

## 映像特殊効果のための構成的パーティクルシステム

三好雅則\* 栗原恒弥\*\* 安生健一\*

miyoshi@hrl.hitachi.co.jp

\*日立製作所日立研究所 \*\*日立製作所中央研究所

〒319-12 日立市大みか町7-1-1

本稿では、水・炎・煙・爆発など映像特殊効果を表現するための構成的パーティクルシステムを提案する。本システムの特徴は、パーティクルシステムを構成する基本的なオブジェクトを自在に組み合わせることで従来より多彩な特殊効果を表現できることである。本手法を用いて、しぶき・泡・波紋といった水の効果を発生させる実験の結果、多彩な表現を簡単にリアルタイムで実現できることを確認した。

## Constructive Particle System for Special Digital Effects

Masanori Miyoshi\*, Tsuneya Kurihara\*\* and Ken-ichi Anjyo\*

miyoshi@hrl.hitachi.co.jp

\*Hitachi Research Laboratory, Hitachi, Ltd.

\*\* Central Research Laboratory, Hitachi, Ltd.

7-1-1, Omika-cho, Hitachi-shi, Ibaraki-ken, 319-12 Japan

This paper proposes a new method for representing special digital effects, such as dynamics of water, fire, smoke, explosion and so forth. In the method various digital effects are described by the constructive particle system: the object-oriented system of particles with their effectors and sensors. An "interactive" movie was made, where spray, bubbles, ripples and tracks are successfully rendered by the constructive particle system.

## 1. はじめに

コンピュータグラフィックス(CG)による映像特殊効果が、映画・CM・展博・テレビゲームなど娯楽の分野を中心に、盛んに利用されている。爆発・炎・煙・霧などの特殊効果を表現することは、映像に現実感や迫力を持たせるのに欠かせない。CG を利用することで、実際には起こり得ないシーンを再現できる。また、実写の場合に比べてコストや安全性の点でも有利な場合が多い。

従来より、微小な粒子、すなわちパーティクルの集合で、炎、水草、砕ける波、花火などの特殊効果を表現する研究がなされている<sup>1)2)</sup>。また、超並列計算機を使って、パーティクルの動作計算と描画を高速に処理する手法も提案されている<sup>4)</sup>。これらは、表現対象が千差万別であるため、表現対象ごとに専用のプログラムを作成したりする必要があった。また、生成する映像の品質を高くすることに重点を置いており、ゲームなど対話的なシステムには適応できない。

本報告では、基本的なオブジェクトを組み合わせることで多彩な特殊効果の表現を可能とする構成的パーティクルシステムを提案する。また、本手法の応用例として水の効果のリアルタイム表現の例を紹介する。これは、仮想空間中のイルカの動きに応じて、水しぶき、泡、波紋、航跡などを表現するものである。

## 2. 構成的パーティクルシステム

本章では、まず構成的パーティクルシステムの基本構成を説明する。次に、これを構成するオブジェクトを紹介する。そして、特集効果を表現する上で重要な、パーティクルの動作生成方法と描画方法について説明する。

### 2.1 基本構成

初めに、パーティクルの運動の最も簡単な例として図1に示すように水滴が自由落下して水面に到達する現象を考える。水滴は鉛直下方向に重力と、落下を妨げる方向に空気抵抗を受ける。水滴が微小であり質点と見なせるならば、その動きはニュートンの運動方程式に従う。つまり、水滴が受ける力により加速度が生じ自由落下する。水滴はやがて水面に到達して消滅する。

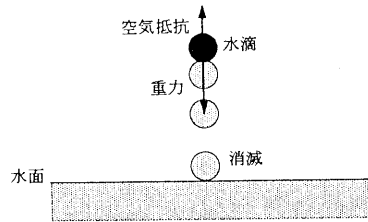


図1 自由落下する水滴

次に、このような運動を一般化することを考える。水滴には上記以外の力、例えば、風力や水滴間の引力も加わることもあるだろう。場合によっては、水滴は水面に達する前に蒸発して消滅するかもしれない。また、水滴を表現する場合と、砂粒を表現する場合には、描画方法も変える必要があるだろう。つまり、表現対象によってパーティクルに対する力の与え方、消滅のさせ方、描画の仕方などを変える必要がある。

そこで、図2に示すように、オブジェクトの組み合わせでパーティクルの運動を表現することにする。パーティクルシステムはパーティクル、エフェクタ、センサで構成される。パーティクルは、水滴などの微小粒子を一般化したものであり、自分自身の描画方法や運動の仕方を知っている。エフェクタは重力や空気抵抗などを一般化したものでパーティクルに力を与える。そして、センサはパーティクルの状態を監視し必要に応じて消滅させる。

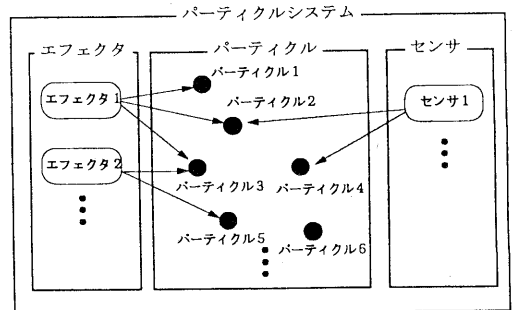


図2 パーティクルシステムの概念

このようなパーティクル、エフェクタ、センサで構成される系のことを、以下では、構成的パーティクルシステムと呼ぶ。

## 2.2 システムの設計

この考えに基づきパーティクルシステムを、オブジェクト指向の分析/設計手法として普及している OMT (Object Modeling Technique)<sup>5)</sup> に従い設計した。結果を図3のオブジェクト図に示す。オブジェクト図とは、オブジェクトの構造を記述するものであり、オブジェクトの集合とそれらの間の依存関係を知ることができる。矩形の箱には上から順に、オブジェクトのクラス名、オブジェクトが保持する属性、オブジェクトが実行する操作を記述してある。

以下、各クラスについて説明する。

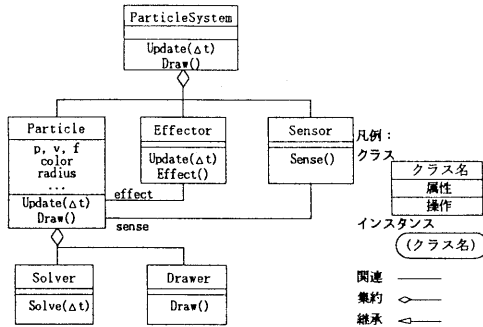


図3 オブジェクト図

### (a) パーティクル(Particle)

パーティクルはパーティクルシステムにおいて微小粒子の役を演じるオブジェクトであり、パーティクルの状態更新を担当するソルバと、描画を担当するドローワで構成される(ソルバとドローワの働きについては2.3、2.4節で述べる)。パーティクルは自分の内部状態を保持し、ソルバに状態更新を委ねることで自動的に自分の状態を更新し、さらにドローワにより自分自身の状態を描画することができる。

パーティクルは自分の現在の状態を表す属性として、位置、速度、外力、色、大きさなどを持つ。

### (b) エフェクタ(Effector)

エフェクタは、パーティクルの現在の状態を変更するオブジェクトである。例えば、重力はパーティクルに対して鉛直下方向に一定の力を与え、パーティクルを動かそうとする。エフェクタはこのようなパーティクルに対する外的な影響をモデル化したものである。3章では、本システムの応用として、水しぶき、泡、波紋などの水の効果を扱うが、この場合、エフェクタの派生クラスとして重力、引力、弾性力、減衰力などを準備している。

エフェクタは操作として、Effect() と Update() を実行することができる。操作 Effect() が呼び出されると、自分の影響下にあるパーティクルに与える影響を計算し、その影響をパーティクルに与える。ばねのような弾性力の場合には、ばねの両端のパーティクルに対して加わる力を、フックの法則から求めて、両端のパーティクルに与える。操作 Update() が呼び出されると、自分自身の状態を指定された時間分だけ更新する。風力のように時間により力の方向や大きさが変化する場合にはこのメッセージにより状態を変化させる。

### (c) センサ(Sensor)

センサは、パーティクルの現在の状態を監視するオブジェクトであり、その派生クラスとして時間センサと位置センサがある。それぞれは、下記のような働きをする。時間センサは、パーティクルの寿命が尽きたかどうかを判定し、尽きている場合はそのパーティクルを消滅させる。位置センサは、パーティクルの位置を監視し、特定の場所に到達した場合に、そのパーティクルを消滅させる。

### (d) パーティクルシステム(ParticleSystem)

パーティクルシステムは、前述したパーティクル、エフェクタ、センサで構成される系全体を管理するオブジェクトである。このオブジェクトは、系全体の状態更新と描画の機能を提供する。系を構成する個々のオブジェクトに注意を払うことなく、パーティクルシステムの操作を呼び出すだけで、系全体を操作することが可能になる。

パーティクルシステムは、Update( $\Delta t$ ) と Draw() の2つの操作を実行する。操作 Update( $\Delta t$ ) はシステム全体の状態を時間  $\Delta t$  だけ更新するた

めのものである。一方、操作 Draw() はシステムの状態を描画するためのものである。

次に、操作 Update( $\Delta t$ ) を受けたときのパーティクルシステムの振る舞いを図 4 で、具体的に説明する。この図は、図 1 で述べた自由落下する水滴を構成的パーティクルシステムで表現する場合の、オブジェクトの組合わせの例を示している。系は、5 つのオブジェクトで構成されている。パーティクルは、水滴を表す。エフェクタの派生クラスである重力はパーティクルに対して鉛直下方向に一定の力を与える。同じくエフェクタの派生クラスである減衰力は、パーティクルの進行を妨げる方向に力を与える。センサの派生クラスである位置センサは、パーティクルが水面に達するのを監視し消滅させる。

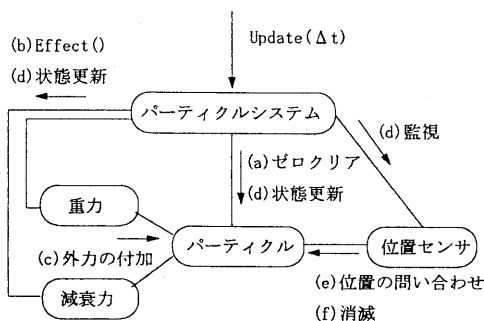


図4 パーティクルシステムの振る舞い

パーティクルシステムはこの操作 Update( $\Delta t$ ) が呼び出されると、次のように振る舞う。

- (1) パーティクルが受ける外力をゼロクリアする (a)。理由は(2)で述べる。
- (2) パーティクルに影響を与えるように、重力と減衰力の操作 Effect() を呼び出す(b)。操作が呼び出されたを受けた重力と減衰力は、パーティクルに対して力を加える(c)。この力はパーティクルの外力に加算されるため、前回の状態更新時の影響が伝播を避けるために、(1)においてパーティクルの外力をゼロクリアする。
- (3) 重力、減衰力、パーティクルに対して、状態更新の操作 Update( $\Delta t$ ) を呼び出す(d)。操作が呼び出されたオブジェクトは自分自身の状態

を更新する。

- (4) 位置センサに対して、操作 Sense() を呼び出す(e)。位置センサは、パーティクルの位置を問い合わせ、水面に達しているならばそのパーティクルを消滅させる(f)。

### 2.3 パーティクルの動作生成

ここでは、パーティクルの構成要素であるソルバが、どのようにパーティクルを動かすのか説明する。

ソルバはパーティクルから動作生成の依頼を受けると、パーティクルの現在の内部状態から次のタイムステップの状態を計算して更新する。この計算方法としては、さまざまなものが考えられるが、パーティクルがニュートンの運動方程式に従うならば、その動きは次式に従う。

$$f = ma \quad (1)$$

ここで、 $m$  はパーティクルの質量、 $f$  はパーティクルに加わる力、 $a$  はパーティクルに発生する加速度を表す。この式は、パーティクルに力  $f$  が加えられると、パーティクルに加速度が生じ、その結果、位置と速度が変化することを意味する。速度と時間を時間で微分したものがそれぞれ加速度と速度であるので、式(1)を時間で積分することで、速度・位置を求めることが可能である。ここで、数値積分法として最も簡単なオイラー法<sup>9)</sup>を適用すると、現在のパーティクルの位置  $p(t)$  と速度  $v(t)$  に対する、時間  $\Delta t$  後のパーティクルの位置  $p(t+\Delta t)$  と速度  $v(t+\Delta t)$  は次式で表現できる。

$$\begin{aligned} v(t+\Delta t) &= v(t) + f(t)/m \Delta t \\ p(t+\Delta t) &= p(t) + v(t) \Delta t \end{aligned} \quad (2)$$

ソルバは当式に従って、時刻  $t$  でのパーティクルの状態  $f(t)$ ,  $v(t)$ ,  $p(t)$  から、時間後の状態  $v(t+\Delta t)$ ,  $p(t+\Delta t)$  を求めることができる。

なお、オイラー法はそのアルゴリズムの単純さから、高速な演算速度が期待できるが、計算精度は高くない。必要に応じて、ホイン法、ルンゲクッタ法などの精度が高い積分法<sup>9)</sup>を利用する。

## 2.4 パーティクルの描画

次に、パーティクルの構成要素であるドローワが、どのようにパーティクルを描画するのか説明する。

ドローワはパーティクルから描画の依頼を受けると、パーティクルの内部状態を可視化する。パーティクルの位置、大きさ、色、透明度などの内部状態の情報を可視化の際に利用する。

具体的な描画形状については、パーティクルで表現すべきものに大きく依存する。そこで、点・線・球・ポリゴン・ビルボードポリゴンなどを描画するオブジェクトをあらかじめ準備しておき、必要に応じて組み合わせ方を変えることにする。これらはドローワクラスの派生クラスとして定義する。

これら描画形状の中で重要なのはビルボードポリゴンである。これは、木のように複雑な形状を簡易表現するためのものであり、ドライブシミュレータなどビジュアルシミュレータで広く利用されている。ビルボードポリゴンは、視点の動きに追随して回転して、視点に正対するので、テクスチャマッピングを施すことで、少ないポリゴン数でパーティクルを表現することができる。

## 3. 水の効果の表現

ここでは、前章で説明した構成的パーティクルシステムの応用例の一つとして、水の効果の表現方法を説明する。大海原を自由奔放に泳ぎまわるイルカの動きに応じて、水しぶき、泡、波紋といった水の効果を付加する。

### 3.1 水の効果の種類

まず、イルカの動きに応じて、いつ、どのような水の効果が発生させれば良いか考えてみる。発生のタイミングとしては、図5に示すように(a)ジャンプする場合、(b)着水する場合、(c)水面を移動する場合、の3種類を挙げることができる。イルカが、ジャンプする場合には、水面には波紋が生じ、水上にはしぶきが飛び散るであろう。同様に、着水する場合には、波紋、しぶき、泡が生じる。水面を移動する場合には、航跡としぶきが生じる。ここで、しぶきは発生条件により振る舞いが異なるので、図5では発生条件を付すことにより区別している。

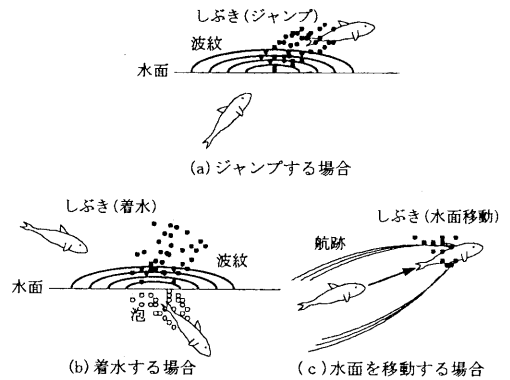


図5 イルカの動きにともなう水の効果

### 3.2 イルカの状態遷移

上述した水の効果は、イルカのジャンプなど特定のイベントを検知して、それに同期させて発生させる必要がある。このようなイベントを検知するために、イルカのとり得る状態を考え、状態の遷移と発生すべき水の効果を対応付けて考えることにする。

まず、イルカのとり得る状態について考える。水面を境界にして、イルカが水中にいる状態、水上にいる状態、そして、ちょうど水面の境界上にいる状態が想定できる。さらに、イルカが水面上にいる状態でも、ジャンプ時に水中から水上へ移動中である場合と、着水時に水上から水中へ移動中である場合と、水面に平行に泳いでいる場合とでは、発生する水の効果が異なるので区別する必要がある。これらから、イルカのとり得る状態としては表1に示す5状態を考えれば良いことになる。

次に、表1に挙げた状態間の遷移の様子を図6の状態遷移図に示す。ただし、イルカが普通に海を泳いでいて起こり得ない状態遷移は図には記述していない。この図の状態遷移と発生させるべき水の効果を対応付けるのが表2である。この表と図6から例えば次のことがわかる。s1 から s2 への状態遷移は遷移1であり、遷移1の場合は、しぶき(ジャンプ)と波紋を発生させれば良い。

以上のことから、イルカの状態を何らかの方法で知ることが可能ならば、その状態の遷移から、発生させるべき水の効果が分かることになる。

表1 イルカのとり得る状態

状態番号	イルカの状態
s1	水中
s2	水上
s3	水面上
	ただし、水中から水上へ移動中
s4	水面上
	ただし、水上から水中へ移動中
s5	水面上
	ただし、水面を移動中

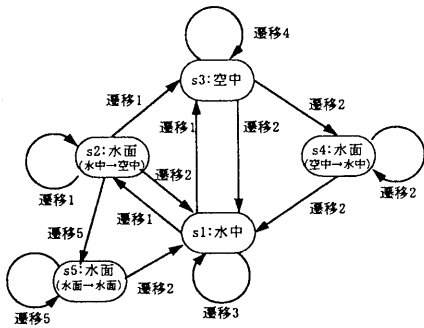


図6 イルカの状態遷移図

表2 状態遷移と水の効果

状態遷移	発生させる水の効果
遷移1	しぶき(ジャンプ)、波紋
遷移2	しぶき(着水)、波紋、泡
遷移3	泡
遷移4	しぶき(ジャンプ)
遷移5	しぶき(水面移動)、航跡

### 3.3 ジェネレータによるパーティクルの発生

前節の議論で、イルカの動きに応じて、いつ、どのような水の効果が発生させれば良いのか分かった。ここでは、所望の水の効果をどのように発生させるのか考える。

ここでは、水の効果をパーティクルの集合で表現

する。そのため、イルカの状態の遷移に応じてパーティクルを発生させる必要がある。ここでは、このようなパーティクルの発生源としてジェネレータオブジェクトを新たに導入する。これはパーティクルの派生クラスであり、状態更新毎にパーティクルを発生する特別なオブジェクトである。ジェネレータをイルカ上に配置することで、イルカの動きに応じたパーティクルの生成が可能になる。

次に、泡を表現する場合に、ジェネレータが発生するパーティクルについて具体的に説明する。泡は、イルカが着水したときに、水中でジェネレータの軌跡上に発生する。その泡は水の浮力の影響を受け、水中から上昇し水面へ達すると消滅する。このような動きを生成するために発生させるパーティクルの概要を次に説明する。

- (1) 初期値：生成するパーティクルの数は、イルカが着水してからの経過時間に比例して少なくなると単純化して考える。パーティクルの位置は、ジェネレータ上に、速度は、はランダムに決める。
- (2) ドロワー：泡の描画はビルボードで行うため、ドロワーとしてビルボードドロワーを選ぶ。このとき、透明度をもつテクスチャをマッピングして、ビルボードが球に見えるようにする。
- (3) エフェクタ：生成したパーティクルには、浮力を生じるように鉛直上方向に重力を与える。また、パーティクルの速度が必要以上に大きくならないように減衰力も与える。
- (4) センサ：水面に達したら寿命が尽きるようにセンサで監視する。

以上、泡の表現を例にジェネレータが発生するパーティクルについて説明したが、ほかのしぶき、波紋、航跡についても同様に考えることができる。

### 3.4 結果と検討

本手法による水の効果の表現を確認するためのプログラムを、シリコングラフィックス社のグラフィックスワークステーション Onyx/RealityEngine上に実装した。まず、第2章で述べた構成的パーティクルシステムと本章で述べたジェネレータを、C++およびC言語を用いてクラスライブラリの形で構築した。そして、イルカの動きに応じて水の効果を発生させるようにした。

仮想空間の海をリアルタイムで泳ぎまわるイルカの例を図 7 に示す。また、アニメーションの例を、図 8、図 9 に示す。イルカのジャンプに応じて水しぶきや波紋を発生させている。これらの効果を表現するパーティクルは、時間とともに位置や大きさを変化させ、やがて消滅する。イルカの形状のポリゴン数は約 1000 である。また、発生するパーティクルの数はフレーム当たり最大で 1500 個程度である。イルカが一匹の場合は、いずれの例も秒 20~30 フレームの映像の更新が可能であり、動きの滑らかさとリアルさを実現している。実験により、イルカ 3 匹までは十分な対話性が実現できることを確認している。

#### 4. 一般化

前章では構成的パーティクルシステムの応用例の一つとして水の効果を扱ったが、ほかにも応用の範囲は広い。水の効果の場合と同様に、霧・炎・火花・爆発などの効果がパーティクル、エフェクタ、センサを組み合わせることで表現可能である。たとえば、霧を表現するには、霧を発生させたい空間に複数のパーティクルを生成して、それらのパーティクルにエフェクタで風力を与えればよい。パーティクルの配置を空間的に不均一にすれば、霧の濃淡も表現可能である。

現実の自然界をよりリアルに表現するには、当然より一般の物理系を扱う必要がある。その中で構成的パーティクルシステムの役割も大きい、例えば、霧深い港に、イルカがやって来るシーンを表すには、霧も波しぶきも一つの大きな系の中で、扱われることになる。このような試みとして、外力として風力場を扱い、それから統一的に複雑なシーンを記述する方法が提案されている<sup>7)</sup>。リアルタイム性という課題を別途検討することにすれば、この考えと、本システムとを組み合わせるといふ単純な一般化によっても得られる効果は大きなものになると期待できる。

#### 5. まとめ

本報告では、映像特殊効果を表現するための構成的パーティクルシステムを提案した。基本的なオブジェクトを組み合わせることで多彩な表現が可能なのが特徴である。また、応用としてイルカの動きに応じて、しぶき、泡、波紋といった水の表現を

現した例を示した。

今後の課題は、対話的にオブジェクトを組み合わせたり、オブジェクトの内部状態を変更するユーザインタフェースの実現などである。

本報告で示したシステムは、映像特殊効果の表現のほか、科学技術計算結果の可視化にも応用可能である。

#### 謝辞

本研究を進めるにあたり御協力頂いた、日立製作所デザイン研究所の西山譲二氏をはじめとする、同デザイン研究所、同中央研究所、同システム開発研究所の各位に感謝致します。

#### 参考文献

- 1) Reeves, W. T., "Particle System. A Technique for Modeling a Class of Fuzzy Objects," *Computer Graphics* 17(3), 1983, pp.359-376.
- 2) Fouriner, A. and Reeves, W., "A Simple Model of Ocean Waves," *Computer Graphics* 20(4), 1986, pp.75-84.
- 3) Peachy, D. R., "Modeling Waves and Surf," *Computer Graphics* 20(4), 1986, pp.65-74.
- 4) Sims, K., "Particle Animation and Rendering Using Data Parallel Computation," *Computer Graphics* 24(4), 1990, pp.405-413.
- 5) Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F., Lorensen, W., "Object-Oriented Modeling and Design," 1991, Prentice-Hall.
- 6) 洲之内治男, "数値計算," 1978, サイエンス社.
- 7) Shinya, M. and Fournier, A., "Stochastic Motion - Motion Under the Influence of Wind," *Proc. Eurographics '92*, 1992, pp.119-128.

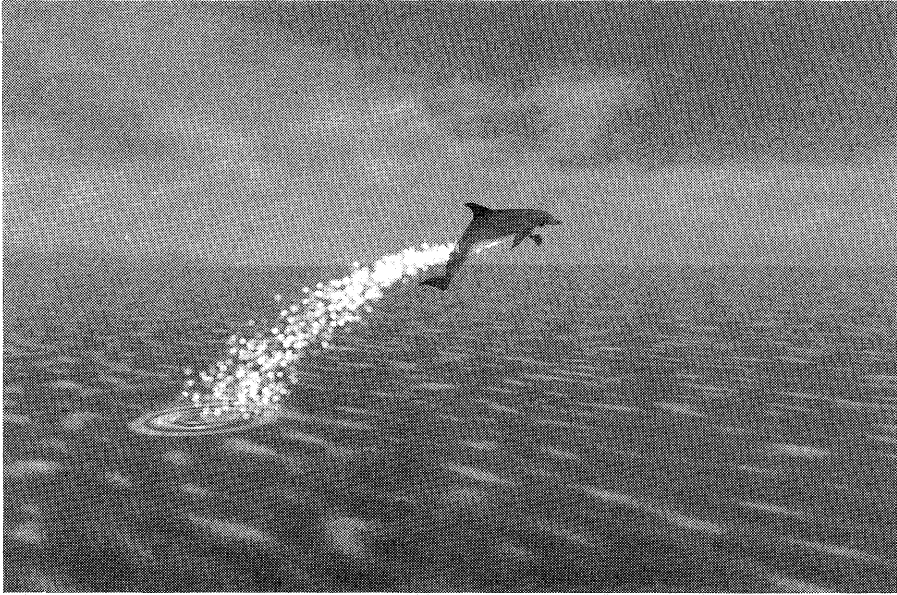


図 7 イルカがジャンプするシーン

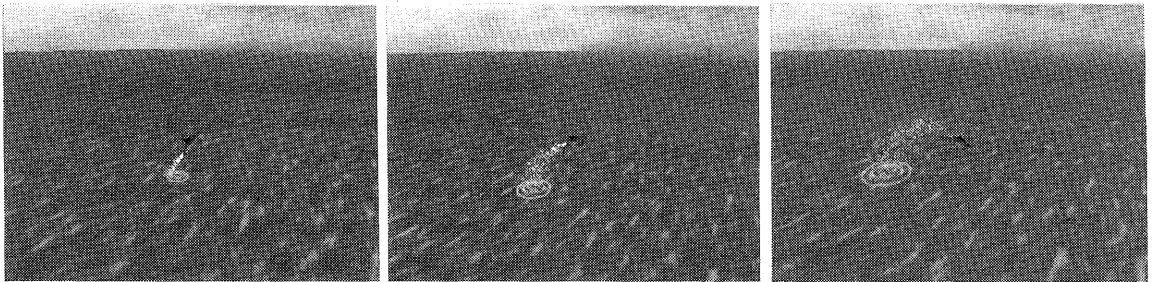


図 8 ジャンプ時のアニメーション

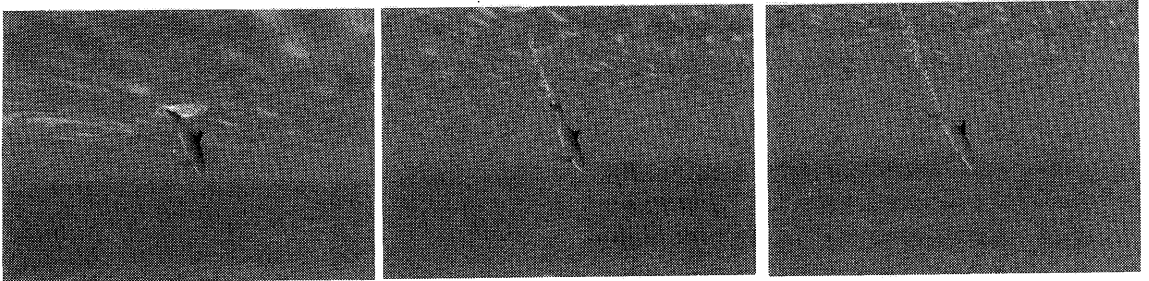


図 9 着水時のアニメーション