

2004年度 卒業論文

リアルタイム3DCGにおける
水しぶきを考慮した水面のシミュレーション

指導教員：渡辺 大地 講師

メディア学部 3DCG コンポーネントプロジェクト

学籍番号 M0101170

笹野 英司

2004年度 卒業論文概要

論文題目

リアルタイム3DCGにおける
水しぶきを考慮した水面のシミュレーション

メディア学部

学籍番号：M0101170

氏名

笹野 英司

指導
教員

渡辺 大地 講師

キーワード

3DCG、リアルタイムレンダリング、
水面、水しぶき

コンピュータハードウェアの顕著な進歩により、リアルタイム3DCGによる表現の幅にも大きな躍進が見られ、様々な自然現象をリアルに表現出来るようになった。それに伴い、水に関する表現も非常に豊かなものになって来ている。しかし、その多くは水面だけをリアルに表現したものであり、水しぶきまで考慮した水面のリアルタイム3DCGシミュレーションは極めて少ない。

昨今のゲーム表現を代表とするリアルタイムコンテンツにおいて、その景観をリアルに演出する為に、水面に水しぶきが上がるシーンが多々ある。しかし、水面に発生する水しぶきをリアルに表現するには非常に処理コストが掛かり、リアルタイム3DCGには不向きである。その為、ゲーム表現における水しぶき表現はテクスチャの利用がメインで簡略化した表現が多く、リアルさに欠けている。

本研究は、昨今のリアルタイムシミュレーションにおいて主流となっている水面モデルの表現に加え、新たに水面モデルと同形状の水しぶきモデルを作成し、水面モデルとは異なった変位を与えるという手法を採用した。水しぶきモデルは水面と一体化している部分の表現に使用し、一体化していない中空に飛び散る水粒子の表現にはパーティクルを用いることで、リアルな水しぶき表現をリアルタイムレンダリングにおいて実現することが出来た。本研究では、比重の重い物体が水面に落下するシーン、比重の軽い物体が水面に落下するシーン、水面上を移動する物体が起こす水しぶきの3つのシーンに分けて、水しぶきモデルにそれぞれ固有の変位を与えることでシミュレーションを行った。そして、各シーンに対し、実写と比較することで、考察と検証を行った。

目次

第1章	序論	1
1.1	はじめに	1
1.2	章構成	2
第2章	水を表現するための既存手法	3
2.1	水面モデルを主体として用いる手法	3
2.1.1	水面モデルの表現手法	3
2.1.2	水面モデルに付加する水しぶき表現の既存手法	4
2.2	流体力学に基づいた水流全体の表現手法	5
2.3	本研究で用いる手法	5
第3章	水面モデルの作成	7
3.1	水面モデルの操作手法	7
3.2	水面モデルの境界線における操作	9
第4章	水しぶきモデルの作成	11
4.1	水しぶきモデルの実装	11
4.2	水しぶきモデルの挙動制御	12
4.2.1	制御手法の前提	12
4.2.2	比重の重い物体が水面に落下した場合	13
4.2.3	比重の軽い物体が水面に落下した場合	17
4.2.4	水面上を移動する物体が起こす水しぶきの挙動制御	18
4.3	パーティクル表現の実装	19
4.4	水しぶきのモデルと水面モデルの相互作用	21
第5章	評価と考察	22
5.1	評価	22
5.2	考察	23
5.2.1	考察の前提	23
5.2.2	比重の重い球体が水面に落下するシーン	24
5.2.3	比重の極端に軽い球体が水面に落下するシーン	25
5.2.4	球体が水面を移動するシーン	26

第6章	まとめ	27
	謝辞	29
	参考文献	30

第 1 章

序論

1.1 はじめに

コンピュータハードウェアの顕著な進歩により、3DCG による表現の幅にも大きな躍進が見られ、様々な自然現象をリアルに表現出来るようになった。それに伴い、水に関する表現の幅も向上し、非常に豊かな表現が可能となって来ている。水の表現を 3DCG で行う手段には、厳密なシミュレーションを利用する方法 [1][2][3] と擬似的で簡略化されたシミュレーションを利用する方法 [4][5] がある。前者は CG 映画や海洋実験を代表するプリレンダリングの世界で利用されることが多く、後者はゲーム表現を代表とするリアルタイムコンテンツで利用されることが多い。

昨今のリアルタイムシミュレーションでは水を表現する場合に、水流全体を考慮するのではなく、水面のみをモデル化することでその計算コストを可能な限り抑さえ、リアルタイム性を保持したシミュレーションを行うことが主流となっている [6][7][8]。このような水面のみをモデル化したリアルタイムシミュレーションでは、水流が見せる水しぶきのようなダイナミックな表現までリアルにシミュレートすることが現状では困難である。しかし、インタラクティブ性が重要であるゲーム表現等のリアルタイムコンテンツにおいては、アクションゲームにおけるキャラクターが水に侵入するシーン、レースゲームにおけるモーターボートが水上を疾走するシーンなど、水面に水しぶきが発生するシーンが多々あり、景観をリアルに表現するための水しぶき表現には多くの需要がある。

昨今のゲーム表現を代表とするリアルタイムコンテンツにおいて、水しぶきの表現は水面モデルとは別に水しぶき専用のモデルを配置する方法が一般的である。配置した水しぶきを表すモデルはテクスチャ表現が一般的である。この方法は水面モデルとは完全に独立したモデルとして表現しているため、水面モデルとの一体感がない。更に、その挙動およびレンダリングも非常に擬似的で簡略化した表現手法でリアルさに欠けている。水しぶきをリアルに表現する手法をリアルタイムレンダリングにおいて実現することが可能ならば、ゲーム表現等のリアルタイムコンテンツにおいて、水の表現の幅が大幅に向上し、更なる景観の臨場感を演出することが期待できる。

本研究は一般的なパーソナルコンピュータにおいてリアルな水しぶきを考慮した水面をリアルタイム 3DCG で表現することを目的とする。その提案手法は水面モデルを作成した後、水面モデルと同じ形状のモデルを作成し、新たに作成したモデルは水しぶきを表現するために用い、水面モデルとは別の変位と挙動を与える。また、中空に飛び散る水粒子の表現にパーティクルを利用し、水しぶきのダイナミックな演出効果を助ける役割を果たす。

1.2 章構成

本論文では第 2 章において、水表現する具体的な既存手法と本研究で用いる手法について述べる。第 3 章において水面モデルの実装とその操作手法、第 4 章においては水しぶきの表現の実装・操作手法を記述する。そして、第 5 章で実装した結果の検証と考察を行う。以上が本研究における論文構成である。

第 2 章

水を表現するための既存手法

2.1 水面モデルを主体として用いる手法

2.1.1 水面モデルの表現手法

水のシミュレーションを行う上で、水面を薄い膜のようにモデル化し、モデルを変形して水を表現する手法がある。この手法は非常に単純な計算で水面の波をリアルに表現することが可能なので、昨今の水のリアルタイムシミュレーションを行うに当たって主流となっている [7][8]。

水面を薄い膜のようにモデル化し、モデルを変形させて水を表現する手法として一般的なものは水面をメッシュ状に区切ったモデルを作成する手法 [6] である。図 2.1 は実際にメッシュ状に分割して生成した水面モデルの例である。そのメッシュの各頂点が高さ方向のみへ移動することにより、波の発生を表現する。

水面を薄い膜のようにモデル化し、モデルを変形して水を表現する手法で表現できるものは水面のみのシミュレートであり、水流が見せる水しぶきの様なダイナミックな運動までシミュレートすることが出来ない。それは、水面モデルの制御上、水面モデルが高さ方向に重なり合うようなシーンを表現することが不可能な為である。

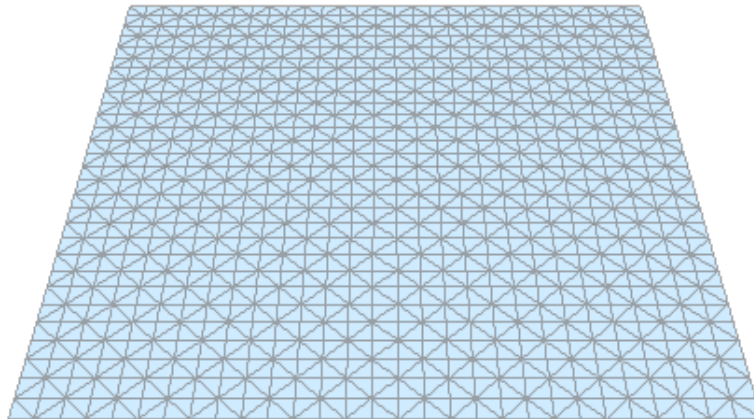


図 2.1: メッシュ状に分割した水面モデル

2.1.2 水面モデルに付加する水しぶき表現の既存手法

昨今のリアルタイム 3DCG コンテンツでは、ダイナミックな水流の運動を表現するために、水面モデルに対し、水しぶきの表現を付加している。リアルタイムレンダリングにおいてはレンダリングの計算コストを可能な限り押さえることが必要な為、非常に簡略化した水しぶきの表現が主流となっている。その表現手法には一般的に 2 つある。第一に水柱の様なテクスチャで円柱を描くように水面モデルに貼り付けた水しぶき表現 [9] がある。この手法では水しぶきを単なる一枚絵として表現しており、水面に迸る水しぶきをダイナミックに表現することが可能だが、モデルの形状が一定な為、水しぶきの挙動をリアルに再現することが出来ない。また、水しぶきが中空に飛び散るような表現も不可能である。第二に中空

に配置されたテクスチャで水粒子の表現を付加したもの [10] がある。この手法は、中空に飛び散る水粒子をダイナミックに表現することが可能だが、水しぶきを持つ特有の滑らかな曲面を表現することが不可能で、水面モデルとの一体感にも欠け、水しぶき個々の挙動表現にも不正確さが目立っているのでリアルさに欠けている。

2.2 流体力学に基づいた水流全体の表現手法

水流全体の表現や、水流を考慮した水中を移動する物体の制御が可能な計算方法で水流をシミュレーションする手法がある。

流体力学に基づいた水流全体の表現手法として代表的なものがナビエ・ストークス方程式 [11] を利用し、ボクセルを制御する手法 [1][2][3] と水粒子をパーティクルとして表現し、個々のパーティクルを制御する粒子法 [12] である。この代表的な両者を初めとする流体力学の計算を考慮した表現手法では、水流全体をリアルにシミュレートすることが出来るが、膨大な処理コストを要求するため、リアルタイムレンダリングには非常に不向きである。その為、CG 映画や海洋科学分野などにおける実験などリアルタイム性を重んじない場合に用いることが多い。

2.3 本研究で用いる手法

本研究では、水しぶきの表現を水面モデルに付加するという手法を採用する。本手法では水面モデルと同じ形状のモデルを作成し、水しぶきを表現するために用いる。新たに作成したこのモデルに水面モデルとは別の変位と挙動を与える。図 2.2 は水面モデルの下部に水しぶきモデルを配置し、挙動を与えたイメージ図である。この様に水しぶきモデルに変位を与えることで、滑らかでダイナミック且つ、水面モデルとの一体感を表現することが出来る。また、中空に飛び散る水粒子を

再現するため、パーティクル表現を補足的に用いる。それにより、水しぶきのダイナミックな演出効果を助ける役割を果たす。

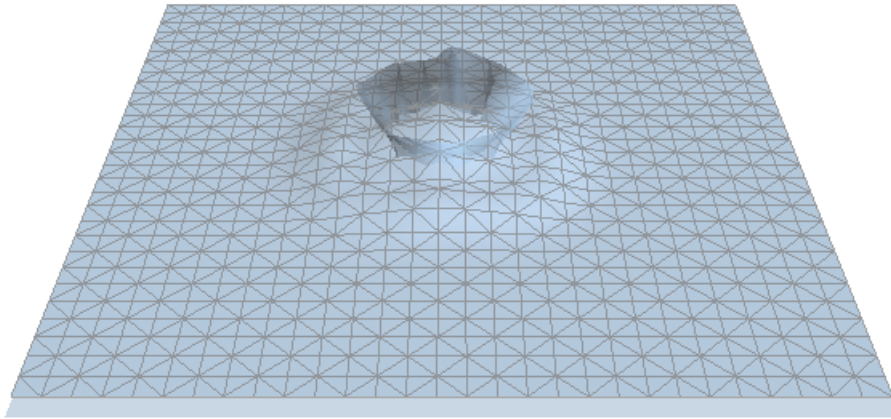


図 2.2: 水面モデルの下部に配置した水しぶきモデルが変位している様子

第 3 章

水面モデルの作成

3.1 水面モデルの操作手法

本研究では水面を表現するために、Miguel Gomez の手法 [6] を参考にした薄い膜のようにモデル化した水面モデルを作成する。ここで作成する水面のモデルは等間隔で分割したメッシュ状のモデルである。図 3.1 はこの手法で作成した水面モデルである。

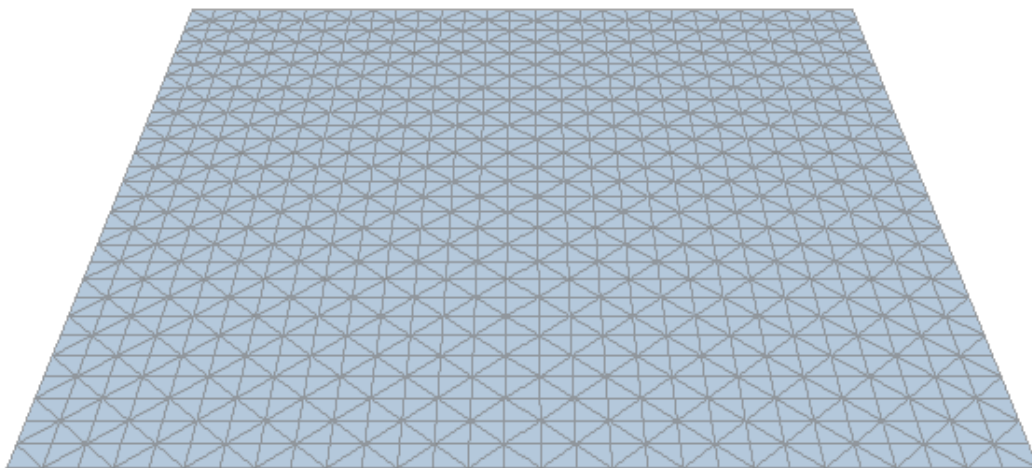


図 3.1: 30×30 の等間隔で分割した水面モデル

このメッシュを構成している水面モデルの各頂点を上下に変位することで、水

面モデルに起こる波をシミュレートする。 xz 座標を横方向、高さは y 座標とする。水面モデルの各頂点における変位計算には二次元波動方程式を使用する。具体的な二次元波動方程式は次式で表すことができる。

$$\frac{\partial^2 y}{\partial t^2} = v^2 \left(\frac{\partial^2 y}{\partial x^2} + \frac{\partial^2 y}{\partial z^2} \right) \quad (3.1)$$

(3.1) 式の t は時刻で v は波が水面上を進む速度である。

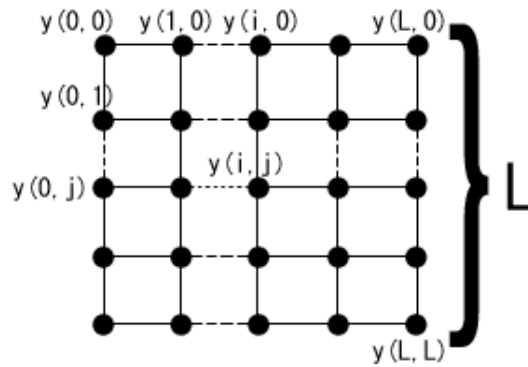


図 3.2: $L \times L$ の水面モデルの図

水面モデルの x 軸方向に i 番目、 z 軸方向に j 番目の頂点の高さを $y(i, j)$ とする。さらに h をメッシュ分割の間隔とする。ここで、 $y_n(i, j)$ を x 方向に i 番目、 z 軸方向に j 番目の頂点における時刻 t_n での高さとする。図 3.2 は $L \times L$ メッシュにおける水面モデルの図である。(3.1) 式を中心差分で近似した (3.2) 式を以下に示す。

$$\frac{y_{n+1}(i, j) - 2y_n(i, j) + y_{n-1}(i, j)}{\Delta t^2} = v^2 \frac{y_n(i+1, j) + y_n(i-1, j) + y_n(i, j+1) + y_n(i, j-1) - 4y_n(i, j)}{h^2} \quad (3.2)$$

$y_{n-1}(i, j)$ と $y_{n+1}(i, j)$ は各々時刻 t_{n-1} と t_{n+1} における高さである。(3.2) 式を $y_{n+1}(i, j)$ について解いたものを (3.3) 式に示す。

$$y_{n+1}(i, j) = \frac{v^2 \Delta t^2}{h^2} (y_n(i+1, j) + y_n(i-1, j) + y_n(i, j+1) + y_n(i, j-1)) + \left(2 - \frac{4v^2 \Delta t^2}{h^2} y_n(i, j) - y_{n-1}(i, j) \right) \quad (3.3)$$

3.2 水面モデルの境界線における操作

水面モデルの境界線における各頂点を変位する計算式は (3.3) 式の一部を変更したものを利用する。(3.3) 式の右辺第二項の括弧内にある、隣接する頂点の高さ情報のうち、境界線の頂点として存在しないものを除外する。(3.4) 式は図 3.2 における、左側の角部分を除いた境界線部分における頂点を変位する計算式である。

$$y_{n+1}(i, j) = \frac{v^2 \Delta t^2}{h^2} (y_n(i+1, j) + y_n(i, j+1) + y_n(i, j-1)) + \left(2 - \frac{3v^2 \Delta t^2}{h^2} y_n(i, j) - y_{n-1}(i, j) \right) \quad (3.4)$$

次に水面モデルの四隅にあたる角の計算方法を述べる。水面モデルの四隅では隣接する頂点が 2 つしか存在しない。(3.5) 式は図 3.2 における、左上部の角にあたる頂点を変位する計算式である。

$$y_{n+1}(i, j) = \frac{c^2 \Delta t^2}{h^2} (y_n(i+1, j) + y_n(i, j+1)) + \left(2 - \frac{2c^2 \Delta t^2}{h^2} y_n(i, j) - y_{n-1}(i, j) \right) \quad (3.5)$$

第 4 章

水しぶきモデルの作成

4.1 水しぶきモデルの実装

水面モデルと同様に水しぶきを表現するためのモデルも薄い膜のようにモデル化する。しかし、水面モデルと同様の分割数では分割数が少なすぎて、水しぶきに見える滑らかな曲面が表現できない。そのため、水しぶきモデルでは可能な限りメッシュの分割数を増やす必要がある。図 4.1 に実際にこの手法で実装した 90×90 で分割した水しぶきモデルを示す。

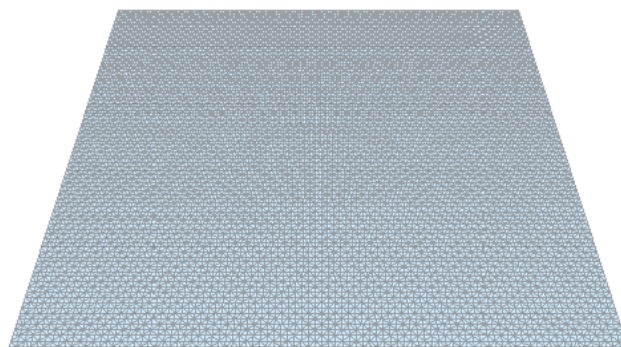


図 4.1: 90×90 でメッシュ分割した水しぶきモデル

単に水面モデルと水しぶきモデルを重ねるように配置した場合、水面モデルに発生する波の変位が起こる際に下部に配置した水しぶきモデルが現れてしまう。その不自然な景観の表現を回避するために水しぶきモデルを水面モデルの下方に配置する。図 4.2 は水しぶきモデルを水面モデルの下方に配置したものである。

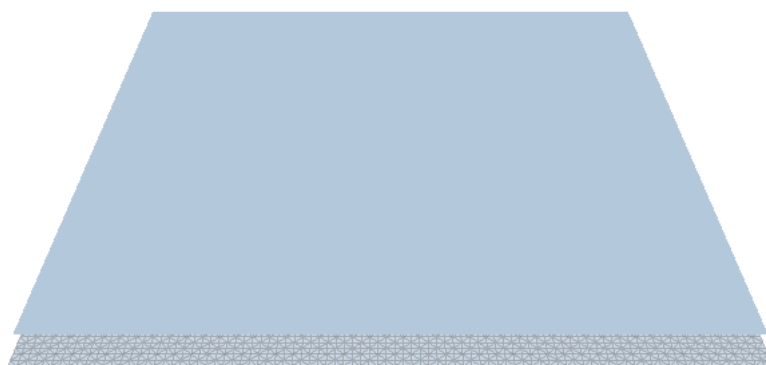


図 4.2: 水面モデルと水しぶきのモデルを配置した様子

4.2 水しぶきモデルの挙動制御

4.2.1 制御手法の前提

本研究において、水面に外力が加わり、水しぶきが発生するシーンとして、水面に物体が落下した状況と、物体が水面を移動する状況を想定する。

水より比重の重い物体により発生する水しぶきの挙動と水より比重の軽い物体により発生する水しぶきの挙動の二種類に分けて、物体が水面に落下した場合に起こる水しぶきの挙動制御をする。

水面に外力を与える物体として、本研究では球体を例として実装する。

4.2.2 比重の重い物体が水面に落下した場合

水より比重の重い物体、つまり落下した際に水中に沈んでしまう物体が水面に落下した際に起こる水しぶきの挙動制御手法を提案する。比重の重い物体は水面に落下した際に、急激に物体が水中に進入する為、物体の外周付近から第一の水しぶきが発生する。図 4.3 はその第一の水しぶきの発生を表現した図である。ここで、この物体は水より比重が重く、水に沈む現象が起こる。その時、物体の水中移動により水面が一瞬大きくめり込む現象が起こる。図 4.4 は物体が水面に侵入した瞬間の様子である。そして、めり込んだ部分に両端の水が流れ出し、その衝突により第二の水しぶきが上がる。図 4.5 はその第二の水しぶきが発生した様子である。この一連の運動は水のような粘性が小さいサラサラの液体において、流体力学的観点からも数式上この様な運動を取る [13]。

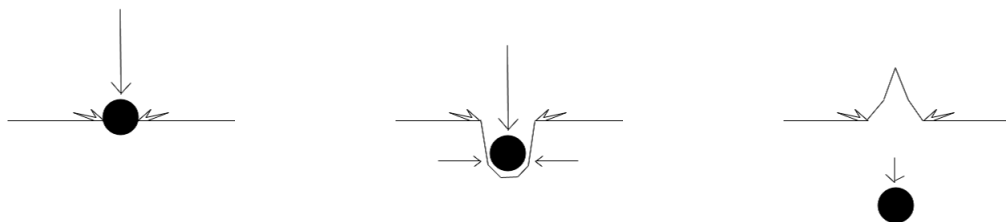


図 4.3: 第一の水しぶき発生 図 4.4: 水面のめり込み 図 4.5: 第二の水しぶき発生

初めに、本研究において、図 4.3 に示した第一の水しぶきを表現するために、水しぶきモデルを変位する。水しぶきモデルの変位する頂点は水面モデルに外力を与える物体の最外周の近傍の頂点を取得する。

落下する物体の大きさや形状によって、与える変位頂点の数が異なる。ここでは、大小異なるサイズの球体を例として、説明していく。下記の図 4.6 は水しぶきモデルのグリッド間隔 d に対し $2d$ の直径を持つ球体が点 A_0 に落下したときに変位する頂点、図 4.7 は同スケールの水しぶきモデルにおいて、 $4d$ の直径を持つ球体が点 A_0 に落下したときに変位する頂点を示したものである。

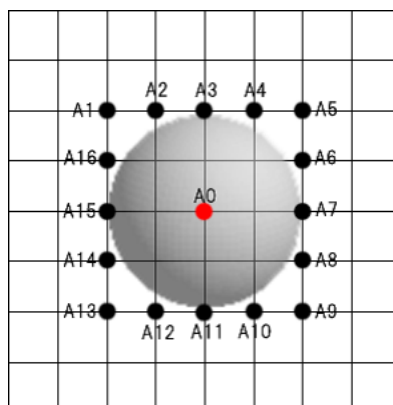
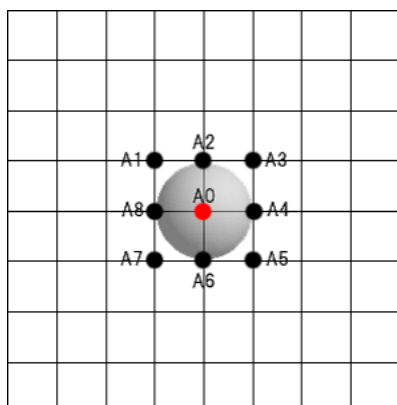


図 4.6: 直径 $2d$ の球体が点 A_0 に落下したときの変位頂点
 図 4.7: 直径 $4d$ の球体が点 A_0 に落下したときの変位頂点

変位する頂点を $A_n(x_n, y_n, z_n)$ とする。図 4.6 においては A_0 の周囲 1 グリッドのスケールに存在する近傍点 ($A_1 \sim A_8$) は変位する頂点である。また、図 4.7 においては A_0 の周囲 2 グリッドのスケールに存在する近傍点 ($A_1 \sim A_{16}$) は変位する頂点である。流体力学的に見て、水しぶきは発生の際に初速度を持っている [13] が、本研究では水のシミュレートとして水面のみを扱っている為、水圧や水流の計算までカバーすることが出来ない。そのため、水しぶきが発生する際にその発生初速度の正確な計算が不可能である。そこで、落下する球体の重さ及び速度から水しぶきの初速度を決定し、その計算を単純化している。また、乱数を初速度に乗算することで、各頂点の変位値を異なるようにし、実際の水しぶきを持つ景観に近似する。実際の水しぶきのように、水しぶきモデルが逆円錐状に放射する各頂点座標の計算を以下に示す。

ここで Δx_n 、 Δy_n 、 Δz_n は頂点 A_n における初速度ベクトルの各成分である。但し、 $n = 1, \dots, \infty$ とする。

$$\Delta x_n = \begin{cases} -\frac{\alpha MV}{2} & (x_n > x_0, |x_n - x_0| = |z_n - z_0|) \\ \frac{\alpha MV}{2} & (x_n < x_0, |x_n - x_0| = |z_n - z_0|) \\ \alpha MV & (x_n > x_0, |x_n - x_0| > |z_n - z_0|) \\ -\alpha MV & (x_n < x_0, |x_n - x_0| > |z_n - z_0|) \\ 0 & (|x_n - x_0| < |z_n - z_0|) \end{cases} \quad (4.1)$$

$$\Delta y_n = \alpha MV \quad (4.2)$$

$$\Delta z_n = \begin{cases} -\frac{\alpha MV}{2} & (z_n < z_0, |x_n - x_0| = |z_n - z_0|) \\ \frac{\alpha MV}{2} & (z_n > z_0, |x_n - x_0| = |z_n - z_0|) \\ \alpha MV & (z_n > z_0, |x_n - x_0| < |z_n - z_0|) \\ -\alpha MV & (z_n < z_0, |x_n - x_0| < |z_n - z_0|) \\ 0 & (|x_n - x_0| > |z_n - z_0|) \end{cases} \quad (4.3)$$

ここで M は落下する球体の重さであり、 V は球体の速度である。また α は変数であり、0.1~1.0までの値を取る乱数である。この時、各処理ステップに対し、水しぶきモデルが変位する速度に重力加速度を加え放物運動を取るように計算する。

図 4.8 は実際に上記の式を実装して変位を加えた水しぶきモデルである。

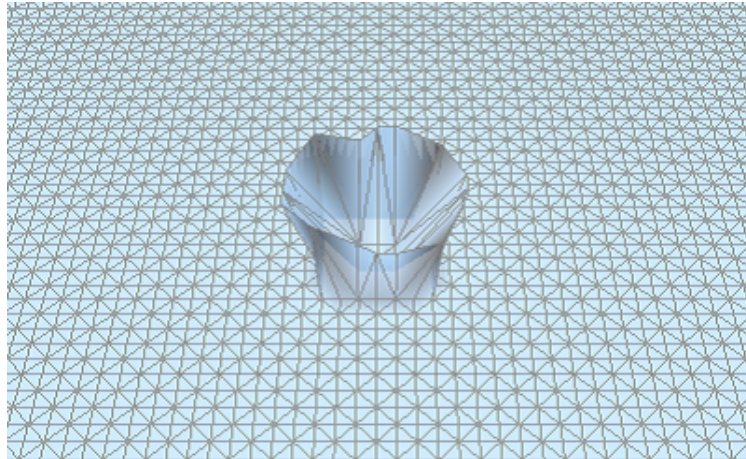


図 4.8: 実装した水しぶきのモデルの挙動

物体が水面に落下した際に発生する第一の水しぶきの挙動を制御した後、水面の変位が元に戻る際に起こる第二の水しぶきを表現する。第二の水しぶきは球体の外周の近傍点から発生するのではなく、球体の落下地点における近傍点からほぼ垂直に初速度を持ち発生する為、第二の水しぶきを表現するためには、球体の落下地点における近傍の頂点 A_0 における y 座標方向のみを変位すればよい。次式がその計算式である。

$$\Delta y_0 = \alpha MV \quad (4.4)$$

ここで M は落下する球体の重さであり、 V は球体の速度である。また、 G は重力加速度である。また α は変数であり、 $0.1 \sim 1.0$ までの値を取る乱数である。この時、各処理ステップに対し、水しぶきモデルが変位する速度に重力加速度を加え放物運動を取るように計算する。

実装した結果を図 4.9 に示す。

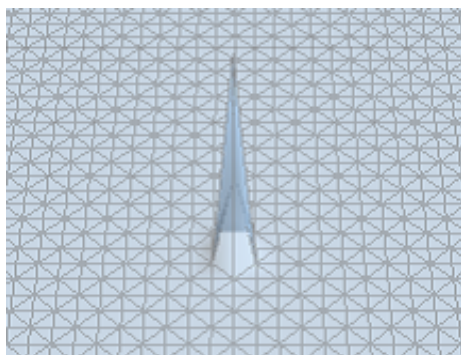


図 4.9: 実装した第二の水しぶきの挙動

4.2.3 比重の軽い物体が水面に落下した場合

水より比重の軽い球体、つまり落下した際に水面に浮かぶ球体が水面より上空から落下した際に起こる水しぶきの挙動制御手法を提案する。比重の軽い球体も比重の重い球体と同様に、水面に外力を与えた際に、一瞬だが球体が水面を小さくめり込ませることで球体の外周付近から第一の水しぶきが発生する。図 4.10 は第一の水しぶきが発生する様子である。ここで、比重の軽い球体は水の浮力を受け、水中に進入することなく、水面においてその座標を固定する。図 4.11 は球体が水面に浮かんでいる様子である。球体が水中に進入することがない為、第二の水しぶきは発生せず、第一の水しぶきだけ発生する。水より比重の軽い、水面に浮かぶ球体では、第一の水しぶきを表現するための制御手法のみ計算すればよい。

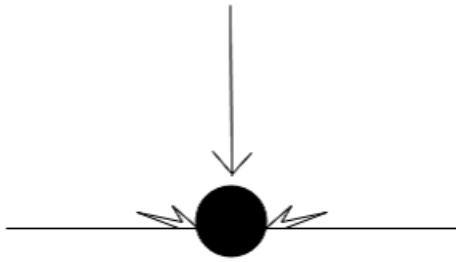


図 4.10: 第一の水しぶき発生

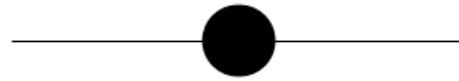


図 4.11: 球体は浮力を受け浮かび続ける

4.2.4 水面上を移動する物体が起こす水しぶきの挙動制御

モーターボートが水上を走行するような、水面に一定方向から外力が加わり続けた際に起こる水しぶきの様子を表現するための制御を行う。水面を移動する物体は連続的に進行方向に対し水面に外力を与える為、水面を切り裂くように水しぶきが発生する。基本的には物体が落下する際に発生する第一の水しぶきと同様の挙動制御を行う。ただし、外周近傍の頂点の内、物体の進行方向とは逆方向に存在する頂点には変位を与えない。

例として、本研究で用いる球体が水面モデル上において $+x$ 方向に平行移動した場合の水しぶきモデル制御を考える。この場合、球体の進行方向とは逆の $-x$ 方向以外の頂点に対し、第一の水しぶきモデルの挙動計算式を適用すればよい。実際に実装した様子を図 4.12 に示す。図 4.12 における球体から伸びている矢印は球体の進行方向を示す。

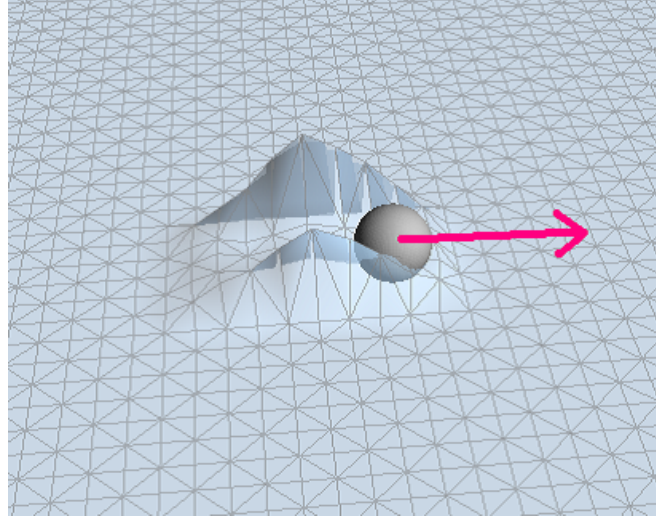


図 4.12: 水面を切り裂くように発生する水しぶき

4.3 パーティクル表現の実装

実際の水しぶきは表面張力が働き、水面と一体化している部分と、表面張力が働かない中空に飛び散る水粒子の部分が存在するが、メッシュに分割した水しぶきモデルを変位するだけでは、水しぶきが水面と一体化し続ける様子しか表現できない。そこで、パーティクル表現を利用し、中空に飛び交う水粒子の表現をデコレートし、発生する水しぶきにリアリティを与える。

パーティクルの初期位置は落下した物体の外周から発生するように設定する。各パーティクルが物体の落下地点付近にランダムで飛び散るように初速度を設定した。また、各パーティクルに働く重力を考慮し、自然な放物運動を取るようにした。各パーティクルが持つ初速度ベクトルを $Q(Q_x, Q_y, Q_z)$ とし、各成分は次式のように設定した。

$$Q_x = \alpha'_0(V_x + V_y + V_z) \quad (4.5)$$

$$Q_y = \alpha'_1(V_x + V_y + V_z) \quad (4.6)$$

$$Q_z = \alpha'_2(V_x + V_y + V_z) \quad (4.7)$$

ここで $\alpha'_0 \sim \alpha'_2$ は個別の変数であり、それぞれ 0.0 ~ 1.0 までの値を取る乱数である。 V_x, V_y, V_z は球体が持つ速度 V の各成分である。この時、各処理ステップに対し、各パーティクルが持つ速度に重力加速度を加え放物運動を取るよう計算する。図 4.13 に実際にパーティクル表現を水しぶきのモデルを変位したシーンにデコレートしたものを示す。

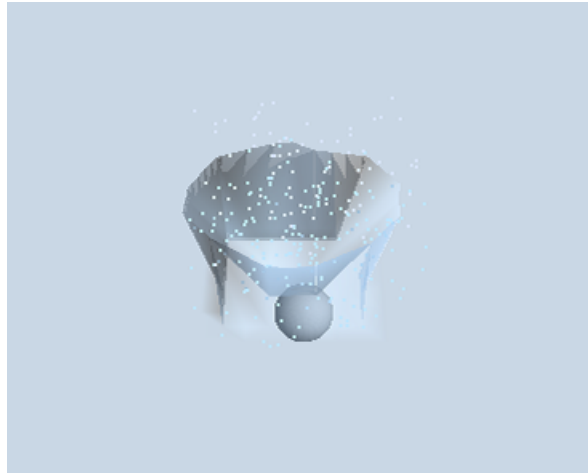


図 4.13: 水しぶき表現にパーティクル表現を加えた図

4.4 水しぶきのモデルと水面モデルの相互作用

水面に外力が加われば、水しぶきが発生すると同時に、その地点から波も発生する。波の挙動計算は第3章で示した手法により水面モデルに実装した二次元波動方程式により制御する。ここでは、水しぶきが上がった地点に対し、水面モデルにおける近傍点の y 座標を瞬間的に変位することで水しぶきの発生による水面モデルにおける波の生成を表現する。また、水しぶきモデルと水粒子の表現に利用したパーティクルが水面モデルに着水した際にも波紋が発生するように制御する。 y 座標の変位値はそれぞれ微量でもリアルな景観を演出できる。図 4.14 は水しぶきが発生する際に起こる水面波の発生シーンであり、図 4.15 はパーティクルが水面に着水した際に起こる水面の波紋を表している。

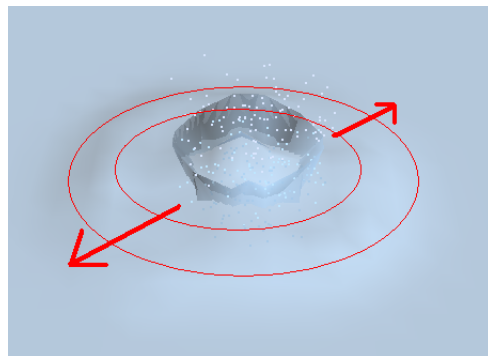


図 4.14: 水しぶきの発生による水面モデルにおける波の発生

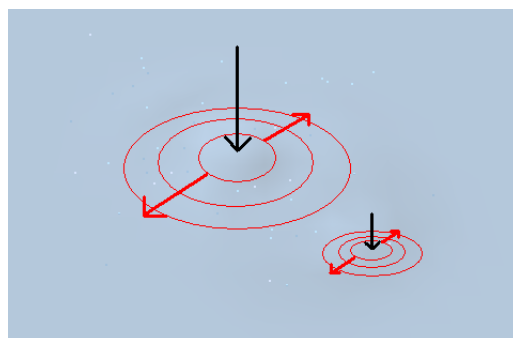


図 4.15: 水しぶきの着水における水面モデルにおける波の発生

第 5 章

評価と考察

5.1 評価

本研究では、3次元グラフィックツールキットである「Fine Kernel Tool Kit」[14]においてプログラミングし、作成したアプリケーションの動作環境はCPUはPentium4 2.8CGHz、メモリは512MBにおいて行った。

実際に、本研究の手法で30×30で分割した水面モデル・90×90で分割した水しぶきのモデル・直径が水しぶきモデルの2グリッド分の幅を持つ球体の3つのモデルを実装し、動作した結果、極端な動作遅延が起こることなく、常時30fps以上を保つリアルタイム処理に適應することが出来た。水面モデルの変位、水しぶきのモデルの変位及び、パーティクル表現を実際に実装した図5.1を以下に示す。

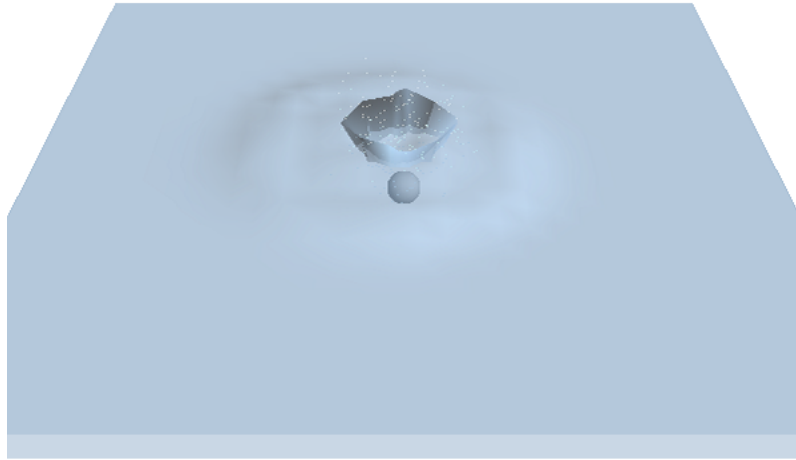


図 5.1: 実際に実装した図

また、本研究において、水しぶきモデルのメッシュ分割数を 90×90 以上に増加した結果、 111×111 に分割した時点でフレームレートが急激に下がり、リアルタイムでレンダリングするには困難な結果となった。今回の動作環境はオンボードのチップセットを利用したので、高性能な GPU を組み合わせて利用すれば、より快適な動作を示すだろう。

5.2 考察

5.2.1 考察の前提

本章では、実際に水しぶきが上がる瞬間の写真と本研究において実装したシミュレーションのワンシーンを見比べて、その景観の考察を行う。用意する実際の写真は第一に比重の重い球体が水面に落下するシーン。第二に比重の極端に軽い球体が水面に落下するシーン。第三に球体が水面を移動するシーンを用意して考察する。

5.2.2 比重の重い球体が水面に落下するシーン



図 5.2: 実際の水しぶき



図 5.3: 本研究における表現

比重の重い球体が水面に侵入した際に発生する第一の水しぶきの発生シーンについて考察する。図 5.2 は実際の写真、図 5.3 は本研究における第一の水しぶきが発生するシミュレーションのシーンである。

図 5.2 における実際の写真でも表面張力が働いて、水面と一体化している様子がわかる。図 5.3 の本研究における表現でもその表面張力が働いているシーンが見取れる為、水面との一体感を持つ水しぶきの表現は達成したといえる。表面張力から脱出した中空を飛び散る水しぶきの表現もパーティクル表現を追加することで効果的に再現している。しかし、 90×90 グリッドで分割した水しぶきのモデルでは発生する水しぶきの滑らかさが実際の水しぶきよりも荒く、少々リアルさに欠ける結果となった。

比重の重い球体が水面に侵入した際に発生する第二の水しぶきの発生シーンについて考察する。図 5.4 は実際の写真、図 5.5 は本研究における第二の水しぶきが発生するシミュレーションのシーンである。



図 5.4: 実際の水しぶき

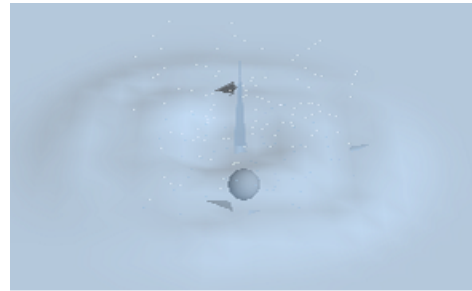


図 5.5: 本研究における表現

図 5.5 は第二の水しぶきの発生座標を 1つのグリッドで制御しているため、図 5.4 の実際の第二の水しぶきのような滑らかな曲面を描くことが出来ずに、景観のリアルさの演出に課題を残す結果となった。

5.2.3 比重の極端に軽い球体が水面に落下するシーン



図 5.6: 実際の水しぶき

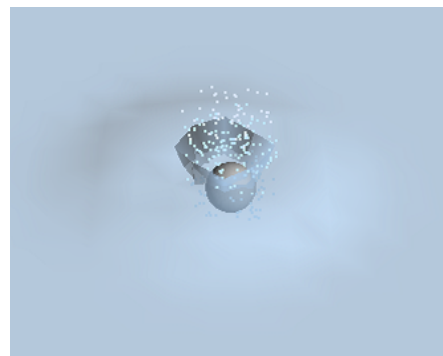


図 5.7: 本研究における表現

図 5.6 は比重の極端に軽い球体 (ピンポン玉) が落下した際に描く水しぶきである。図 5.7 は本研究において、比重の軽い球体を水面に落下するシミュレーションで得た水しぶきである。

比重の重い球体が落下する場合と同様に、図 5.6 に示したような軽い物体が落下

する場合でもその表面張力が働いているシーンや中空に飛び散る水しぶきが見取れる。シミュレーション結果においても、水面との一体感を持つ表面張力による水しぶきの表現やパーティクルで補った飛び散る水しぶきの表現は達成したといえる。

5.2.4 球体が水面を移動するシーン



図 5.8: 実際の水面上を移動する球体

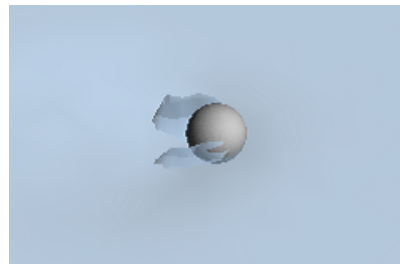


図 5.9: 本研究における表現

図 5.8 は実際に球体を水面上で移動した写真である。図 5.9 は本研究において球体を水面で移動して得たシミュレーション結果である。図 5.8 のように実際に低速で移動した場合には水面はある程度変位するものの、中空に飛び散る水しぶきまで発生していないのがわかる。これは水面に与える外力が非常に小さいからである。本研究で実装した球体もモーターボートのような高速で移動するものではない為、図 5.9 のように中空に飛び散る水粒子の表現を除外した。それにより、自然な水しぶきと波の発生を再現出来ている。

第 6 章

まとめ

本研究ではリアルタイムレンダリングにおいて、水しぶきを考慮した水面のシミュレーションを行った。水面モデルとは別に、水面モデルと同形状の水しぶきモデルを実装し、その水しぶきモデルに水面モデルとは異なった変位を与えると
いう手法を提案することでリアルな水しぶきを演出した。水しぶきモデルは実際の水しぶきの様に表面張力が働き、水面と一体化している部分を表現するために利用する。また、表面張力が働かない中空に飛び散る水しぶきを表現するためにはパーティクルを用いた。本研究では水しぶきが発生する3つのシーンについてシミュレーションを行った。第一に比重の重い物体が水面に落下するシーン、第二に比重の軽い物体が水面に落下するシーン、第三に水面上を移動する物体が起こす水しぶきのシーンである。各シーンに対し、水しぶきモデルに固有の変位を与えることで水しぶきをシミュレーションした。

本研究は簡略化したシミュレーションで、その精度も非常に低いものであるが、非常に単純な計算において、様々な水しぶきが発生する景観を常時 30fps 以上とい速度のリアルタイムレンダリングにおいて演出することが出来た。水しぶきが発生する各シーンについて、実写と比較し、検証を行った。その結果、本研究で用いた 90×90 グリッドの水しぶきモデルでは発生する水しぶきの滑らかさが実際の水しぶきよりも荒く、少々リアルさに欠ける結果となった。しかし、全体的に見て、水しぶきが発生する景観の臨場感を向上することが出来たと考える。今後、コン

コンピュータの性能が現在よりも向上し、リアルタイムレンダリングにおいて水流まで考慮したシミュレートが可能となり、更なる水の表現の向上が可能となることを期待する。

謝辞

本論文を締めくくるにあたり、終始熱心なご指導、ご鞭撻を下さいました渡辺大地講師及び電気通信大学の和田篤氏に心から感謝の意を表します。また、沢山の助言や励ましをくれた研究室の仲間たちや家族に心から感謝いたします。

参考文献

- [1] Douglas Enright, Stephen Marschner, Ronald Fedkiw. “ Animation and Rendering of Complex Water Surfaces ”. *Stanford University*, 2003.
- [2] Nick Foster and Dimitri Metaxas. “ Realistic Animation of Liquids ”. *University of Pennsylvania*, 1996.
- [3] Nick Foster and Ronald Fedkiw. “ Practical Animation of Liquids ”. *SIGGRAPH2001*, 2001.
- [4] Jens Schneider and Rudiger Westermann. “ Towards Real-Time Visual Simulation of Water Surfaces ”. *Aachen University of Technology*, 2001.
- [5] 近藤秀樹. “ 浮遊する剛体の挙動を含めた水面のリアルタイムシミュレーション ”. 東京工科大学, 2003.
- [6] MARK DELOURA, 川西裕幸, 狩野智英. “ GAME PROGRAMING GEMS ”. 株式会社ボーンデジタル, 東京, 2001.
- [7] Kano. “ Kano’s webpage ”. <http://cgi3.tky.3web.ne.jp/~tkano/tlwater.shtml>.
- [8] かみやん. “ Kamiyan’s Homepage ”. <http://www.asahi-net.or.jp/~qs7e-kmy/tawagoto/990914/index.html>.

- [9] 任天堂. “ スーパーマリオサンシャイン ”. <http://www.nintendo.co.jp/ngc/sms/>.
- [10] 任天堂. “ ウェーブレースブルーストーム ”. <http://www.nintendo.co.jp/ngc/gwrj/>.
- [11] 八田圭爾, 鳥居平和, 田口達夫. “ 流体力学の基礎 ”. 日新出版, 東京, 1991.
- [12] 越塚誠一. “ 数値流体力学 ”. 培風館, 東京, 1997.
- [13] 笠原英司, 清水正之, 前田昌信. “ 図解 流体力学の学び方 ”. オーム社, 東京, 1986.
- [14] 渡辺大地. “ Fine Kernel Tool Kit System ”. <http://www.teu.ac.jp/aqua/~earth/FK/>.