Vol.2020-CG-180 No.15
Vol.2020-DCC-26 No.15
Vol.2020-CVIM-223 No.15
2020/11/5

データベースを用いた流体音の合成のための速度場に基づくデータ検索手法

齊藤 彪雅[1,a)]　佐藤 周平[1,b)]　土橋 宜典[2]　高 尚策[1]　唐 政[1]

概要：今日では，映画やゲームといった分野においてコンピュータグラフィクス（CG）が用いられることが一般的になっている．幅広い分野において活用されるようになった CG 技術は映像の生成のみならず，効果音の生成においても活用されている．CG の映像に効果音を合成する主な方法としては，物理ベースで計算した音を利用する場合や，実際に音を録音して使用する場合がある．しかし，物理ベースの方法では大きな計算コストを要し，録音では映像に合うよう調整する作業も含めた労力が必要となる．そこで本研究では，あらかじめ用意した流体音のデータを利用して，流体アニメーションに流体音を合成するための手法の研究を行っている．本稿では，液体シミュレーションの速度場を利用し，流体音のデータベースから最もシミュレーションに適したデータを検索する手法を提案する．本手法では，シミュレーション中の液体同士の衝突を検出し，その際の速度のノルムを毎フレーム記録する．そして，記録した速度のデータと音データの類似度を算出し，最も類似度が高いものをシミュレーションの映像に合成する．ただし本稿では，映像と音の細かなずれは手動で調整する．これにより物理ベースで計算を解くよりも小さい計算コストでシミュレーションに流体音を合成することが期待できる．本稿では，格子で定義されるシミュレーションを対象とする．

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．特に煙や水を始めとした流体の CG 映像はリアルな表現の追求のため多くの研究者がより高速な計算や精度の向上のための方法の開発に臨んでいる．

CG 技術が発展した結果，映像のみにとどまらず効果音の生成にも適用されるようになった．効果音の生成技術はサウンドレンダリングと称されている．流体アニメーションにおけるサウンドレンダリングの研究の一例として，James らによって発表された研究 [1] が挙げられる．この研究ではシミュレーションにおける水の気泡によって起こる水面の振動に着眼し，その振動をシンプルな関数に当てはめることで，高速な効果音の生成を実現した．また James らは，更なる精度向上のために改良した研究 [2] も発表した．

しかし，この研究ではリアルな効果音を作成できる一方で，莫大な計算コストを要することが問題となっている．この研究のおかげで，シミュレーションと並行して効果音を生成することが可能となったのだが，依然としてその計算コストは膨大である．そこで本研究では物理ベースの計算によって効果音を生成するのではなく，あらかじめ用意したデータベースの中から最も与えられた液体の流れに適合する効果音を検索し，その映像に合成する方法について提案する．先行研究では水泡の振動から効果音を物理ベースで計算するのに対し，本研究ではシミュレーション内における液体同士の衝突を検出し，その瞬間の速度ノルムを毎フレーム保存する．これは流体音が液体内部の気泡から発生しており，液体の速度と発生する気泡の大きさや量に相関があるという仮定に基づいている．記録した速度ノルムのデータとデータベースとして用意した複数の音データの波形の類似度を Dynamic Time Warping（DTW）を用いて求める．そして，類似度が最も高くなった音データをシミュレーションにより得られた映像に合成する．これにより，先行研究よりも短時間でシミュレーション映像に効果音を合成することが期待できる．本項では実験として，

[1] 富山大学
University of Toyama
[2] 北海道大学
Hokkaido University
[a)] s1770229@ems.u-toyama.ac.jp
[b)] ssato@eng.u-toyama.ac.jp

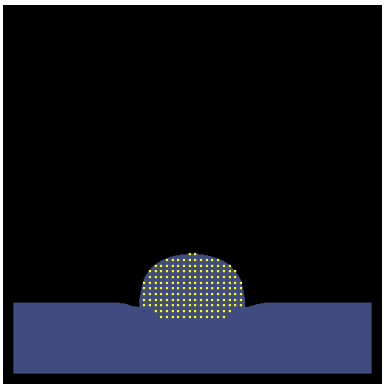
Vol.2020-CG-180 No.15
Vol.2020-DCC-26 No.15
Vol.2020-CVIM-223 No.15
2020/11/5

2 次元の格子ベース流体シミュレーションを対象とする．

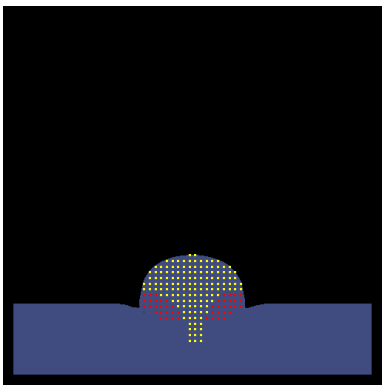
2. 関連研究

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流体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上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研究 [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．



(a) 移流前の様子



(b) 移流後の様子

図 1: 液体同士の衝突判定の様子

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い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．長さが N の時系列データ $S = s_1, ..., s_N$ と長さが M の時系列データ $T = t_1, ..., t_M$ の DTW によって求められる距離 $D(S,T)$ は以下のようになる．

$$D(S,T) = d(N,M),$$

$$d(n,m) = |s_n - t_m| + min \begin{cases} d(n-1,m), \\ d(n,m-1), \\ d(n-1,m-1) \end{cases}$$

$$d(0,0) = d(n,0) = d(0,m) = \infty$$

$$(n = 1, \ldots, N; m = 1, \ldots, M)$$

DTW は上記の式のように単純なアルゴリズムである一方，総当たりで計算を行っているため，用いるデータによっては計算コストが大きくなる可能性がある．本手法ではこの DTW を用いて速度データと流体音データとの類似度を計算する．

3. 提案手法

本手法ではまず，レベルセット法による格子ベースの液体シミュレーションを実行し，そのシミュレーションにおいて速度データを記録する．記録後，データベース内の液体音のデータから波形のデータを抽出し，これらのデータの類似度を計算する．本稿では，DTW を用いて類似度を算出する．以降では，類似度を算出するまでの過程について示す．

3.1 シミュレーションにおける液体同士の衝突判定

まず始めに液体シミュレーションから速度の情報を記録する．本研究では，音の発生源である気泡の発生量と大きさが，液体同士が衝突する際の液体の速度および面積に相関があると仮定する．液面に落下状態の水滴が衝突した瞬間が，本研究において気泡が生成され，その結果流体音が生成されると仮定する状況の一例である．この液体同士の衝突を判定するためにはシミュレーションにおいて液体が，下に溜まっている部分の液体なのか，これから液面に衝突しようと落下している状態の液体なのかを毎フレーム定義する必要がある．本稿では経験的に以下の 2 つの条件

Vol.2020-CG-180 No.15
Vol.2020-DCC-26 No.15
Vol.2020-CVIM-223 No.15
2020/11/5

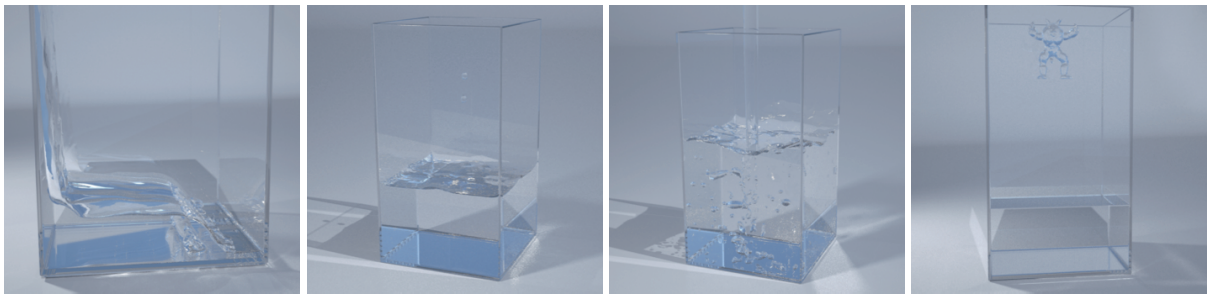


図 2: データベースに用いた各流体音データの生成に利用されたシミュレーションの結果の一部（出典：文献 [2]）

を用いて，液体が落下状態であるかどうかを定義する．

- 格子の速度方向 θ が $(-\frac{6}{12}\pi \leq \theta \leq -\frac{5}{12}\pi)$ であること
- 格子の x 方向の速度 v_x と y 方向の速度 v_y のいずれかの大きさが 1.0 以上であること

1 つ目の条件は格子の持つ速度の向きが下方向かどうかを判別するためであり，2 つ目の条件は動きの小さい部分は音の生成に影響しないと考慮したため設けた．この 2 つの条件による判定は液体の存在する格子点においてのみ実行される．

次に衝突の判定について述べる．先述した 2 つの条件を満たし落下状態と判定された領域を Ω，それ以外の液体領域を Ω' とする．ここで，t フレーム目の速度場を用いて移流後の速度を求める．そして移流後における領域 Ω 上の格子の速度が，領域 Ω であるための条件を満たしているかを再度判定し，満たしていなかった場合に衝突したと判定する．図 1(a) は衝突した瞬間の移流前の様子．図 1(b) は移流後の様子である．2 つの図において，黄色の点部分が領域 Ω を示している．黄色の点が配置されいない青色の部分が領域 Ω' である．また図 1(b) における赤い点部分は衝突が検知された部分を示している．そして，衝突したと判定された全格子点の速度のノルムの総和を毎フレーム記録し，音データの検索に利用する．

3.2 速度データと流体音データの類似度計算

シミュレーションから算出したデータをグラフ化したものと流体音の波形に対し DTW を適用することで類似度を求める．なお，流体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．また前処理として 2 つのデータには正規化と平滑化を行う．平滑化については DTW は外れ値のようなノイズに弱いという特性から有効な処理であると考えたため行なった．平滑化の際には，移動平均を用いる．移動平均の式を以下に示す．

$$\overline{x}_i = \frac{x_{i-k/2}+,\ldots,+x_i+,\ldots,+x_{i+k/2}}{k+1} \qquad (1)$$

ここで k は項数を示している．今回の実験では k の値をシミュレーションのデータでは 20，音声データでは 40 に設定した．移動平均は k の大きさを大きくするにつれてより滑らかになるのだが，値が大きすぎた場合にグラフの形状が損なわれる恐れがあるため，今回はデータの種類に応じて大きさを調整した．

図 3: DamBreak のシミュレーションの一部

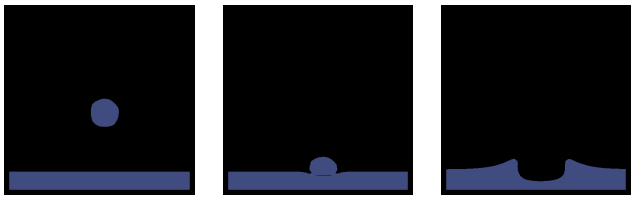


図 4: Drip のシミュレーションの一部

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	pour	drop
DamBreak	0.1781	20.1267	13.2326	2.3912
Drip	48.0300	7.1147	108.3955	20.6080

最後に上記の前処理を施した速度データと各流体音データとに DTW を適用し，最も結果の値の小さかったデータを合成に用いる．ただし，現状本手法はデータベースの中からシミュレーション結果に最も近いデータを検索するのみであるため，シミュレーションと音にはずれが生じている場合もある．その場合は手動で流体音データを調整する必要がある．今後これを自動で調整する方法を開発する予定である．

4. 実験結果

提案法による結果を示す．今回はデータベースとして，物理ベースの方法 [2] により作成された音データを 4 つ（”dambreak”，”drip”，”drop”，”pour”）用意した．”dambreak”は水が波打つ様子の音，”drip”は一定間隔

で水滴を水面に垂らす様子の音，”pour”は水面に水を注ぎ続ける様子の音，”drop”は水面に向かって一つの水の塊が衝突する様子の音を表している．いずれのデータも wav 形式のデータである．また，それぞれの音データを生成する際に利用されたシミュレーションの結果の一部を図 2 に示す．流体音を合成する水のシミュレーションとしては，水面が揺れて波打つシーン（DamBreak）と静止した水面に水滴を垂らすシーン（Drip）の 2 種類を用意した．それぞれのシミュレーションの一部を図 3 と図 4 に示す．表 1 に DamBreak と Drip のシミュレーションの速度データと音データの類似度の計算結果を示す．表の値は，小さいほど類似度が高い．

表の 1 行目は前述のデータベース内に存在する各音データの名称である．図 5 と図 6 は速度データと音データをそれぞれグラフ化したものである．両方の図において，青色のグラフは音データ，橙色のグラフは速度データを示す．またこれらのグラフにおいて，横軸は時間，縦軸は速度データでは速度ノルムの総和，音データでは振幅を示す．

表 1 から DamBreak のシミュレーションでは dambreak，Drip のシミュレーションでは drip の音データが選ばれていることが読み取れる．それぞれグラフにおいても類似した波形となっている．一方，両方のシミュレーションと drop の音データには一部類似した個所が見て取れ，類似度の値としても 2 番目に近くなっている．このようにシミュレーションの状況が異なっていても，類似度としては近くなる場合があり，今後データを増やして，現在のアプローチでどの程度正確に検索が可能かを調査する必要がある．

5. まとめと今後の課題

本稿では，液体シミュレーションにおける速度を利用し，液体音のデータベースの中から最も映像に合う音データを検索し，それを映像に合成するための手法を提案した．サウンドレンダリングのように物理ベースで計算して効果音を生成するのではなく，すでにあるデータから検索するため，データベースさえ準備できれば物理ベースの方法より高速に音の合成が可能である．

今回，速度データと音データの類似度を測るために DTW を用いた．ただし，2.2 節でも記載したように，DTW は計算コストが大きくなる場合があり，今後データを増やした際に問題となる可能性がある．そのため，他の類似度計算方法について調査および比較の実験を行う予定である．

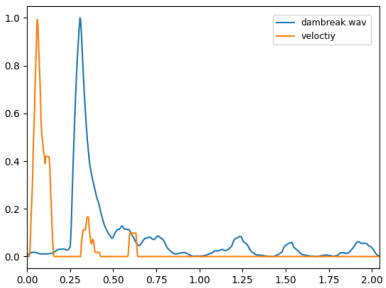
加えて，シミュレーション上における衝突の判定において，図 1(a) から分かるように，落下状態とされた領域が水面の領域に侵入したにも関わらず，衝突したと判定できていない格子が存在しており，より正確な値を記録するため，衝突の判定の定義を改める必要があると考えている．

また本手法では音データの検索のみを行っており，水の映像に音を合成するためには，検索したデータをさらに映像に合わせて調整する必要がある．現状ではこれを手動で行う必要があり，今後自動での調整方法を考案することを予定している．

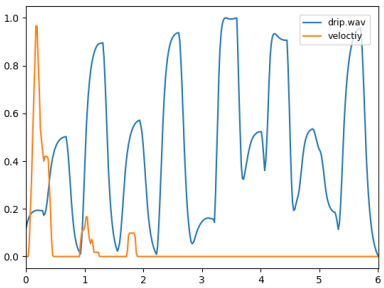
謝辞　本研究は JSPS 科研費 JP20K19945 の助成を受けたものです．

参考文献

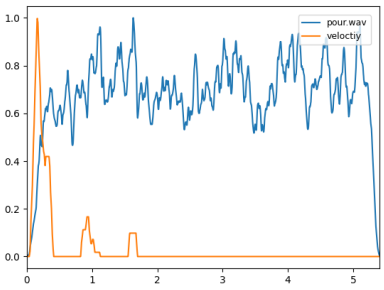
[1] Changxi Zheng, and Doug L. James: *Harmonic fluids*, ACM Trans. Graph., Vol. 28 (3), Article No. 37 (2009).
[2] Timothy R. Langlois, Changxi Zheng, and Doug L. James: *Toward Animating Water with Complex Acoustic Bubbles*, ACM Trans. Graph., Vol. 35 (4), Article No. 95 (2016).
[3] Keogh, E., and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



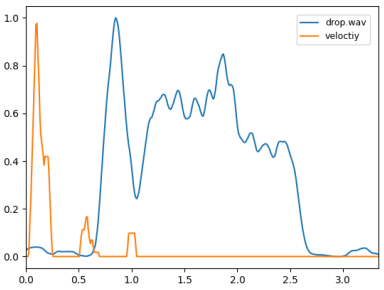
(a) dambreak を用いた比較



(b) drip を用いた比較

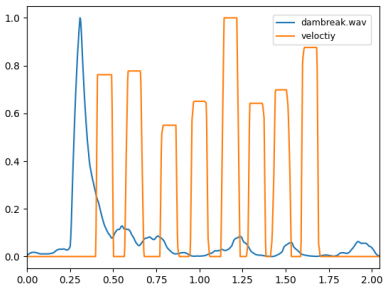


(c) pour を用いた比較

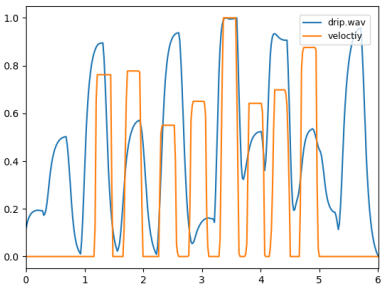


(d) drop を用いた比較

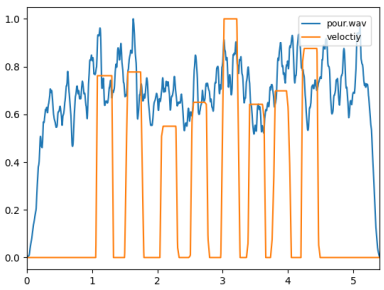
図 5: DamBreak のシミュレーションと音データの比較



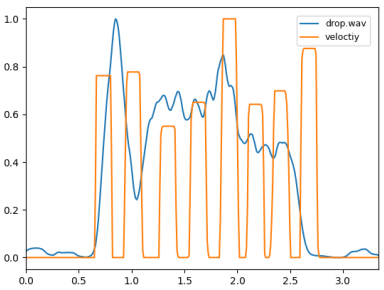
(a) dambreak を用いた比較



(b) drip を用いた比較



(c) pour を用いた比較



(d) drop を用いた比較

図 6: Drip のシミュレーションと音データの比較