数を用い

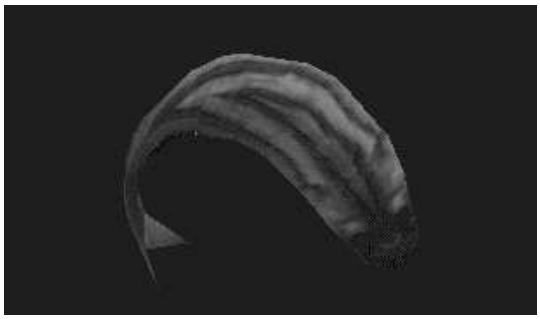

図 7　単関節逆運動学の多段化の適用例 ($a = 3$)
An example of multistage one-joint inverse kinematics.

た.「右ひねり」はこれと対称な関数によっている. 各変数, 定数の意味は「自然ひねり」と同様である.

制御点群 i の X 座標が正のとき

$$\theta_i = \frac{\pi}{6} a \cdot |\sin(\omega t - d_i)| \qquad (18)$$

制御点群 i の X 座標が負のとき

$$\theta_i = -\frac{\pi}{6} a \cdot |\sin(\omega t - d_i)| \qquad (19)$$

魚のロコモーション (速さ, 進行方向) と同期させて, 振幅 a, 角速度ωの 2 つのパラメータを変化させれば自在な魚の動きが実現できる. Tu らが筋肉モデルで実現した魚が反転する動き, 捕食魚から逃げる動きなども表現可能であり, 実現を図っている.

「胸びれ」などのひれは別オブジェクトとして生成し, 魚の胴体に付着させている.「胸ひれ」は付着させる際の角度や向きを変化させることにより制御できる. 口や胸びれの動きについての詳細は省略するが, 適当な数学関数を用いて泳ぎに同期させている.

ヨスジフエダイの例 (図 8) では, 各速度ωを$\pi/2$ とした (SGI indy R4000SC 上). 定数は表 2 のようにまとめられる, 各変数, 定数の意味は「自然ひねり」と同様である.

表 2　左ひねりにおける定数例
Data example of left twist.

i	x_i	r_i	d_i
0	0.2466	0.0269	0
1	0.2197	0.1345	0
2	0.0852	0.0852	$\pi/8$
3	-0.0763	-0.0763	$\pi/8$
4	-0.2376	-0.1613	$\pi/8 \times 2$
5	-0.3990	-0.1614	$\pi/8 \times 3$
6	-0.5066	-0.1076	$\pi/8 \times 4$
7	-0.6142	-0.1076	$\pi/8 \times 5$

図 8　左ひねりのアニメーション (方向転回を加えている)
Animation of left twist. (with the direction change)

6.　魚のロコモーション

ロコモーションはデフォメーションと同様にして, 直進, 左ターン, 右ターンの 3 つの動きを基本とし, オブジェクト座標系で生成された魚をグローバル座標系上で, アフィン変換させることで実現する. 媒質中を運動する物体は, 媒質に対する相対速度が比較的小さいときには, 相対速度に比例する抵抗を受けることが知られている. そのことから以下のような運動方程式が成立すると考えられ, モデル化を行った. ただし, グローバル座標系における魚の位置ベクトルを$\vec{r}$, 魚の質量を

m, 水から受ける抵抗係数を k, 魚の推進力を$\vec{F}_1$, 魚の浮ぶくろに基づく浮力および沈降力を$\vec{F}_2$, 水の流れなどの外乱を$\vec{F}_3$とする.

$\vec{F} = \vec{F}_1 + \vec{F}_2 + \vec{F}_3$として

$$m\frac{d^2\vec{r}}{dt^2} + k\frac{d\vec{r}}{dt} = \vec{F} \qquad (20)$$

ここで, 速度は次フレームまでの魚の移動量を表している, 推進力は (17) ~ (19) 式の振幅 a, 角速度ωに比例する大きさを持ち, 向きは魚の向きと一致する. 前フレームの速度を$\vec{v}$とすれば, 上式は以下のように変形される.

$$\frac{d\vec{r}}{dt} = \frac{\vec{F}}{k} + (\vec{v} - \frac{\vec{F}}{k})exp(-\frac{k}{m}t) \qquad (21)$$

この計算結果から, フレームごとの魚の移動量が決定される.

Tu らの研究では自律的な魚の動きを実現するためにやや複雑な mental state モデルを使っているが, 本研究では一様乱数を用いたごく簡単な意志モデルで自律的な動きを表現した. 今後は情動, 学習などの人工知能分野的な機能の導入を考えている.

ターンについては若干の説明を要する. 図 8 のように方向転回をするときには, 尾を固定したような具合で頭をひねり, ひねりを戻すときには頭を固定したような具合で体をまっすぐにする. これはパラメータ θ_0(頭の角度), θ_7(尾の角度) を用いて, オブジェクト座標系を Y軸に関して回転させることで実現している. 魚は上方, 下方にも方向転回はできるが, この際にはデフォメーションは必要ない考えている.

図 9 は自律的に泳ぎ回る魚の動きをフレームを重ねて表示している.

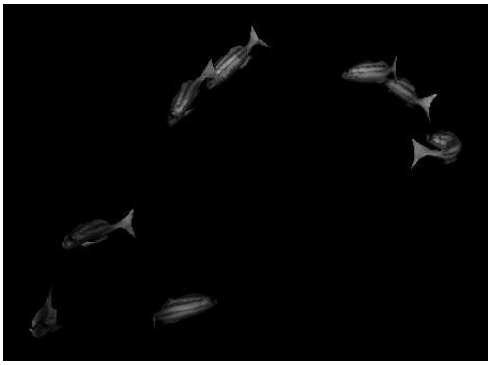

図 9　ヨスジフエダイが自律的に泳ぎ回る様子
The way of fish's autonomous swimming. ($m = 20.0, k = 1.5, \vec{F}_1 = 10.0\vec{d}, |\vec{d}| = 1$)

7.　群れアルゴリズム

群れに関しては Reynolds が先駆的な研究を行っている[9]. Reynolds が提案したアルゴリズムを以下に示す.

(1) 衝突回避：近接する仲間との衝突を回避する
(2) 速度合わせ：近接する仲間と速度を合わせる
(3) 群心力：近接する仲間のそばに留まろうとする

本研究では, Reynolds の衝突回避と群心力の条件を非常にきつくし, 一度群れに引き込まれた魚は速度と間隔をほぼ同じくして泳ぐように設定した. 速度と間隔は自分より前の方にいる魚に合わせる. 前の方にいる魚の判定は以下のように行う. ここで, 自分から相手に向かうベクトルを$\vec{c}$, 魚の進行方向ベクトルを$\vec{d}$とする.

$$\vec{c} \cdot \vec{d} = f \qquad (22)$$

fが正のとき相手は前方にいることになる.

魚が障害物から回避する様子は以下のように実現した. 障害物との距離が境界値よりも近付いたならば, 障害物との距離に基づくパラメータを p として, 次のように新たな進行方向を決める (ただし, すべてのベクトルは正規化されているものとする, 図 **10**) .

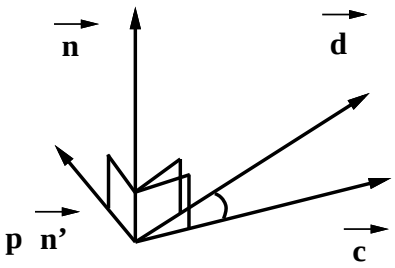

図 10　ベクトルに基づく衝突回避
Collision avoidance based on vectors.

$$\vec{c} \times \vec{d} = \vec{n} \qquad (23)$$

$$\vec{n} \times \vec{d} = \vec{n}' \qquad (24)$$

$$\vec{d} = \vec{d} + p \cdot \vec{n}' \qquad (25)$$

このようにすることで左下の方に障害物があれば, 相対的に右上の方に回避行動をとるようになる. p の取り方で, 障害物が遠くにあればゆっくりと回避し, 近付き過ぎたならば急に回避する様子が実現できる. 障害

物の回避は前述の左ターン, 右ターンをパラメータに基づいて行えば良い.

このアルゴリズムでは, 魚の個体数を N, 障害物数を M とするとき, 次の計算時間が必要になる.

$$O(N^2) + O(MN)$$

これが本アルゴリズムの大きな課題であり, 今後は並列処理などの改良が必要である.

図 11 は群れをなして泳ぐ魚の画像のスナップショットを示している.

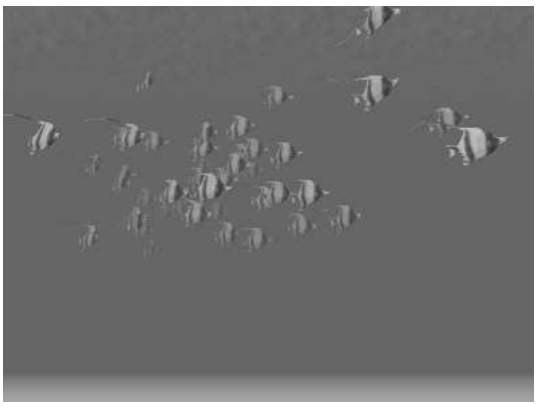

図 11　ツノダシの群れ (SGI Onyx RE2 使用)
Schooling behavior of Zanclus cornutus. (on SGI Onyx RE2)

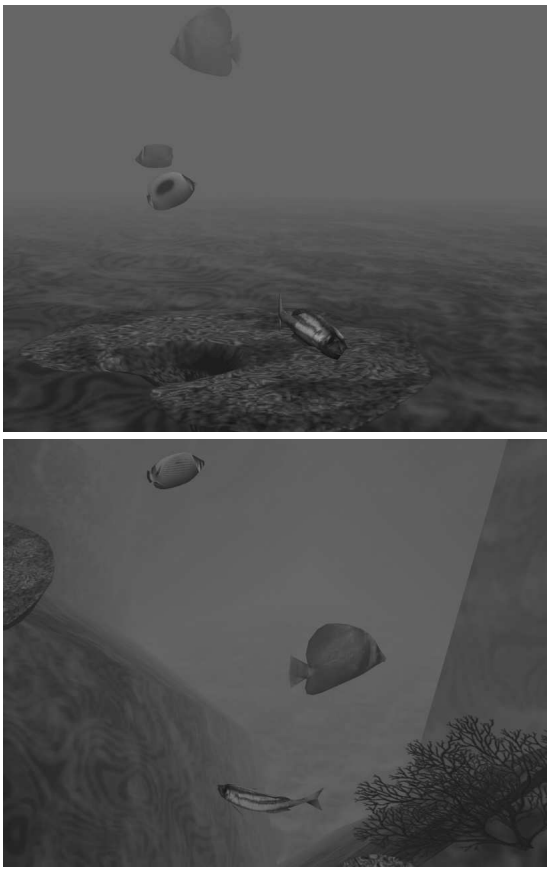

図 12　海中の仮想環境 (SGI Onyx RE2 使用)
Virtual underwater environment.(on SGI Onyx RE2)

8. む　す　び

デフォメーションとロコモーション, 更には群れのアルゴリズムを組み込み, 非常に自然感の高い魚の実時間 CG 画像の生成を実現した. 熱帯魚等の魚は人間に自然のやすらぎを与える重要なオブジェクトであるが, 水槽での飼育は餌や水の交換など大変なので, このような仮想の魚は有用になると思われる. また, 計画されている電子水族館の基盤となる技術でもある. 本システムでは Tu らのように, avoid, escape, school, eat, mate, leave, wander という多くの行動パターンを持たせることが出来ていないが, 今後は考慮したい. 魚一匹 (400×400pixels 程度の大きさ) は, SGI Indy R4000 SC 上で 4 frames/sec で動き, テクスチャマップをしなければ 15 frames/sec で動く. SGI Onyx RE2 上では, ウォークスルーのできる海中の仮想環境も完成しており, 本手法で作成した 20 種に及ぶ魚が泳いでいる様子を見ることができる (図 12).

〔参　考　文　献〕

1) X. Tu and D. Terzopoulos: “Artificial Fishes: Physics, Locomotion, Perception, Behavior”, SIGGRAPH'94, pp.43-50(1994)

2) C.W.Lee, O.Hasegawa, W.Wongwarawipat, H.Dohi and M.Ishizuka: “Realistic Image Synthesis of Deformable Living Thing based of Motion Understanding”, Jour. of Visual Communication and Image Representation(Academic Press), **2**, 4, pp.345-354(1991)

3) M.Fujiki, O.Hasegawa and M.Ishizuka: “A Prototype of Goldfish Software Robot with Real-time Response Function by a Parallel Computer”, Proc. IEEE Int'l Workshop on Robot and Human Communications(ROMAN'92), pp.323-328(1992)

4) 長谷川 修, 藤木 真和, 石塚 満: “指サイン認識による自然感を有する金魚像との実時間インタラクション” , 画像電子学会誌, **23**, 2, pp.86-95(1994)

5) H.Dohi and M.Ishizuka: “Realtime Synthesis of Super-Realistic Moving Goldfish using Small-scale Parallel Transputers”, Communicating with Virtual Worlds (Proc. CGI'93, N.M.Thalmann, D.Thalmann(eds.)), Springer-Verlag, pp.462-472(1993)

6) Y.Yamauchi, H.Dohi and M.Ishizuka: “ A Visual Software Robot with Autonomous Distributed Behav-

ior Model implemented on a Transputer-based Parallel Computer", Transputer/Occam Japan 6(S.Noguchi, M.Ishizuka, M.Ota(eds.)), IOS Press, pp.288-299(1994)
7) 末広 恭雄 監修・訳: "クストー海の百科 5 泳方の秘密", 平凡社 (1975)
8) T. W. Sederberg and S. R. Parry: "Free-Form Deformation of Solid Geometric Models", SIGGRAPH'86, pp.18-22(1986)
9) C.W. Reynolds: "Flocks, Herds, and Schools: A Distributed behavioral Model", SIGGRAPH'87, pp.25-34(1987)
10) 佐々木 忠義 監修, 鈴木 裕 訳: "クストー海の百科 7 見えない信号", 平凡社 (1975)
11) 山口 富士夫 著: "コンピュータディスプレイによる形状処理工学 [I],[II]", 日刊工業新聞社 (1982)
12) 三浦 曜 監修, 中嶋 孝行・大野 敏則 著: "ＮＵＲＢＳ早わかり", 工業調査会 (1994)

河村貴弘 (かわむらたかひろ) 平成 7 年, 東京大学工学部電子情報工学科卒業. 現在, 同大学院情報工学専攻在学中. コンピュータグラフィックスに興味をもつ.

土肥 浩 (どひ ひろし) 昭和 60 年, 慶応義塾大学理工学部電気工学科卒業. 昭和 62 年, 同大学院修士課程修了. 同年, 東大生産技術研究所入所. 平成 5 年, 東大工学部電子情報工学科助手. 擬人化エージェントシステム, 並列処理の研究に従事. 工学修士.

石塚 満 (いしづか みつる) 昭和 46 年, 東京大学工学部電子工学科卒業. 昭和 51 年, 同大学院博士課程修了. 同年, NTT 入社, 横須賀研究所勤務. 昭和 53 年, 東大生産技術研究所助教授. 平成 4 年より工学部電子情報工学科教授. 昭和 55, 56 年に米国パデュー大学客員準教授. 知能工学, 擬人化エージェントシステムの研究に従事. 工学博士. 正会員.