

データベースを用いた 水のアニメーション編集

千葉雄太[†] 土橋宜典[†] 山本強[†]

近年、コンピュータグラフィックスの分野において、数値流体解析を利用して流体の挙動を計算することで、リアルな流体のアニメーションを作成することを目的とした研究が盛んに行われている。それにともない、流体の所望の動きを実現することを目的とした研究も数多く行われている。しかし、流体シミュレーションは多くのパラメータに依存しているため、それらの値を調整して所望の流体表現を得ることは容易ではない。本論文では、流体シミュレーションの一つである水のシミュレーションに着目し、多数のシミュレーション結果を組み合わせ、所望の流体の映像表現を実現する手法を提案する。

Editing Water Animation by Using Database

Yuta Chiba[†] Yoshinori Dobashi[†] and Tsuyoshi Yamamoto[†]

Recently, in the field of computer graphics, simulating realistic motion of fluids based on computational fluid dynamics has been studied by many researchers. In addition to these researches, many methods for generating desired motions of fluids have also been developed. However, it is generally difficult to create the desire motion of fluids since the fluid simulation depends on many parameters. In this paper, we focus on the simulation of water and propose a method that enables us to create the desire motion of water by using a pre-computed simulation database.

1. はじめに

コンピュータグラフィックスの分野において、雲、煙、水、炎などの流体の写実的な表現は重要な研究課題の一つであり、これまで盛んに研究が行われてきている。リアルな流体の映像を作成するためには、その外観のみでなく、違和感を与えない流体の挙動も非常に重要な要素となる。近年、非常にリアルな流体の挙動を再現した流体のアニメーションがコンピュータグラフィックスにおいて作成可能となり、ゲームや映画等の映像分野において様々な演出を生み出すために効果的に利用され、その威力を発揮している。

流体のアニメーションを作成する手法は数多く存在する。それらの手法の中でも、数値流体解析を利用して流体の動きを計算する手法は、物理現象に基づいた方程式を数値的に解くことで複雑な流体全体の挙動を算出するため、実際の自然現象における動きに近い非常にリアルな流体のアニメーションを作成することが可能である。そのため、近年は数値流体解析を利用した数多くの手法が確立されてきた。そして、コンピュータにおけるメモリの増加、CPUの発達およびGPUを利用した高速計算により、多くの流体現象に関するシミュレーションが容易に行えるようになった。

上記のような背景のもと、近年ではそれらの流体のシミュレーションに対して、ユーザの意図を反映させたいという欲求が高まってきた。すなわち、流体のアニメーションにおいて、対象としている流体がユーザの所望する動きや所望する形状となるようにしたいという欲求である。しかし、数値流体解析による流体の挙動の計算は多くのパラメータに依存しているため、その実現は容易ではない。そのような要望を達成するための手法もこれまで数多く開発されてきた。それらの手法を利用することにより、流体が動きながら人や動物、文字など、現実にはあり得ない形状を形成する映像の作成が可能となり、映像制作における自由度と容易さは格段に向上した。

本論文では、流体の一つである水の表現、特に、砕波に着目する。映像制作における水の演出では、目的の位置および規模での津波・砕波の発生などのように、自然な動きを保持しつつ、かつ、特定の効果を表現することが望まれる場合が多い。しかしながら、従来の流体シミュレーション手法においては、上述のように多くのパラメータの調整が必要になるため、そのような演出の生成は容易ではない。また、従来の流体シミュレーションを制御することによりユーザの所望する流体表現を生成する手法においても、実際には存在しない仮想的な外力などを発生させることで強制的に流体の挙動を制御するため、津波や砕波のような自然な流れにおいて適用した場合、不自然な挙動になってしまう場合が多い。現状では、映像制作者が使用する流体シミュレータにおいて設定可能な多数のシミュレーションパラメータを試行錯誤的に調整する

[†] 北海道大学 大学院情報科学研究科
Graduate School of Information Science and Technology, Hokkaido University, Japan

ことで目的の映像を作成しており、極めて煩雑な作業が必要となっている。

このような背景から、本論文では、上記の問題を解決するための方法として、あらかじめ流体シミュレーションにより生成した複数のシミュレーション結果を組み合わせることにより、ユーザの所望の水のアニメーションを生成する手法を提案する。提案手法では、まず、前処理としてパラメータを変化させた砕波のシミュレーションを多数実行し、その結果を記憶しておく。ただし、前処理におけるシミュレーションは二次元で行う。そして、それらのシミュレーションの結果を三次元空間に平行に配置し、その間を補間することにより三次元での砕波のシミュレーションを再現する。ユーザは波の進行方向に対して垂直な断面での砕波の形状を入力できる。これにより、砕波の形状を直感的に編集することが可能となる。

以下、まず2節において、いくつかの関連研究について議論する。そして、3節で提案法の概要について説明する。次に、4節において、提案法において利用する水のシミュレーション方法とデータベース作成方法について説明した後、5節においてアニメーションの編集方法について説明する。提案法を用いた実験例について6節で示した後、7節で本稿のまとめを行う。

2. 関連研究

流体解析を利用して水の動きをシミュレーションする手法は数多く提案されている[1][2][3]。流体シミュレーションの手法は粒子法と格子法の2種類に分類される。粒子法は、流体を多数の自由に動き回る粒子の集合とみなし、それらを用いて流体の方程式を解き、それにより得られた結果に従って粒子を移動させることで流体としての挙動を実現する[1][2]。格子法は、計算点を格子上の点とし各格子点に対して物理量を割り付け、その値を用いて流体方程式を解き、その結果に従って各格子点の物理量を移流させることで流体としての挙動を実現する[3]。これらの流体解析手法の確立により、リアルな挙動の水のアニメーションを作成することが可能となった。しかし、これらのシミュレーションにおいて、所望の水の動きを得るためにはシミュレーションに影響する多数のパラメータを試行錯誤的に設定しなければならず、極めて煩雑な作業が必要となる。このことから、シミュレーションにより得られる結果の編集性の向上が必要である。この問題を解決するための方法の一つとして、水の動きを制御する手法が提案されている。

Shiらはフィードバック制御の考え方を利用した手法を提案している[4]。また、Thureyらは、詳細形状を保持しつつ、水の動きを制御する手法を提案している[5]。しかしながら、これらの制御手法では、水の形状を整えるために実際には存在しない仮想的な外力を強制的に付与している。そのため、水が人や動物、物体などの形状になるような実際には有り得ない映像を演出するために利用する場合は非常に効果的であ

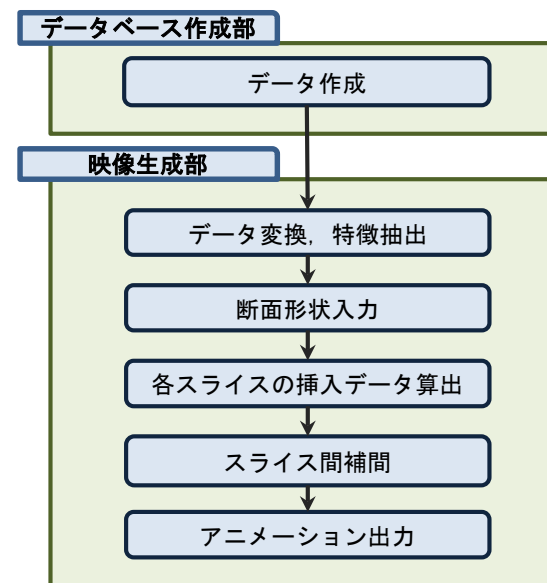


図 1. 提案手法の処理の流れ

るが、津波や砕波などのような自然な流れの制御に利用することは困難である。

本研究では、これらの問題を解決するため、シミュレーション結果のデータベースを利用して所望の波の演出を実現する手法を提案する。

3. 提案手法の概要

提案手法の処理の流れを図1に示す。提案手法は、データベース作成部と映像生成部とに分かれている。まず、前処理として、データベース作成部において、初期条件およびパラメータを様々に変更して流体シミュレーションを実行して、砕波のシミュレーションデータを複数生成し、シミュレーションのデータベースを構築する。本手法では、粒子法の一つであるMPS (Moving Particle Semi-Implicit) 法(4節参照)を用い、二次元のシミュレーションによりデータベースを構築する。次に、映像生成部では、まず、編集作業を容易にするため、これらのシミュレーションから得られる粒子データを格子データに変換する。これは、粒子データでは計算対象となる粒子の数が非常に多く、計算コストが増大してしまうだけでなく、異なるシミュレーションデータ間の対応関係を取ることが困難であるためである。その後、編集処理に使用する各

シミュレーションデータの特徴量を抽出する。本稿では、特徴量として水の高さと各シミュレーションデータ間の類似度を算出する。アニメーションの編集作業は、水の進行方向に対して垂直な断面での波の断面形状（水面の高さのライン）をユーザが入力することで、その形状を満たすシミュレーションデータをデータベースから検索して配置する。最後に、各シミュレーションデータ間を補間することで、二次元のアニメーションを三次元のアニメーションとして表示する。

4. データベース作成部

本節では、データベース作成部での処理について述べる。まず、本手法で利用した水のシミュレーション方法について説明する。次に、その水のシミュレーション方法を用いたデータベースの作成方法について説明する。

4.1 MPS (Moving Particle Semi-implicit) 法

MPS 法は、水に代表される非圧縮性の流体を多数の粒子の集合とみなし、その粒子を用いて流体の方程式を解くことで、流体としての動きを計算する手法である。非圧縮性流れの支配方程式を以下に示す。

$$\frac{Du}{Dt} = -\frac{1}{\rho} \nabla P + \nu \nabla^2 u + g \quad (1)$$

$$\frac{D\rho}{Dt} = 0 \quad (2)$$

ここで、 ρ は密度、 u は速度、 P は圧力、 ν は粘性係数、 g は重力加速度である。式 (1) は運動量保存則を表しており、ナビエ-ストークス方程式と呼ばれる方程式である。右辺は、それぞれ、第一項が圧力勾配項、第二項が粘性項、第三項が重力項と呼ばれる。式 (2) は質量保存則の式である。MPS 法において、勾配や発散、ラプラシアンなどの微分演算子は、それぞれに対応する粒子間相互作用モデルを定義し、各モデルを微分方程式中の微分演算子に適用することで離散化を行う。また、非圧縮性流れの支配方程式に含まれる双曲線型（対流項）、放物型（粘性項）、楕円型（圧力ポアソン方程式）の三種類の偏微分方程式に対応するために半陰的アルゴリズムにより上記方程式を解く [6]。

4.2 データベース作成

前処理において、以下の方法によりデータベースを作成する。ここで、出力するデータはファイルサイズなども考慮し、粒子座標とシミュレーション時間など必要最低限に留める。また、粒子座標をそのまま出力すると、次節で述べる格子データ変換時の格子幅や格子数を自由に決定することが可能となる利点がある。

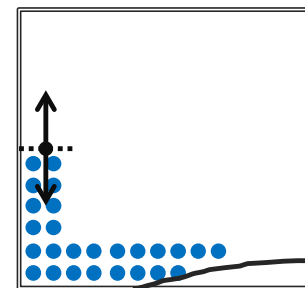


図 2. シミュレーション初期状態

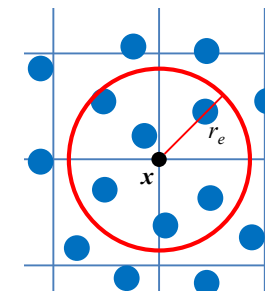


図 3. 影響半径

砕波データ作成のための初期条件として、図 2 に示すような初期条件を使用する。この図 2 の左側に示す矢印部分のように粒子を積み上げる高さを変化させることで、シミュレーション中の粒子数を変化させ、様々な砕波のシミュレーションデータを作成する。以降、ここで作成した N 個のシミュレーションデータのうち n 番目のシミュレーションデータを $Data_n$ と表記する。

5. 映像生成部

映像生成部の処理の流れは図 1 で示したとおりである。はじめに、データベース作成部により作成した複数のシミュレーションデータから後の編集処理に使用する特徴量を抽出する。その後、ユーザはインターフェースを用いて波の進行方向に対して垂直な断面での波の断面形状を入力・編集する。そして、その編集結果をもとに、三次元のアニメーションとして出力する。本節では、それぞれの処理について詳細に述べる。

5.1 特徴抽出

映像生成部では、まず、前処理として作成したシミュレーションデータの特徴量を抽出する。本研究では、以下の二つの特徴量を抽出する。一つ目は各シミュレーションデータにおける任意の位置での波が到達する最大高さ、二つ目はシミュレーションデータの編集処理において各データ間の違いを表現する指標となる各データ間の類似度である。これらの特徴量の計算は、粒子データを格子データに変換して行う。変換後の格子サイズおよび格子幅はもとのシミュレーションデータの粒子数にかかわらずすべて同じ値とする。これにより、粒子数が異なるデータ間でも容易に対応関係をとることが可能となる。格子データへの変換は次式により行う。

$$V(x) = \sum_i W(|x - r_i|) \quad (3)$$

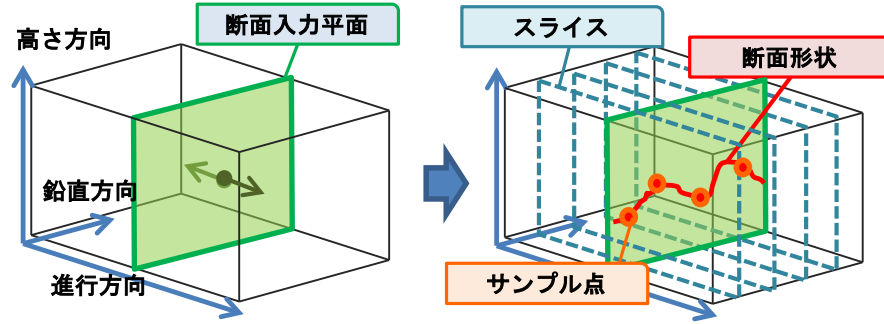


図 4. 断面入力位置の指定 (左) と断面形状の入力 (右)

$$W(r) = \begin{cases} \frac{r_e}{r} - 1 & (0 < r \leq r_e) \\ 0 & (r_e < r) \end{cases}$$

ここで、 $V(\mathbf{x})$ は変換後の各格子点の水の密度、 $\mathbf{x}$ は格子点の座標、 r は粒子座標、 r_e は粒子が格子点に対して影響を及ぼす半径である (図 3 参照)。

式 (3) を用いて作成した格子データを用いて前述の二つの特徴量を算出する。一つ目の波の高さについては、波の進行方向を軸としたとき、その軸上の任意の位置において水が到達した最大高さを記憶する。 $Data_n$ に対する位置 a における波の高さは以下の式により算出する。

$$H_n(a) = \max_t \{h_n(a, t)\} \quad (4)$$

ここで、 t は時間、 $h_n(a, t)$ は $Data_n$ の時刻 t での位置 a における波の高さである。

二つ目のシミュレーションデータ間の類似度については、すべての時刻におけるすべての格子点の密度差の二乗和を正規化したものとし、次式により算出する。

$$D_{i,j} = \frac{D'_{i,j}}{\max_j (D'_{i,j})} \quad (5)$$

$$D'_{i,j} = \sum_x \{V_i(\mathbf{x}) - V_j(\mathbf{x})\}^2$$

ここで、 $D_{i,j}$ は $Data_i$ から見た $Data_j$ の類似度を表している。

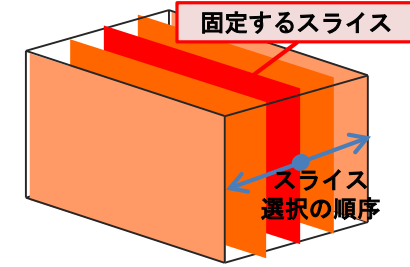


図 5. スライスに配置するデータの選択

以降の編集処理では、本節で算出した二種類のデータ H_n 、 $D_{i,j}$ を用いて波のアニメーションを編集する。

5.2 断面形状指定による編集

本処理では、図 4 に示すように、ユーザにより波の進行方向に対して垂直な断面上において目的の波の形状を指定する。まず、入力を行う断面の位置を指定したのち、その断面上での波の形状を入力する。次に、システムは、指定した波の断面形状を一定間隔にサンプルし、そのサンプル点での高さに近い値を取るシミュレーションデータを各サンプル点の位置に配置する (図 4 参照)。以降、このシミュレーションデータを配置する位置をスライスと呼ぶ。すなわち、各スライスには次式を最大にするシミュレーションデータ $Data_n$ を配置する。

$$S_h(n, i) = 1 - \left[\frac{|H_n(s) - T(i)|}{\max_m \{H_m(s) - T(i)\}} \right]^2 \quad (6)$$

ここで、 s は入力断面の位置、 T は目標とする断面形状の高さ、 i はスライス番号である。式 (6) により、各スライスにおいて最も近い高さとなるシミュレーションデータが選択され、指定した形状通りの波を作成することができる。しかし、式 (6) のみを用いた場合、隣接するスライス間で類似度が大きく違うデータが選択される可能性があり、スライス間の不連続さが目立つアニメーションが作成されてしまうことがある。その問題に対処する方法として、式 (6) に加えて、隣接スライスとの類似度を考慮する項を評価式に追加する。このとき、各スライスに対して配置する最適なシミュレーションデータは、次式を最大化するシミュレーションデータの組み合わせとなる。

$$S = \sum_i \{\alpha S_h(d(i), i) + (1 - \alpha) S_d(d(i), i)\} \quad (7)$$

$$S_d(n, i) = \frac{1}{2} (1 - D_{n, d(i-1)}^2 + 1 - D_{n, d(i+1)}^2)$$

ここで、 i はスライス番号、 $d(i)$ はスライス番号 i において選択されたデータ番号、 α は $0 \leq \alpha \leq 1$ を満たす係数である。

式 (7) において、右辺第二項が新たに追加した隣接するスライスとの類似度を考慮する項であり、 α の値を調節することで形状の近さと隣接スライスとの類似性のどちらを優先するかを設定できる。しかし、式 (7) を最大化するためには、すべてのシミュレーションデータの組み合わせを探索するという最適化問題を解かねばならず、計算コストが非常に高い。そこで、本手法では以下のような手順により各スライスに配置するデータを決定する。

まず、全スライス中でデータを固定するスライスを 1 つ選択する。そのスライスは、ユーザ操作により決定し、そのスライスにおいては式 (6) を最大にするデータを配置する。次に、選択したスライスを中心として隣接スライス間の類似性を考慮した評価式を用いて各スライスに配置するデータを決定する (図 5 参照)。また、このときの評価には次式を用いる。

$$S'(n, i) = \alpha S_h(n, i) + (1 - \alpha) (1 - D_{n, d(j')}^2) \quad (8)$$

$$j' = \begin{cases} i+1 & (i < k) \\ i-1 & (i > k) \end{cases}$$

ここで、 k は固定したスライスの番号である。式 (7) が両隣のスライスとの類似性を見ていたのに対し、式 (8) は隣接するスライスのうち固定したスライス側のスライスの類似性のみを考慮している。

上記処理により、各スライスに配置するシミュレーションデータを決定した後、各スライス間を補間することで三次元のデータを作成する。補間方法は、各スライスにおけるデータの距離場を線形補間により補間する。ここで、各スライスのデータの距離場とは、式 (3) により変換したデータの各格子点について、水面からの距離を格納したものである。ただし、水の領域は正の距離、それ以外の領域は負の距離を割り当てる (図 6 参照)。このように定義した各スライスのデータの距離場を補間することにより、滑らかに水面を接続することが可能となる。

6. 実験結果

提案手法による実験結果を図 7 に示す。実験環境は、CPU: Core 2 Quad 2.66GHz 2.67GHz, GPU: GeForce GTS250, メモリ: 4.00GB, グラフィックス API: OpenGL であ

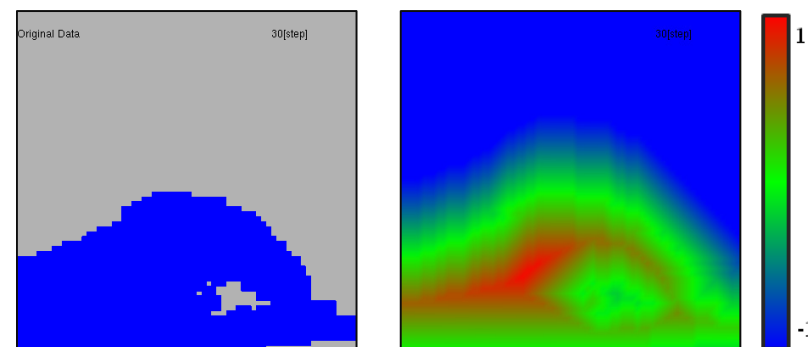


図 6. 水の状態 (左) と距離場 (右)

り、15 個のシミュレーションデータを用意し、各データのステップ数は 120 ステップ (約 2 秒間) である。また、図 7 の左側の図における二種類の線はそれぞれ、赤線がユーザの指定した断面形状、青線が実際に選択された各スライスのデータがとる高さを示しており、緑色の平面は断面形状を編集する平面の位置を示している。水に関しては、左側の図においては距離場の正の格子点を点として表示しており、右側の図は左側の図の距離場をマーチングキューブ法によりポリゴンデータに変換し POV-Ray を用いてレンダリングしたものである。各データのスライス数は 10 個であり、全データの入力形状 (図 7 の赤い線) は同じ形状を指定している。さらに、スライス選択において固定としたスライスは左から 2 枚目のスライスである。すなわち、式 (8) を用いたスライス選択処理は、そのスライスを中心として実行されている。

図 7 上段は $\alpha=1.0$ 、つまり、入力形状に最も近いデータを各スライスで選択したときの編集結果である。目標である入力形状を表す赤線と選択されたデータを表す青線がほぼ一致しており、希望通りの形状を作成できていることが確認できる。しかし、各データで波の進行速度が異なっているため、位置によっては波が到達していない部分があり、同時刻に入力した形状を達成できているわけではない。そのため、各データ間での不連続さが目立ってしまっている。

図 7 中段は $\alpha=0.7$ を指定し、入力形状との誤差に加え隣接するスライスとの類似性を僅かに考慮した結果である。 $\alpha=1.0$ の時と比較して、隣接するスライス同士の類似性が高くなるデータが選択されていることが確認できる。これにより、 $\alpha=1.0$ のときに見られた水の到達していない部分が減少しており、各スライス間の不連続さは多少目立ちにくくなっている。

図 7 下段は $\alpha=0.4$ の時の結果であり、 $\alpha=0.7$ の時よりもさらに隣接するスライスとの類似性を重視した場合の結果である。この場合、類似性は高まるが、入力した断面形

状とは異なる結果となってしまう。このような問題は、設定したサンプル数にも影響してくるため、サンプル点数を変更するなど異なった条件に対しても実験を行い検証していく必要がある。

7. おわりに

本論文では、複数のシミュレーション結果を組み合わせ、水のアニメーションを編集する手法を提案した。提案手法では、ユーザが波の断面形状を入力することで波の形状を調整するため、直感的な編集が可能である。また、目的の形状を満たし、かつ、アニメーションでの自然さを目指すための方法として、スライスに配置するデータ選択の評価式を形状の近さを表す項と隣接するスライスとの類似性を表す項の二つの項を用いて定義し、係数においてその比重を選択可能にした。そのため、ユーザに負担をかけることない編集が可能である。

今後の課題としては、まず、各スライス間の補間方法の改善が挙げられる。これは、本手法で用いた水面からの距離を表した距離場同士の補間では、詳細形状を補間することができないという問題があるためであり、これが各スライス間での不連続さが目立つ原因の一つとなっている。また、各シミュレーションデータにおける波の進行速度が違ってもシミュレーションデータ間の不連続さが目立つ要因にもなっており、各データの波の速度を調節するなどの改善が考えられる。

謝辞 本研究は独立行政法人科学技術振興機構（JST）CREST によりサポートされています。

参考文献

- 1) Koshizuka S., Oka Y.: Moving particle semi-implicit method for fragmentation of incompressible fluid, *Nuclear Science and Engineering*, 123, pp.421-434(1996)
- 2) Monaghan J. J.: An introduction to SPH, *Computer Physics Communications*, Volume 48, Issue 1, pp.89-96(1988)
- 3) N Foster., R Fedkiw.: Practical Animation of Liquids, *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pp.23-30(2001)
- 4) L Shi., Y Yu.: Taming Liquids for Rapidly Changing Targets, *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp.229-236(2005)
- 5) N Thurey., R Keiser., M Pauly., U Rude.: Detail-Preserving Fluid Control, *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp.7-12(2006)
- 6) 越塚誠一: 粒子法, 丸善株式会社(2005)

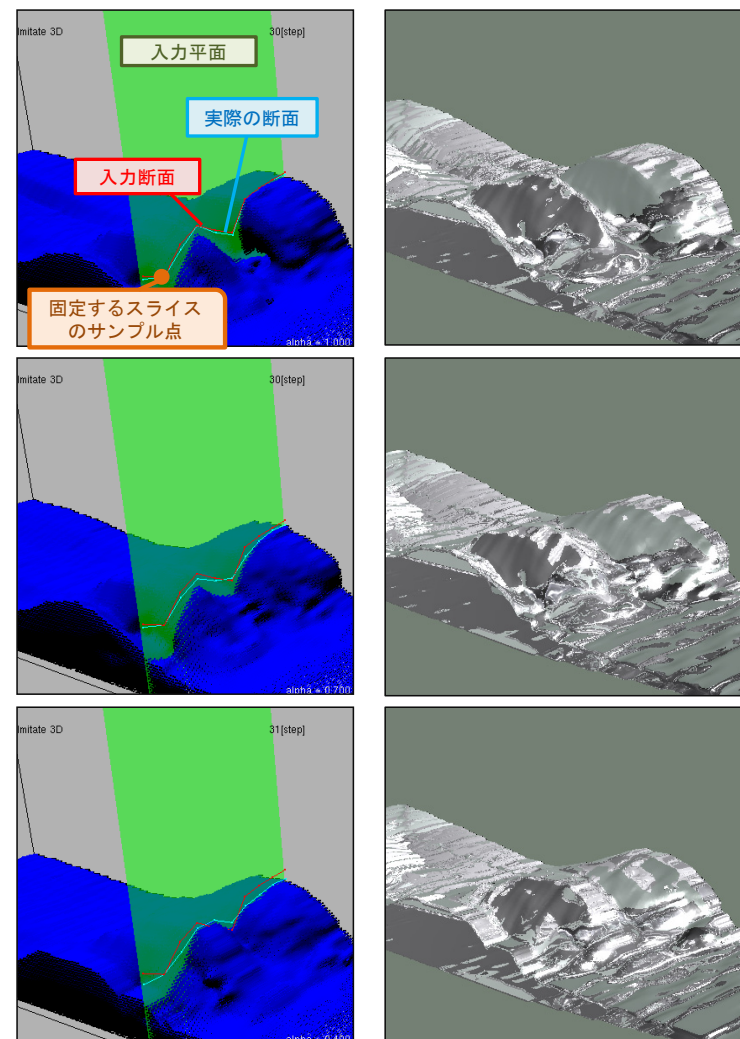


図 7. 実験結果（上段: $\alpha=1.0$, 中段: $\alpha=0.7$, 下段: $\alpha=0.4$ ）