

仮想水泳のための対話的な波と水しぶきの表現

舟橋健司*¹ シドニー・フェルズ*²

Expression of Water Wave and Splash for Interactive Virtual Swimming

Kenji Funahashi*¹ and Sidney Fels*²

Abstract – In this paper, we describe the method to express water wave and splash for virtual swimming as artwork; *Swimming Across the Pacific*. Our water simulation is based on the method in which the water surface is modeled as a thin film. For the sea surface plane waves, we use recurrence relations to solve the partial differential equation for the 2D wave as is standard practice. In this method, the sea surface is modeled as a mesh. We approximate the volume of the swimmer with bounding boxes to improve performance and determine when they intersect any of the water grids when they move. Then waves are then made and propagated. If the height is over a pre-defined threshold value and/or the swimming avatars motion interferes with waves, some particles are made in the air to simulate splashing.

Keywords : virtual swimming, interactive manipulation, water wave, water splash, sea surface

1. はじめに

バーチャルリアリティ (VR) 技術の発展とともに、アートやエンターテイメントでも VR 技術が 1 つの表現手段として用いられてきている。我々が現在取り組んでいるアートワーク、*Swimming Across the Pacific* (SAP) もその 1 つである。この作品は 1982 年、Alzek Mischeff により行われた *Swimming Across the Atlantic* [1] がきっかけとなっている。Mischeff は大西洋を航海する客船上のプールで泳ぐことにより、「泳いで大西洋を横断した」。我々の SAP では、仮想水泳システムを構築し、これを太平洋を横断する飛行機内に設置し、「泳いで太平洋を横断する」ことを目標としている。

ところで、我々人間は 3 次元世界で生活しているとはいえ、坂道や階段などを利用して 3 次元の移動をしているだけで、実際には「面の上」に拘束されている。少々の跳躍はできても、鳥のように飛び回ることはできない。ただし、水の中を除いては、である。プールや海の中であれば、我々人間も道具を使うことなく 3 次元の移動が可能なのである。高度な仮想水泳システムが完成されれば、新たな、そして自由度の高い口コモーションインタフェースとなるであろう。

本論文では、SAP、仮想水泳システムのための海面における波や水しぶきの表現方法について述べる。液体を扱う学問に流体力学があるが、計算量が多いためリアルタイム処理を行う必要がある対話操作には直接適用することができない。一方、コンピュータグラフィックス (CG) や VR の分野においても流体に関する研究は行われているが [2] ~ [5]、これらの研究も、主にリアリティの高い CG を作成することを目的としているため対話操作には向いていない。

筆者はこれまでに、仮想空間における容器による液体の対話操作を実現する仮想液体操作モデルを提案している [6] ~ [8]。しかし、このモデルでは仮想空間内に液体との干渉を判定すべき仮想の容器を定義し、容器と液体との相互作用についてのみ考えているため、仮想水泳システムへの適用は困難である。そこで、対話操作が可能な 2 次元波動方程式に基づく波の表現とパーティクルに基づく水しぶきの表現手法について以下で述べる。なお、このモデルはアートワーク SAP を第 1 の目的としたものであり、挙動の厳密な再現を主な目的とはしていない。そのため各手法やパラメータは物理法則を参考にはしているが、経験的、定性的評価に基づいて決定したものである。

本論文で述べる表現手法をインプリメントした SAP プロトタイプシステムは、SIGGRAPH2004、Emerging Technology などに展覧された [9] ~ [11]。

*1: 名古屋工業大学 情報メディア教育センター

*2: プリティッシュコロンビア大学

*1: Center for Information and Media Studies, Nagoya Institute of Technology

*2: Department of Electrical and Computer Engineering, University of British Columbia

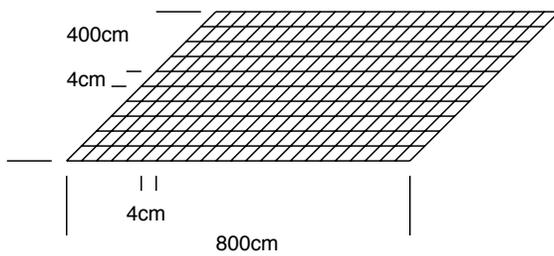


図1 メッシュによる表現

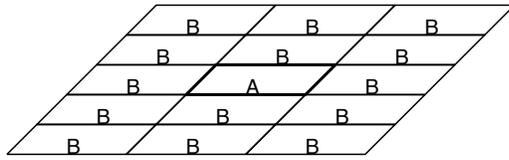


図2 有限サイズの水面による表現

2. 対話操作のためのモデル

2.1 海面モデルと波の表現

まず水面を平面として考え、メッシュ状に分割する(図1)。各格子間隔は4cmとして考える。各格子点は上下方向のみに移動可能とし、時刻 t における格子点 (i, j) の高さを $h[i][j]$ とする。このメッシュの波を考え、2次元波動方程式を差分方程式として解く。

2次元波動方程式を次式に示す。

$$\frac{\partial^2 z}{\partial t^2} = v^2 \left(\frac{\partial^2 z}{\partial x^2} + \frac{\partial^2 z}{\partial y^2} \right)$$

ここで、 v は波が水面上を伝播する速度であり、座標 (x, y) における波の高さ z の関係を表している。これを格子点間隔 d で、中心差分で近似することにより、次式を得る。

$$\begin{aligned} & (h[i][j](t + \Delta t) - 2h[i][j](t) + h[i][j](t - \Delta t)) / \Delta t^2 \\ & = (h[i + 1][j](t) + h[i - 1][j](t) \\ & + h[i][j + 1](t) + h[i][j - 1](t) - 4h[i][j](t)) v^2 / d^2 \end{aligned}$$

ここで簡略化のために、 Δt に対して次式

$$\frac{v^2 \Delta t^2}{d^2} = \frac{1}{2}$$

が成立するように v と d を設定すると仮定することにより、 $h[i][j]$ に関する漸化式を得る。実際には波の減衰を考慮する必要があるため、次の漸化式に示すように右辺に減衰係数を乗じる。

$$\begin{aligned} h[i][j](t + \Delta t) = & ((h[i - 1][j](t) + h[i + 1][j](t) \\ & + h[i][j - 1](t) + h[i][j + 1](t)) / 2 \\ & - h[i][j](t - \Delta t)) \times c \end{aligned}$$

ここで c は減衰係数で、0.97 に設定する。この手法はゲームなどでも用いられている方法である [12]。し

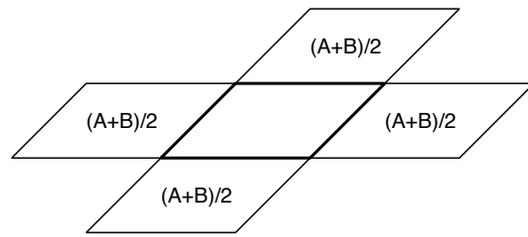


図3 境界における波の伝播

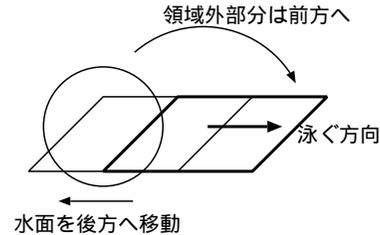


図4 泳ぎによる波の移動

かし、対話操作を行うとなると格子ごとに操作を指定する必要があり、困難である。水しぶきの表現も不可能である。

またこの漸化式では、 i, j の上限値と下限値において値を更新せずに0としておくことにより波の反射も表現可能であるが、海のように極めて広い範囲を対象とすると、計算時間が非常に増大し対話操作が不可能となる。そこで、泳ぎの進行方向へ800cm、左右に400cm相当の有限の水面を定義する。さらに、同一サイズの水面を準備し、図2のように配置する。ここで水面Aは後述する干渉判定を行い、インタラクションを可能とする領域である。水面Bは全て同一とし、同じ形状の波となっているものとする。すなわち、いずれか一つの領域について計算を行い、そのコピーを他の13の領域にも表示する。

各水面の境界における波の伝播は次のように処理する。まず、水面AとBの各格子点の高さを求める。すなわち、水面AとBの各格子点の高さをそれぞれ $h_a[i][j]$ と $h_b[i][j]$ で表した場合、平均水面(図3では「 $(A+B)/2$ 」と表記)の各格子点の高さを $h_{ab}[i][j] = (h_a[i][j] + h_b[i][j]) / 2$ として求める。実際には境界付近の格子点の計算のみが必要となる。境界付近の格子点の高さを更新したい水面(AまたはB)を図3の中央に当てはめて考え、前述の漸化式を拡張して計算することにより、各水面の境界を超えた波の伝播を実現する。例えば水面メッシュのサイズが $I+1 \times J+1$ の場合、水面Aの境界格子 $(0,0)$ の高さ

は以下の式により更新する。

$$h_a[0][0](t + \Delta t) = ((h_{ab}[I][0](t) + h_a[1][0](t) + h_{ab}[0][J](t) + h_a[0][1](t))/2 - h_a[0][0](t - \Delta t)) \times c$$

さらに、泳ぐことにより前方に進む場合には、格子点の高さ情報を後方へ移動し、領域から出た部分を前方部分に当てはめる処理を行う(図4)。泳ぐ向きを左右に変えた場合にも、人間は水面 A の中央部分に、長辺の方向を向いたままとするようになる。これらの処理により、少ない計算量で、対話操作の可能な広い海面を表現することができる。

図2により描画される海面は十分な領域となるため、その外側には水面の描画色と同一の色、透明度で平面を描くことにより十分な臨場感を得ることができる。

2.2 対話操作による波の生成

前述した水面 A との対話操作を実現するためには、各格子点の高さに作用を加える必要がある。しかし、水泳を実現するためには複雑な形状の人間を、水面付近に配置する必要がある。この場合、水面メッシュに「穴」を開けることになる。すなわち、水面メッシュのうちの複雑な領域を計算の対象外とする必要があり、具体的には、メッシュの形状を変更する等の処理が必要となる。しかし、これをリアルタイムに実現することは容易ではない。そこでまず、人体のモデルを描画用と対話操作の2つに分けて考える。対話操作モデルを、図5に示す軸に平行な直方体の集合で表現する。腕や脚については、泳ぐときの波や水しぶきの生成に影響が少ないと判断し上腕と大腿部を省略する。

これらの直方体は完全に人体と一致するものではなく、水面格子との干渉判定を容易とするために、平行移動のみを認めるものとする(図6、描画用モデルは、説明のための仮表示)。腕や脚が前後左右の軸に対して斜めになるような場合に、直方体を2分割して表現することや、前後方向に長い直方体で表された腕や脚が左右方向に向いたときに、直方体を軸に平行なまま変形して表現することも可能である。また、人体の内側にあたる部分は必然的に不可視となるため、水面メッシュに「穴」を開けることなく、以下に示すように一時的に壁があるとして計算することにより波の反射を実現する。これらのモデリングにより、短時間での干渉判定が可能となる。

水面 A のある格子点に注目した場合、その格子点が時刻 $t - \Delta t$ において人体対話操作直方体のいずれか(以下、直方体)に含まれ、時刻 t で含まれない場合、格子点の高さを 4cm 増加させる(干渉1)。ある格子点が時刻 $t - \Delta t$ において直方体に含まれず、時刻 t で含まれる場合、格子点の高さを 4cm 減少させる(干

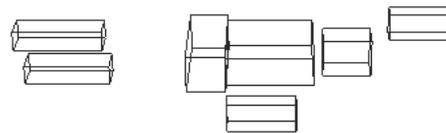


図5 対話操作作用人体モデル

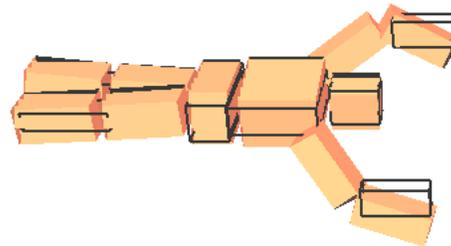


図6 描画用モデル(仮)の同時表示

渉2)。この場合は不可視となるが、隣の格子点への伝播により、可視部分へと影響していく。時刻 $t, t - \Delta t$ いずれにおいても直方体に含まれる場合は、格子点の高さを 0 とする。この手続きにより、静止している人体に波が衝突した場合の反射波の生成と、人体が動くことによる波の生成が可能である。

海には、泳ぎにより発生する波だけでなく、風などの様々な要因による波もある。そこで、一連の処理ごとに次に示す方法により、水面 B に自然な波を発生させる。

1. 各格子 (i, j) に対して 0 から 500 の実数の範囲で乱数 r を発生させる。
2. 乱数 r が 499.9 より大きい値の場合には、次の手順 3,4 で波を発生させる。
3. $h_b[i][j], h_b[i][j+1], h_b[i][j-1]$ の各値に $r / 250$ を加える。
4. $h_b[i+1][j], h_b[i-1][j]$ および $h_b[i][j](t - \Delta t)$ の各値に $r / 500$ を加える。

これは波に向かって泳ぐことを想定したものであり、横長の楕円状の波紋を発生させている。上記手順における $r/250$ と $r/500$ はそれぞれ、およそ 2 と 1 となる。システム起動直後はいくつかの波紋が発生するだけであるが、次第に波が伝播していき、水面 A も含めて自然な海面が形成されていく。

2.3 水しぶきの生成と表現

水しぶきの対話操作可能なシミュレーションは極めて困難な問題である。CG の分野ではパーティクルシ

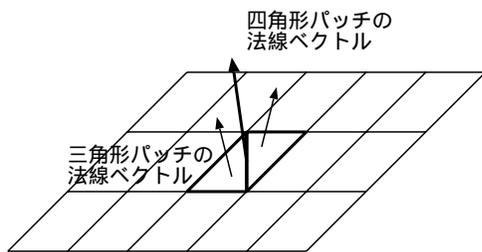


図7 単純な三角形パッチ表示

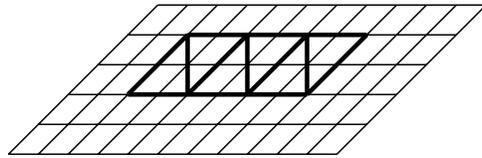


図8 周辺水面の簡易表示

ステムが良く用いられている [13]。本システムでも、このパーティクルシステムをベースとし、単純化したモデルを用いる。まず水しぶきを粒子として扱う。各粒子は干渉しないものとし、重力と慣性に従い自由落下するものと定義する。ただし、微小の乱数により空気抵抗によるばらつきを表現している。水しぶきは水面 A でのみ考慮するものとし、各粒子が水面下 10cm に達した場合、粒子を消滅させ、その位置に相当する格子点の高さを 1cm 増加させる。水面下 10cm とするのは、水しぶきの表現をそのまま水泡に流用するためである。

次に、水しぶきの発生について説明する。ある格子 (i,j) に注目し、その 4 近傍の格子の高さの平均 a を求める。条件式 $h[i][j] > a + 12\text{cm}$ が成り立つ場合、 $h[i][j] = a + 12$ と設定し、格子 (i,j) 上の時刻 $t - \Delta t$ での高さの位置を中心に半径 4cm の範囲で乱数により座標を決定し、次式で表される初速度 s の粒子として、10 個の粒子を発生させる。

$$s = \begin{pmatrix} (h[i-1][j] - h[i+1][j]) - 6, \\ h[i][j-1] - h[i][j+1], \\ (h[i][j] - h[i][j](t - \Delta t))/24 \end{pmatrix}$$

さらに、前節で述べた人体モデルと水面との干渉 (干渉 1,2) においても、水しぶきを発生させる。発生させる位置は格子 (i,j) 上の高さ $h[i][j]$ を中心とし、初速度は $(-1.5, 0, 1)$ とする。これは、進行方向と反対方向に水しぶきを飛ばすためである。乱数により決定する発生座標範囲の半径と粒子数は、それぞれ、干渉 1 では 12 cm, 50 個、干渉 2 では 4cm, 10 個とする。

水しぶきのリアルタイムでの生成は、先にも述べたように極めて困難であるため、得られる結果が自然に受け入れられるようなモデルを、手続きやパラメータを変更しながら経験的に作成した。

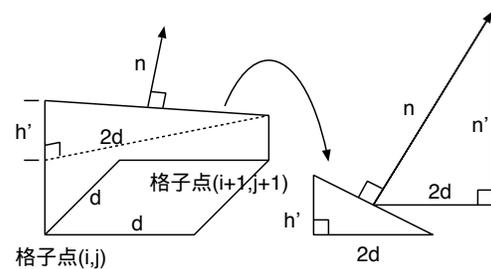


図9 法線ベクトルの算出

3. 波と水しぶきの描画処理

3.1 水しぶきの描画方法

まず本節で、簡単な水しぶきの描画に関して述べる。各粒子は XGA 相当の画面を想定し、2.25 から 6.75 の範囲で発生させた乱数 p に対し、小数以下を切り捨てた値 p' を求め、 p' ピクセル \times p' ピクセルの正方形として OpenGL の機能を用いて単純に描画する。粒子の色を R,G,B で指定するとき、 $G=B=1$ とし、R を 0.7 から 1.0 の乱数で設定し、透明度を 0.35 から 0.5 の乱数で設定する。ただし、1つの粒子データに対して1つの正方形描画とせず、粒子の位置を中央に 8 近傍の座標を考え、これらの位置も含めて合計 9 個の正方形を描画する。近傍へのオフセットは $(p + 0.5) \times 3$ とする。この方法により、粒子に関わる計算量を押えながら広がりのある水しぶきを表現する。

3.2 波の描画方法

メッシュで表現された波の描画は、まず各四角形パッチを三角形パッチに 2 分割し、それぞれの法線ベクトルの平均を四角形パッチの法線ベクトルとする。各格子における疑似的な法線ベクトルを、隣接する 4 つの四角形パッチの法線ベクトルの平均として求め、OpenGL の機能により半透明でスムーズシェーディング表示する (図 7)。ただし、水面 B に対しては 4 つのセルごとにまとめて考え、それらを 2 つの三角形パッチに分割し、表示処理を削減する (図 8)。また、水面 B に関しては OpenGL の機能、ディスプレイリストを利用して描画処理を一旦記憶させておき、水面 A を囲むように 14 個の水面 B を描画することにより処理速度を向上する。さらに、これらを囲むように前後左右方向に 2,000cm ずつ同じ色と透明度で、平面を描画する。

3.3 描画に関わる処理時間の短縮

対話操作のための十分な処理速度を得られるようなモデルとして構築してきたが、実際にアートワーク SAP ヘインプリメントするためには、さらに空や海底、海の生物の表示、3次元位置センサからの泳ぎ情報の取得など、様々な処理が必要となり、さらなる高速化が必要となった。そこで実際に各処理に費される

処理時間を解析したところ、以下のような結果が得られた。なお、OS や Window システム、他の処理に関しては除外している。

入力処理：シーンの構築：実際の描画

= 0.027 : 0.727 : 0.245

(入力処理：マウスなどによる人体の簡易制御用)

さらに、シーンの構築部分の詳細な解析結果は以下の通りである。

粒子の落下：波の伝播と対話操作：描画の準備

= 0.01 : 0.10 : 0.89

描画の準備とは、分割した三角形パッチの法線ベクトルを求め、四角形パッチの法線ベクトルを求め、さらに各格子点での疑似法線を求めるための処理と、水面 B のディスプレイリストの作成である。また、水面 B の表示は 4 つのセルを統合して行っているが、各格子の疑似法線は全て計算していた。

水面 A の準備：水面 B の準備 = 0.36 : 0.64

そこで、まず水面 B の各格子点での疑似法線の算出に必要な格子のみに限定する。また、以下の手順により、簡易的、疑似的に求めたベクトルを四角形パッチの法線とし各格子の法線を求める。図 9 に示すように、格子点 (i,j) を含む 1 つの四角形パッチについて考える。このとき、図に示されるベクトル n の高さ方向の成分 n' は、三角形の相似関係から以下のように求まる。

$$n' = 2d^2 / (h[i][j] - h[i+1][j+1])$$

すなわち、ベクトル n は (d, d, n') として求まる。ただし、 n' が負の値となる場合には $(-d, -d, -n')$ とする。ここで、 $d = 4\text{cm}$ である。また、 h' が 0 の場合には $(0, 0, 1)$ とする。同様にして四角形パッチのもう 1 つの対角に位置する格子点の組でもベクトルを求め、これらの和を四角形パッチの法線ベクトルとする。以上の変更により、以下のように処理時間が短縮された。

準備 A : 準備 B : 準備 A (新) : 準備 B (新)

= 0.36 : 0.64 : 0.224 : 0.382

また全体では以下のようになり、30% 程度の処理時間が削減できた。

入力処理：シーンの構築：実際の描画

= 0.027 : 0.454 : 0.245

アートワーク SAP では、泳いでいるところを鳥の視点から見ただけでなく、HMD を利用して泳いでいる人間の視点からのシーンも見られるようにする。これらは、他の観客からも見られるようにプロジェクタなどによる表示も行う。同時に複数の視点からのシーンを作成することを考え、複数のマシンを利用する方法もあるが、今回は 1 台のマシンで 2 つのシーンを描

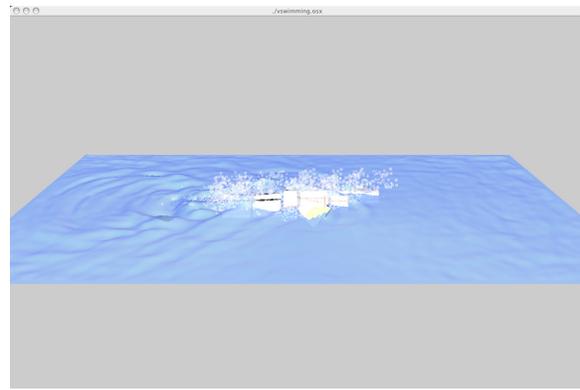


図 10 対話操作作用水面のみの表示

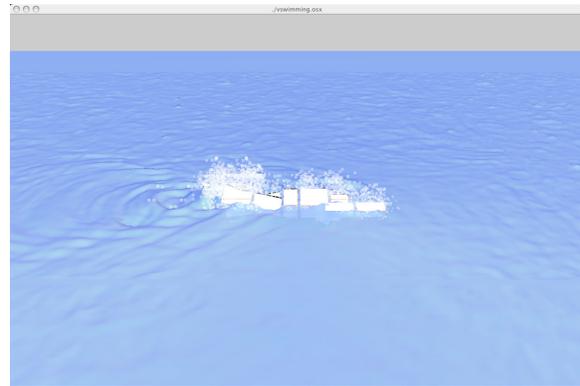


図 11 水面全体の表示

画することにした。すると、当然のことであるが、実際の描画にかかる時間がおよそ 2 倍になり、システムのパフォーマンスが低くなる。そこで、以下の手順により描画を行い、表現に違和感を感じない範囲で精度を低くし、処理時間を短縮した。

1. 入力処理
2. シーンの構築
3. 視点 1 での描画
4. 入力処理
5. シーンの構築
6. 視点 2 での描画
7. 入力処理
8. シーンの構築
9. :

4. 実験および結果

上述のモデルに基づいた仮想水泳の実験システムを iBook/PowerBook 上で、C 言語、OpenGL により構築した。図 10 は、水面 A と仮の人体描画用モデルによる表示である。図 11 は、全ての水面と仮の人体である。自然な波や水しぶきが表現されており、また水面 A と B との境界も大きな違和感はないことが分かる。図 12 は同時に 2 つの視点からの様子を表示したものである。静止画取得のためにウインドウ表示としてい

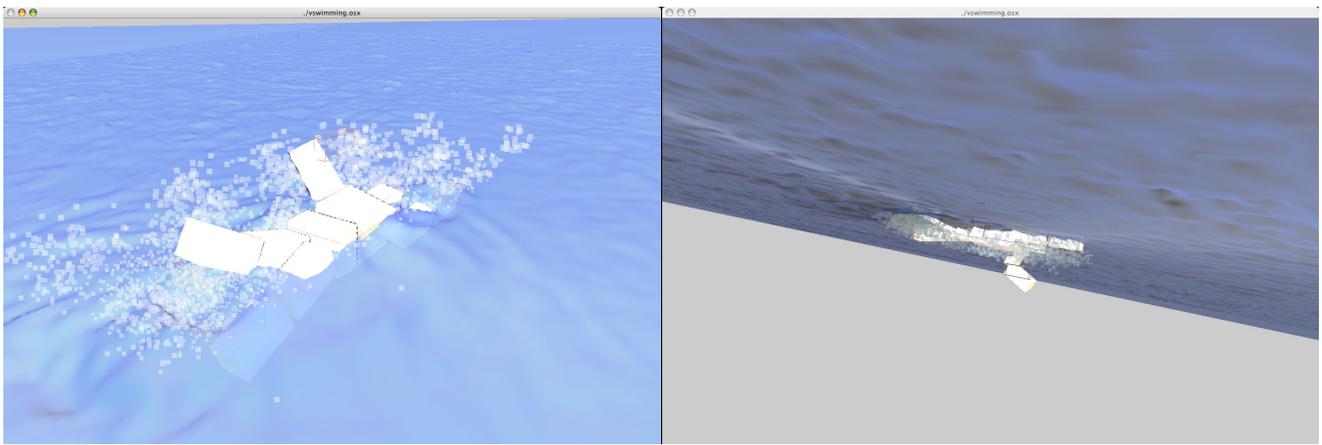


図 12 異なる視点からの 2 画面表示



図 13 一つの直方体による干涉

るが、実際には 2 つのディスプレイそれぞれに全画面表示を行っている。処理時間に関しては、1 画面の場合、PowerBook G4 1.33GHz, 2GB DDR SDRAM で約 24 fps での描画更新が可能であった。

単純な直方体一つの場合について検証するために、人体モデルを水面より上方に設置し、右手をあらわす直方体のみで水面への干涉を行った。図 13(左)より直方体が水面に干涉すると水しぶきが発生していることが確認できる。このとき 2 回連続で同じ場所に干涉を行っており、図 13(右)からは水しぶきと同時に発生した波紋が 2 つ広がっているのが確認できる。

また、本手法をインプリメントし、SIGGRAPH2004 Emerging Technology に出展した SAP システムの様

子を図 14-16 に示す。こちらは Linux 上で実行し、人体各部の位置は Polhemus FASTRAK 2 台、8 個のセンサにより計測している。また、波の音なども付加している。多くの方々に体験してもらい、様々な感想を頂いた。CG に詳しい方などからは、リアルタイムでこれだけの波や水しぶきの表現ができていることに対して良い評価を頂いた。また詳しくない方などからは、特に CG に関するコメントがない場合もあった。これは体験型システムのため、そちらに興味がいき、また CG に関しては特に違和感を感じなかったためと判断できる。

5. むすび

本論文では、アートワーク *Swimming Across the Pacific (SAP)* のために、仮想空間で水泳を実現するための、対話的な操作が可能な波と水しぶきの表現手法について述べた。構築した実験システムでは、泳ぎの動きをする人体のモデルにより対話的に波を生成することが可能である。この波はリアルタイムで伝播していき、既にある波に対して対話的に新たな波を干渉させることも可能である。また、泳ぎにより水しぶきを生成することも可能である。水しぶきは重力と慣性に従い自由落下し、新たな波を生成する。

ここで述べた手法はアートワーク *SAP* を第1の目的としたものであり、他の用途に直接適用できるとは限らないが、目的に応じた改良、パラメータの調整を行うことにより、様々な応用が可能であると考えられる。また今後は、他用途への適用も実際に行い、適用可能範囲、問題点を調査してみたいと思う。さらに物理法則を積極的にとりいれるなどして、汎用的な表現手法へと改良していくことも検討したい。

謝辞

最後に、本研究を進めるにあたり有益な議論を頂き、また様々な経験を与えて下さった *SAP* プロジェクトのメンバー、HCT lab, ECE, UBC の皆さんに感謝致します。

参考文献

- [1] Misheff, A.: "Swimming Across the Atlantic," Ciesse Piumini, Milano. (1982)
- [2] Michael Kass, Gavin Miller: "Rapid, Stable Fluid Dynamics for Computer Graphics," Proc. SIGGRAPH 90, pp.49-57 (1990)
- [3] N. Chiba, S. Sanakanishi, K. Yokoyama, I. Ootawara: "Visual Simulation of Water Currents Using a Particlebased Behavioural Model," The Journal of Visualization and Computer Animation, Vol.6, pp155-171 (1995)
- [4] Jim X. Chen, Niels da Vitoria Lobo, Charles E. Hughes, J. Michael Moshell: "Real-Time Fluid Simulation in a Dynamic Virtual Environment," IEEE CG&A, Vol.17, No.3, pp.52-61 (1997)
- [5] Jos Stam: "Stable Fluids," Proc. SIGGRAPH 99, pp.121-128 (1999)
- [6] 舟橋 健司, 岩堀 祐之: "仮想容器による仮想液体の対話操作モデルと一実現法", 日本バーチャルリアリティ学会論文誌, Vol.5, No.4, pp.1087-1094 (2000)
- [7] Kenji Funahashi, Yuji Iwahori: "Virtual Liquid Manipulation Using General Shape Vessel," Proc. IEEE VR2001, pp.283-284 (2001)
- [8] Kenji Funahashi, Yuji Iwahori: "Representation of Swinging Liquid on Virtual Liquid Manipulation," Proc. ICAT2001, pp.179-184 (2001)
- [9] Tzu-Pei Grace Chen, Yuichiro Kinoshita, Yasufumi Takama, Sidney Fels, Kenji Funahashi, Ashley Gadd: "Swimming Across the Pacific: A

Virtual Swimming Interface," Proc. ACM SIGGRAPH 2004, Emerging Technology (2004)

- [10] Sidney Fels, Yuichiro Kinoshita, Tzu-Pei Grace Chen, Yasufumi Takama, Kenji Funahashi, Ashley Gadd: "Swimming Across the Pacific: A Swimming Interface for Virtual Reality," IEEE Computer Graphics and Applications, Vol.25, No.1, pp.24-31 (2005)
- [11] Sidney Fels, Yuichiro Kinoshita, Steve Yohanan, Sachio Takahashi, Yasufumi Takama, Kenji Funahashi, Tzu-Pei Chen, Ashley Gadd: "Swimming Across the Pacific - University of British Columbia," Guide of IMAGINA2005, p.48 (2005)
- [12] Jens Schneider, Rudiger Westermann: "Towards Real-Time Visual Simulation of Water Surfaces," Proc the Vision Modelling and Visualization Conference 2001, pp.211-218 (2001)
- [13] William T. Reeves, Ricki Blau: "Approximate and probabilistic algorithms for shading and rendering structured particle systems," Computer Graphics Vol.19, No.3, pp.313-322, SIGGRAPH'85 (1985)

(2004年12月8日受付)

[著者紹介]

舟橋 健司 (正会員)



1998年名古屋大学大学院情報工学専攻修了、同年名古屋工業大学情報処理教育センター助手、2000年同情報メディア教育センター助手、2002年同センター講師、2005年同センター助教授、現在に至る。バーチャルリアリティ、コンピュータグラフィックスに関する研究に従事(工学博士)。

シドニー・フェルズ



1990年トロント大学コンピュータサイエンス終了、Ph.D.取得、1996-1997年ATR客員研究員、1998年よりブリティッシュコロンビア大学助教授。ヒューマンコンピュータインタラクション、ニューラルネットワーク、インテリジェントエージェント、インタラクティブアートの研究に従事。

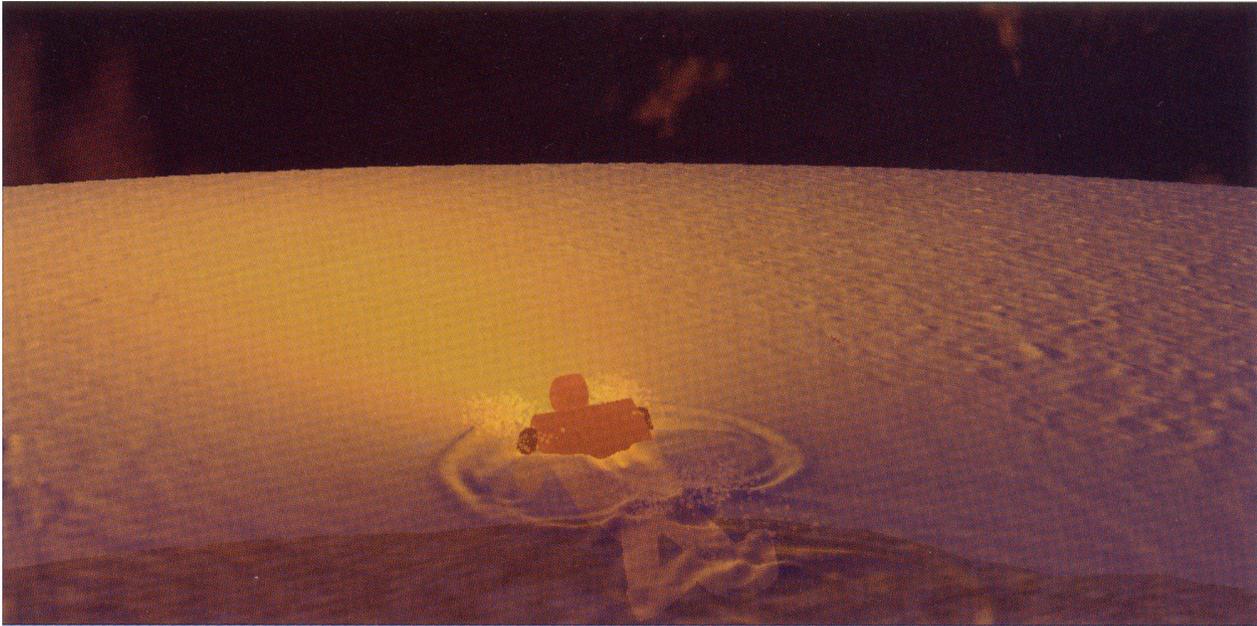


図 14 SAP 表示画面、夕日



図 15 SAP 表示画面、海底

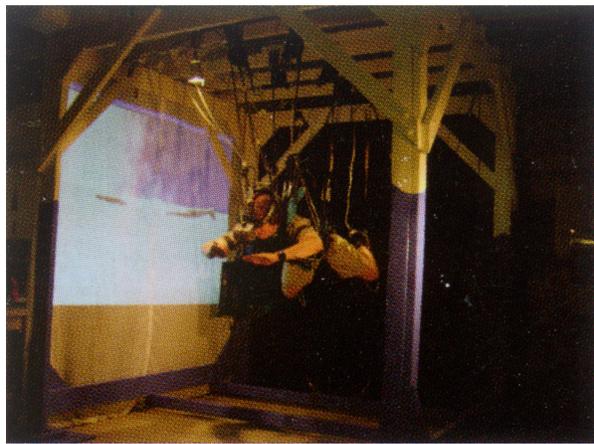


図 16 SAP システムの概観