

物理シミュレーションにおける速度の履歴を用いた 液体の音データの検索手法

齊藤 彪雅^{1,a)} 佐藤 周平^{1,b)} 土橋 宜典²

概要: 本稿では流体アニメーションにおける効果音の合成手法を提案する。従来のアプローチでは物理シミュレーションによって流体の音を合成するが、このアプローチでは膨大な計算コストを必要とする。そこで我々は、流体の効果音を合成するためのデータ駆動型の手法を提案する。本稿では液体の音に着目する。まず事前に、録音した音源を含む液体の音のデータベースを用意し、入力した液体の動きに最も合う効果音をデータベースの中から自動で検索する。この検索は、効果音の波形とシミュレーションによって計算された液体の速度の履歴を比較することによって実現する。速度履歴は、液体の音が発生すると予測される領域について計算される。そして、速度履歴とデータベース内の各音データの波形との間の距離を計算し、最も距離が小さくなるデータを選択する。我々の手法は、1度データベースが用意されれば、シミュレーションされた液体の動きに対し高速に効果音を合成できる。本稿では初期実験として、少数の音データを含んだ小さいデータベースを用い、我々の検索手法の有効性を評価する。

キーワード: 流体シミュレーション, 流体音の合成, データベース検索, Dynamic Time Warping

1. はじめに

近年コンピュータグラフィックス (CG) 技術が、映画やゲームにおいて用いられることが一般的となっている。特に水や煙、炎といった流体では、物理シミュレーションによりリアルなアニメーションを作成できる。しかし、流体シミュレーションはリアルさを追求するにつれて、膨大な計算コストを必要としてしまう。そのため、シミュレーションの計算を高速化するための様々な手法が提案されている。

流体音もリアルな映像を作成する上で重要な要素の一つである。流体の効果音は様々な方法で作られており、録音した本物の音を使用する場合や、物理法則に則り効果音を生成する場合などが挙げられる。録音した実際の音を使用する場合、当然その音質はリアルであるが、映像と同様のシーンを実世界で再現するのは困難な場合が多い。一方、物理ベースのシミュレーションに基づいた方法は映像に合った効果音の生成が可能であり、生成される音もリアルである [1, 2]。この方法では非常に高精度な液体シミュレ

ーションを使用した際に再現可能である水の泡を利用し、この泡によって生じる液体の表面の振動を計算することで、効果音を生成する手法となっている。しかし、高精度のシミュレーションおよび効果音の計算コストは非常に大きく、計算に一日かかってしまう場合もある。

そこで、より高速な流体音の合成を実現するため、我々は水の効果音を集めたデータベースを使用し、与えられた液体シミュレーションの挙動に合った効果音をデータベースの中から検索し合成する手法を提案する。我々の手法ではシミュレーションで記録した速度の履歴と水の効果音の波形とを比較する。今回の手法では、水の音は気泡から生じるという事実と泡の発生量と液体の速度の間に相関関係があるという仮定を基に速度の履歴を利用する。速度の履歴は音が生じると予想される領域において計測する。この領域は水面の高さから得られるハイトフィールドを用いて決定する。その後、速度の履歴とデータベース内の各効果音の波形との距離を計算し、この距離が最も小さくなる効果音をデータベースから選択する。この距離は Dynamic Time Warping (DTW) を用いて求める。

本稿では、少数の効果音データを含む小規模のデータベースを用い、水の動きにあった効果音をデータベースから選べるかどうかを検証する。我々の手法は、データベースを一度用意さえすれば高速に効果音の合成が可能であ

¹ 富山大学
University of Toyama

² 北海道大学
Hokkaido University

a) m2171121@ems.u-toyama.ac.jp

b) ssato@eng.u-toyama.ac.jp

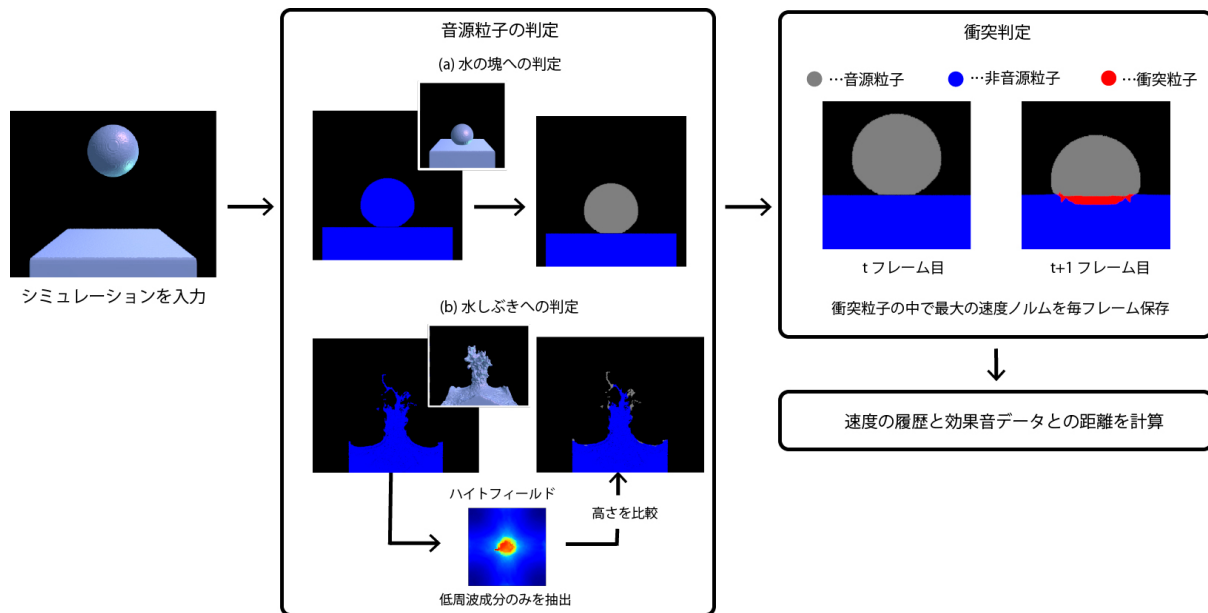


図 1: 本手法の概要図

る。本稿では、効果音の検索のみを行い、効果音を正確に映像に合わせる部分については手動で行っている。

2. 関連研究

2.1 シミュレーションによる流体音の生成

James らのグループは物理ベースで流体シミュレーションに水の効果音を合成するための方法を提案した [1]。この方法ではシミュレーションにおいて生成される水泡の表面の振動を音源とすることで音を推定した。この研究では、従来の研究にはなかったシミュレーションと音の生成の並行処理を可能とした。その結果、より高速な計算かつリアルな効果音の生成を実現している。しかしこの研究は音の生成の際に単相流のソルバを用いており、短時間で近似解を求めることは可能ではあるが、リアルな効果音の作成には限界がある。この問題を解決すべく James らのグループはのちに新たな手法 [2] を発表した。この研究では、従来の方法よりも精度の高い二相流の流体ソルバにより生成される詳細な気泡の動きに基づいて、よりリアルな流体音を生成することを可能とした。しかし、この方法は非常に多くの計算時間を要し、また、映像制作にはあまり利用されない精度の高いソルバを用いており、実際に利用するコストは大きい。本研究では、そのような精度の高いソルバを必要とせず、また既にあるデータを再利用することで安価にシミュレーション結果に流体音を合成することを目的としている。

2.2 Dynamic Time Warping (DTW)

DTW は 2 つの時系列データの各点の距離を総当たりで計算し、その距離が小さくなるように点を結ぶアルゴリズムである [3]。DTW は総当たりでの計算でかつ、点の重複

を許すアルゴリズムであるため、2 つの時系列データの周期が異なっても定義可能であり、単にデータの形状の類似度を測る際に有用なアルゴリズムであると言える。またアルゴリズム自体も非常に実装が容易であるという利点を持ち合わせている。本手法ではこの DTW を用いて速度データと効果音との類似度を計算する。

3. 提案手法

本手法では、格子と粒子の両方を用いる FLIP 法 (Fluid-Implicit-Particle) によってシミュレーションを実行し、データベースの中からシミュレーションに合う効果音を検索し合成する。この検索は液体の動きから得られる速度の履歴を基に行う。図 1 に本手法の概要図を示す。以降で各処理の詳細について説明する。

3.1 音源粒子の判定

まず初めに、液体部分の粒子を音の発生に関わる粒子とそうでない粒子の二通りに分類する。以降ではこれらをそれぞれ音源粒子と非音源粒子と呼ぶ。今回の実験における音源粒子は水しぶきや落下してくる水の塊などを想定している。音源粒子は水面の高さから得られるハイトフィールドに基づいて決定される。ハイトフィールドは水平面における 2 次元の格子点上に定義され、各格子点には近傍の粒子の中で最も鉛直方向の座標値が大きい粒子の高さが格納される。ハイトフィールドは図 2 に示す。図中の (a) はシミュレーションのある一フレーム、(b) はそのフレームは真上から見た時の様子、(c) はそのフレームにおいて生成されるハイトフィールドとなっており、(c) では青い箇所ほど低く、赤い箇所ほど高いことを示している。このハイトフィールドにフーリエ変換を適用し、低周波成分

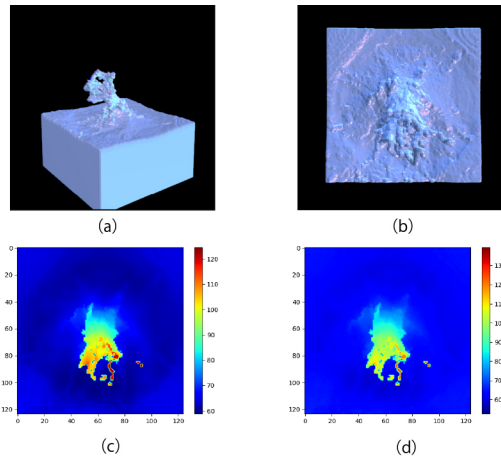


図 2: シミュレーション中の一フレームとその瞬間のハイトフィールド

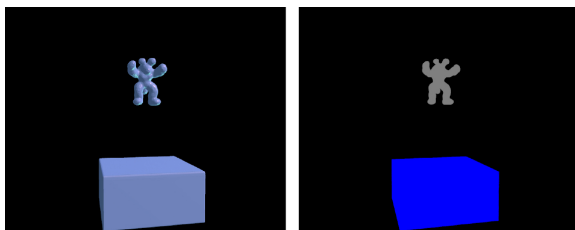


図 3: 水の塊に対する判定の様子

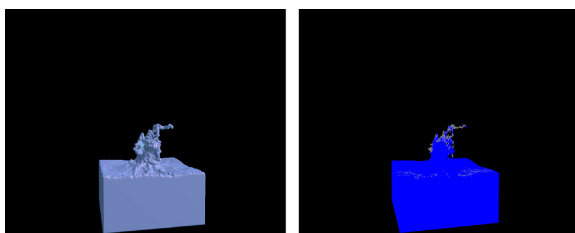


図 4: 水しぶきに対する判定の様子

のみを抽出することでフィールドを平滑化する。平滑化後のハイトフィールドの一例が図 2 の (d) である。抽出する低周波成分は実験的に決定する。平滑化後のハイトフィールドの高さに対応する近傍の粒子の高さを比較し、ハイトフィールドよりも高い位置に存在する粒子を音源粒子として判定する。加えて、シミュレーションの途中で追加される水滴や水の塊も音源粒子として設定する。図 3, 図 4 は実際に粒子に判定を与えた際の様子を示している。図 3 は水の塊に対する判定, 図 4 は液体同士の衝突によって生じた水しぶきの部分に対する判定を示している。図中の青色の箇所が非音源粒子, 灰色の箇所が音源粒子として判定されている。

3.2 水面への衝突の判定

次に音源粒子の衝突の検出方法について述べる。以降では水面に衝突したと判定した音源粒子を衝突粒子と呼ぶ。衝突粒子を検出するに当たり, 各格子点に対して, 近傍の

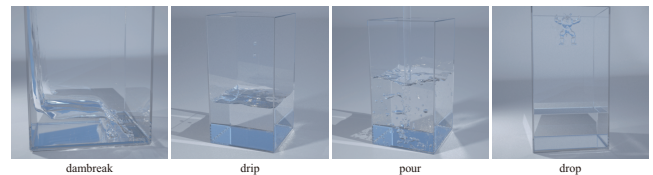


図 5: データベースに用いた各流体音データの生成に利用されたシミュレーションの結果の一部 (出典: 文献 [2])

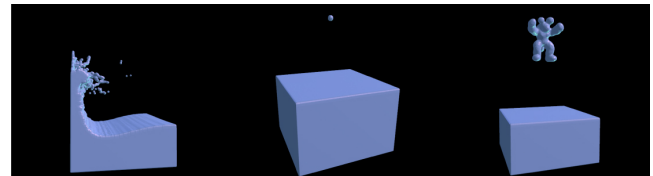


図 6: FLIP 法による各シミュレーションの一部

表 1: 各シミュレーションの速度データと流体音データ間の類似度

	dambreak	drip	drop	pour
(a) damBreak	0.7535	2.3572	1.5441	2.9895
(b) drip	13.8423	6.2993	16.0298	21.1976
(c) drop	2.4614	3.8745	0.9624	4.775

粒子が音源粒子なら音源領域, 非音源粒子なら非音源領域という情報を格納した場を新たに用意する。音源粒子を現在の速度に従って移動させ, 非音源領域に侵入した場合に水面に衝突したと定義し, 音源粒子を衝突粒子と判定する。ここで衝突が判定されなかった音源粒子は, 次のフレームにおいても音源粒子のままとする。全ての音源粒子に対し衝突検出を行った後, 衝突粒子の中で最も速度のノルムが大きい値を速度の履歴として記録する。シミュレーションが終了するまでフレーム数分の速度の履歴を効果音の検索に用いる。

3.3 速度の履歴と効果音の波形との類似度計算

シミュレーションから算出した速度の履歴と水の効果音の波形に対し DTW を適用することで類似度を求める。なお, 効果音の波形については正数部分のみを切り出したデータを使用する。そして, 最も距離が小さくなる効果音を選択し, シミュレーションの映像に合成する。最後に, 効果音が映像に正確に合うように手動で調整する。

4. 実験結果

提案法による結果を示す。今回は Langlois らによる物理ベースの方法 [2] で生成された 4 種類の効果音をデータベースに含めた。ここでそれぞれのデータを "dambreak", "drip", "drop", "pour" (図 5 に示す) と呼ぶこととする。図 6 は MantaFlow [4] を使用して作成した液体のシミュレーション結果であり, これらに効果音を

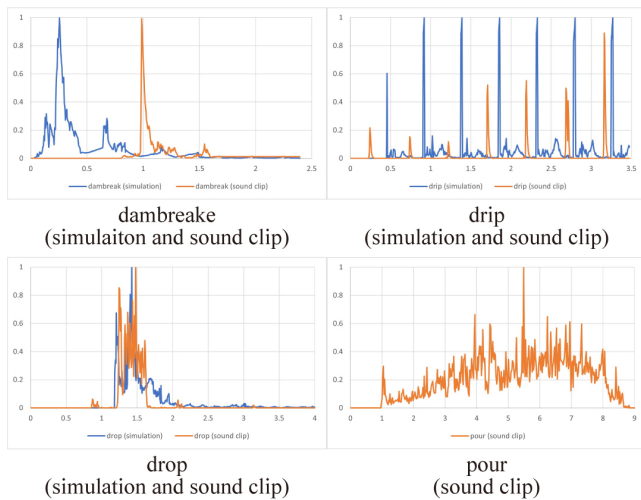


図 7: 速度の履歴のグラフと効果音の波形

合成する. 今回は検索手法の有効性を評価するため, 図 6 の (a), (b), (c) は, それぞれ図 5 の dambreak, drip, drop に類似したシーンを作成した. 図 7 は速度の履歴のグラフ (青) とデータベースの各効果音の波形 (橙) を示している. なおグラフの縦軸は振幅 (橙) あるいはノルムの大きさ (青) を示しており, いずれも正規化した値となっている. 表 1 は DTW による計算結果を示しており, 表中の値が小さいほど類似度が高い.

図 7 が示すように, 速度の履歴とデータベースから選ばれた効果音の波形の形状が対応したシーン間において類似していることが分かる. また, 表 1 からそれぞれの DTW により計算された距離も最短となっているのがわかる. DTW に要する時間は約 3 秒であった.

しかしながら, フーリエ変換後のハイフィールドで抽出する低周波成分をシミュレーション毎で調整しており, "drip", "drop" で使用した抽出範囲と "dambreak" で使用した抽出範囲が大きく異なっている. "dambreak" に "drop" と "drip" の抽出範囲を用いた場合, "pour" の音声 が最も類似するという結果になった. "drop" や "drip" ではシミュレーション中に生じる小さい波の部分の音源領域化を防ぐため, 広範囲の周波数成分を抽出したが, "dambreak" の場合, 音源領域として判定されない水しぶきの部分が増え, ノルム値が大きい粒子を音源粒子として判定できなかったことが原因と考えられる.

5. まとめと今後の課題

本稿では, 事前に用意された液体の効果音のデータベースを用いて, 液体アニメーションに効果音を高速に合成するための手法を提案した. 我々の手法は液体の効果音を集めたデータベースの中から, 与えられた液体アニメーションに最も合う効果音を, シミュレーションから得られる速度の履歴と効果音データの波形を比較することで検索した. 比較は DTW によって計算した. 我々の手法はデータ

ベースが用意されれば, 液体の動きに合った効果音を高速に合成することが出来る.

本手法では音データの検索のみを行っており, 映像と音を正確に合わせるためには, 検索した効果音を手動で調整しなければならない. そのため, 今後自動での調整方法を考案することを予定している. また, 今回の実験ではシミュレーションにより作成された少規模のデータベースを用いて検証を行ったが, 録音された音源を利用した検証や, 大規模なデータベースを構築した場合の本手法の有効性の検証を今後行う.

謝辞 本研究は JSPS 科研費 JP20K19945 の助成を受けたものです.

参考文献

- [1] Changxi Zheng, and Doug L. James: *Harmonic fluids*, ACM Trans. Graph., Vol. 28 (3), Article No. 37 (2009).
- [2] Timothy R. Langlois, Changxi Zheng, and Doug L. James: *Toward Animating Water with Complex Acoustic Bubbles*, ACM Trans. Graph., Vol. 35 (4), Article No. 95 (2016).
- [3] Keogh, E., and Ratanamahatana, C.: *Exact indexing of dynamic time warping*, Knowledge and Information Systems, Vol. 7, pp. 358-386 (2005).
- [4] Nils Thuerey and Tobias Pfaff. 2018. MantaFlow. <http://mantaflow.com>.