

直接操作による3D流体シミュレーションの制御

早川 雄登^{1,a)} 藤代 一成^{1,b)}

概要: 直接操作によって、3D流体シミュレーションを簡易に作成できるインタフェースを提案する。本稿では、流体シミュレーションおよび流体制御に用いるパラメータ調整を特定のモーションデータに対応させた直接操作によって、直感的な流体制御を試みる。Leap Motionを使い、両手のモーションデータを取得、特定のモーションデータをジェスチャとして識別し、3D空間上でジェスチャに応じた流体シミュレーションを実行する。これにより、流体シミュレーションの知識を必要とせず、3D流体シミュレーションを意のままに制御することが可能となる。

3D Fluid Simulation Control by Direct Manipulation

YUTO HAYAKAWA^{1,a)} ISSEI FUJISHIRO^{1,b)}

Abstract: We propose a simple interface for novices to design 3D fluid simulation through direct manipulation. In this article, we try to control fluid intuitively through direct manipulation, where parameters of fluid simulation are specified with hand motions. Recognizing specific hand motions to be captured through the use of Leap Motion as hand gesture commands allows the users to control the dynamics of 3D fluid at will, without detailed knowledge of fluid simulation.

1. 序論

近年、コンピュータグラフィックスによる流体のビジュアルシミュレーションはさまざまな映像作品に用いられている。流体を写實的に表現するだけでなく、映像の内容や演出に合わせて非写實的な挙動をする流体が扱われることも多い。このような流体を実現するためには、流体らしさを維持しながら、期待通りの動きをするようにその挙動を制御しなければならない。よって、流体の写實性とともにも制御性も重要であるといえる。

ところで、流体シミュレーションを解く手法の1つに粒子法がある。粒子法は、流体を粒子の集合と仮定し、それぞれの粒子を質点として物理量をもたせ、支配方程式に則り計算する。粒子ベースの流体シミュレーションにはさまざまなパラメータが存在する。さらに、流体制御には多種類の制御粒子が用いられ、各々の制御粒子を扱うためのパラメータも必要となる。しかし、ユーザに流体シミュレ

ーションの知識がなければ、これらのパラメータの物理的意味や感度を正しく理解して操作することは容易ではない。

そこで本稿では、流体シミュレーションおよび流体制御に用いるパラメータ調整を特定のモーションデータに対応させた直接操作により、直感的な流体挙動の制御を試みる。本研究で開発したシステムは、リアルタイムの3D流体シミュレーション上でハンドジェスチャによる制御が可能である。流体シミュレーションへの深い知識を必要とせずに直感的な操作で、映像作品への流体表現の設計を可能とするシステムを開発することが、本研究の最終目的である。

2. 関連研究

流体の制御に関する関連研究を挙げ、本論文の位置づけを明確にする。

Thureyら[8]は制御粒子を用いる流体の制御手法を提案した。制御粒子は流体の粒子と同様に物理量をもつが、流体の粒子とは独立した挙動を示す。流体の粒子を制御粒子に纏わせながら、制御粒子を任意に動かすことで流体形状を制御する。McNamaraら[3]は入力となるキーフレームを満たすような外力を反復計算で求める手法を提案した。

¹ 慶應義塾大学 大学院理工学研究科
Graduate School of Science and Technology, Keio University
a) yuto.hayakawa@fj.ics.keio.ac.jp
b) fuji@ics.keio.ac.jp

Raveendran ら [5] は、モーフィングで生成したメッシュを境界条件として使用することで流体を変形させている。Shi ら [6] は対象のボーンモデルを動かしたときの流体との形状差と速度差から外力を求めることで流体変形を行った。Stomakhin ら [7] は流束を考慮した境界条件を用い、この境界条件の形状を変えることで、境界面に流れのある流体形状制御を可能とした。本システムでは流体の制御に制御粒子を用いる。制御粒子は計算コストが低く、境界条件とは違い、内的な力であるため直感的に扱うことができる。このため、リアルタイムにモーションデータに対応する本研究と相性がよい。

Zhang ら [11] はボーンモデル、メッシュモデルを制御粒子の集合に置き換えることで、モーションデータによって制御粒子を動かした。また、吉野ら [10] はシェイプマッチング法によって対象形状まで制御粒子を移動させる手法を提案した。これらの研究では対象となる形状があらかじめ入力されており、流体形状変化の流れがすべて決まっている。本システムでは、モーションの動きや軌跡に制御粒子を対応させる。モーションにより流体を制御させ、その結果に応じて、次の流体の制御を指定することができるため、インタラクティブな流体制御が可能である。

スケッチによる流体制御システムに関する研究も存在する。上田ら [9] は、ユーザのスケッチの軌跡に沿ってサーフェイス上に血液が流れるリアルタイム出血シミュレーションを行った。Pan ら [4] は、流体シミュレーションに対してキーフレームを指定し、スケッチを加えることで、スケッチに沿った流れのシミュレーションを行った。Emilien ら [1] は、軌跡に沿った滝を生成するシステムを提案した。

本システムでは、入力として 3D 空間を考慮したモーションを扱うため、3D 流体シミュレーションにおいて直感的に座標空間を扱うことができる。また、流体制御システムとして、形状制御などの流体のモデリングと、流れの向きや制御する範囲の指定などの流体シミュレーションのフレームワーク設定の両方がモーションデータによって操作できる。

コンピュータグラフィックスソフトウェアである blender [14], Houdini [15], RealFlow [17], NVIDIA FleX [16] など流体シミュレーションを行うことができる。しかし、このようなソフトウェアで流体シミュレーションによるアニメーションを制作する場合、ひじょうに多くの工数が必要となる。本システムであれば、1つのモーションに複数のパラメータ操作をもたせることができる。これにより、少ない工数で流体シミュレーションによるアニメーションが制作できる。

本稿では、直接操作によって流体をシミュレーションする空間を先行報告 [2], [13] の 2 次元から 3 次元に拡張した。

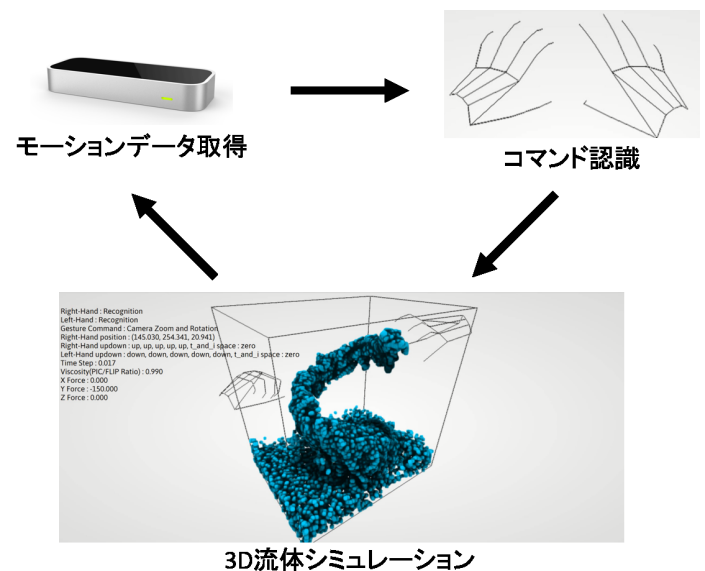


図 1 本システムの処理ループ



図 2 本システムの使用例

3. アプローチ

図 1 に本システムの処理ループを示す。キャプチャデバイスには Leap Motion 社 Leap Motion を使い、両手のモーションデータを用いる。右手と左手のモーションの組合せからハンドジェスチャを認識し、ジェスチャコマンドに応じて流体挙動を制御しながら 3D 空間上で流体シミュレーションを実行する。図 2 に本システムの実際の使用例を示す。

4. モーションデータの識別

Leap Motion により取得した両手のモーションデータから、ジェスチャを識別する流れを説明する。両手各指の相対位置関係、移動距離および方向をもとにジェスチャを識別する。まず、各指の関節位置ベクトル、手のひらの位置ベクトルを取得する。これらの位置ベクトルの組合せから

方向ベクトルを求める。次に、これらの方向ベクトルの組合せから内積を計算し、閾値を使い判別することで、特定の両手各指の相対位置関係とする。最後に、前フレームの位置ベクトルを用いて移動距離を求め、上記で求めた内積を用いて特定の移動方向を検出する。

ジェスチャでは右手の操作を基準とし、左手の操作は右手の操作の補助的な役割を担う。

5. 3D 流体シミュレーション

本システムには、格子法と粒子法のハイブリット流体シミュレーションである PIC/FLIP (Particle-In-Cell/Fluid-Implicit-Particle) 法 [8] を用いる。PIC/FLIP 法では、格子法による非圧縮性と粒子法による移流表現の両方を実現することができる。また、PIC 法による圧縮性の高い流体と FLIP 法による非圧縮性の高い流体を混合することで、流体の粘度を調節することができる。

流体の支配方程式にはナビエ・ストークス方程式を用いる:

$$\rho \frac{D\mathbf{v}}{Dt} = \mathbf{F}^{\text{pressure}} + \mathbf{F}^{\text{viscosity}} + \rho\mathbf{g} + \mathbf{F}^{\text{shape}} + \mathbf{F}^{\text{external}} \quad (1)$$

PIC/FLIP 法ではこの支配方程式を格子法で解く。このとき、形状制御で用いる $\mathbf{F}^{\text{shape}}$ と外力場生成で用いる $\mathbf{F}^{\text{external}}$ を支配方程式に追加する。

6. ジェスチャコマンド

6.1 ジェスチャコマンドリスト

本システム上で実行可能なコマンドのリストを表 1 に示す。ジェスチャによって制御できるコマンドはフレームワーク、シミュレーション、モデリングに大別できる。フレームワークでは、流体シミュレーション全体のセッティングや流体のパラメータに直接関わらない制御を行う。シミュレーションでは、通常の流体シミュレーションの支配方程式に用いられるパラメータを制御する。モデリングでは、流体の形状制御を行う。

6.2 コマンド遷移

表 1 のコマンド遷移により選択、実行、終了を行う。図 3 のようにコマンドの選択は階層構造になっている。初期状態はフレームワーク制御である。この状態で選択できるジェスチャコマンドは 1 つ下の階層にある流体制御、再起動、視点移動、制御範囲の指定、タイムステップ、空間サイズである。選択を行った後に、実行をするとコマンドが起動する。流体制御のジェスチャコマンドを実行した場合、さらに 1 つ下の階層である粘性、外力場生成、形状制御、形状サイズ調節、流れ場のコマンドを選択できるようになる。コマンドを終了すると 1 つ上の階層に遷移する。図 3 のように、コマンドの選択は右手を横に払うジェスチャ、コマンドの実行は右手の甲を上にして握るジェスチャ、コマンドの終了は右手の甲を下にして握るジェスチャで行う。

表 1 コマンドリスト (現在実行可能でないコマンドも含む)

| No | 分類 | 名前 | 内容 |
|----|----------|---------|-------------------------------|
| 0 | | コマンド遷移 | ジェスチャコマンドの切替え、実行、終了 |
| 1 | フレームワーク | 視点移動 | シミュレーション空間を中心に視点を移動 |
| 2 | | 再起動 | ジェスチャコマンドによる設定と流体シミュレーションの再起動 |
| 3 | | 空間サイズ | シミュレーション空間のサイズを調節 |
| 4 | | 制御範囲指定 | 流体粒子/制御粒子をジェスチャコマンド適用対象に指定 |
| 5 | | タイムステップ | シミュレーションのタイムステップを調節 |
| 6 | シミュレーション | 粘性 | 流体の粘性を調節 |
| 7 | | 外力場生成 | 特定方向の外力場を生成 |
| 8 | モデリング | 形状制御 | ジェスチャコマンドによる軌跡に沿った流体の任意形状制御 |
| 9 | | 形状サイズ調節 | 形状制御を行った流体の形状サイズを調節 |
| 10 | | 流れ場 | 形状制御を行った流体を流れ場に変更 |

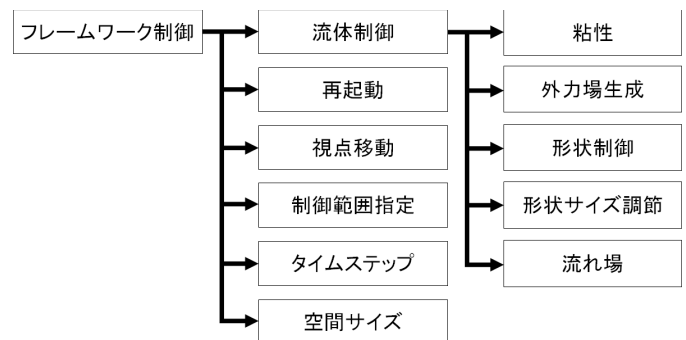


図 3 コマンド選択の階層

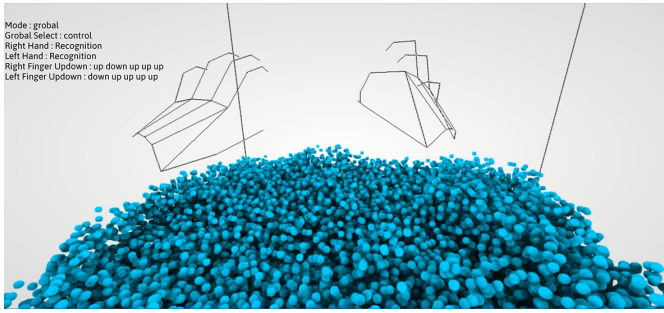
6.3 フレームワーク

6.3.1 再起動

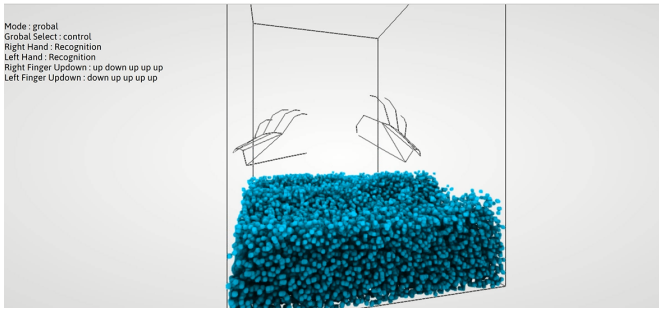
シミュレーションの再起動を行う。コマンド遷移で再起動を実行と同時に実行される。再起動では、視点移動以外の流体シミュレーションの制御状態をすべて初期化する。

6.3.2 視点移動

シミュレーション空間内の視点位置を移動する。視点はシミュレーション空間へ向いたまま上下左右に移動、接近または遠ざかることができる。結果として、画面上にはシミュレーション空間に対する上下左右の回転、拡大縮小として反映される。視点移動は右手の親指先と人差し指先を合わせたジェスチャにより起動する。このジェスチャを維持したまま、起動位置に対して上下方向に移動させると上下回転、左右方向に移動させると左右回転、奥行き方向に移動させると拡大縮小となる。図 4 のように、画面に投影されている両手の表示はシミュレーション空間と独立しているため、視点移動前後でも表示位置は変わらない。



(a) 視点が近づくことによるシミュレーション空間の拡大表示



(b) 視点が離れることによるシミュレーション空間の縮小表示

図 4 視点移動

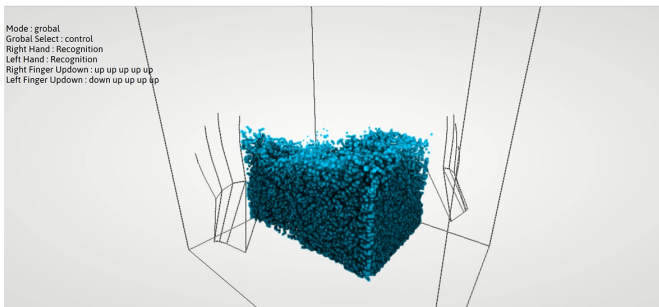


図 5 XZ 方向の空間サイズの調節

6.3.3 空間サイズ

シミュレーション空間の XYZ 方向のサイズを変更する。図 5 のように、右または左の手のひらを押すまたは引くジェスチャによってサイズを調節する。手のひら位置および方向が、現在の視点におけるシミュレーション空間の XY 面, YZ 面, XZ 面の位置および方向と一致しているとき、ジェスチャが起動し、その後、手のひらを押しと空間のサイズが小さくなり、引くと空間のサイズが大きくなる。

6.3.4 タイムステップ

シミュレーションのタイムステップを制御する。右と左の手のひらを合わせるジェスチャによりコマンドが起動する。この状態では、タイムステップがゼロとなり流体シミュレーションが停止する。このジェスチャから右手と左手を離していくと、右手と左手の距離が大きくなるほど、タイムステップがゼロから増えていくため、シミュレーションをスローに調節することができる。右手と左手の距離が一定値を超えると、タイムステップが初期状態と同じ

になる。シミュレーションを止める、またはスローにすることで特定の流体シミュレーションの様子を観察することができる。

6.3.5 制御範囲の指定

流体の粒子および制御粒子を指定することで、特定の粒子のみに流体制御を働かせる。初期状態では粘性の変更、外力場の影響、形状制御に用いる流体の粒子はすべての流体の粒子が対象となる。制御範囲の指定で流体の粒子を指定すれば、これらの制御に用いられる流体の粒子を限定することができる。また、制御範囲の指定で制御粒子を指定することで、特定の制御粒子の形状サイズを変更することも可能となる。

6.4 シミュレーション

6.4.1 粘性の調節

粘性の調節では、PIC/FLIP 法における PIC 法で求めた速度と FLIP 法で求めた速度の混合比 α を変更する。

$$\mathbf{v} = (1 - \alpha) \mathbf{v}_{\text{pic}} + \alpha \mathbf{v}_{\text{flip}} \quad (2)$$

PIC 法で求めた速度 \mathbf{v}_{pic} は、支配方程式により更新した格子速度となる。FLIP 法で求めた速度 \mathbf{v}_{flip} は、支配方程式により更新した格子速度から前フレームの格子速度を引き、粒子速度を足した速度となる。PIC 法で求めた速度 \mathbf{v}_{pic} では粘度の高い流体が得られ、FLIP 法で求めた速度 \mathbf{v}_{flip} では粘度の低い流体が得られる。この混合比 α をジェスチャにより調節する。図 6 のように、右の手のひらを下向きにするジェスチャにより起動し、手の位置が起動位置よりも高ければ α が大きくなり粘度が低くなる。手の位置が起動位置よりも高ければ α が小さくなり粘度が高くなる。

6.4.2 外力場生成

流体に任意方向への外力を加える。図 7 のように、右または左の手のひらを押すまたは引くジェスチャにより、手のひらが移動した方向に対して、移動距離に応じた外力を発生させる。初期状態の流体シミュレーションでは下向きの重力が外力として発生している。

6.5 モデリング

6.5.1 形状制御

ジェスチャにより、流体を任意形状に変形する。形状制御には、Thurey ら [8] の制御粒子による流体制御の手法を用いる。流体粒子を制御粒子に引き付けることで、任意の流体形状が表現される。制御粒子への引力は式 (1) の外力項に組み込む。粒子 i における制御粒子 c への引力は以下の式で表される:

$$\mathbf{F}_i^{\text{attraction}} = w \sum_{c_j \in C_i} \alpha_{c_j} \frac{\mathbf{r}_{c_j} - \mathbf{r}_i}{\|\mathbf{r}_{c_j} - \mathbf{r}_i\|} W(\mathbf{r}_{c_j} - \mathbf{r}_i, h), \quad (3)$$

ここで、 w は係数であり、正のとき引力、負のとき反発力

6.5.3 流れ場

制御粒子によって変形している流体を流れ場に変える。制御粒子に引き付けられている流体に対して、引力に加え、次に配置された制御粒子へ向かう力を発生させる。これにより、制御粒子を配置した始点から終点に向かう流れ場が生成される。

7. 実装

開発および実行環境として、PC(CPU: Intel Core i7-5500U 2.40GHz, RAM: 8.00GB)を使用した。リアルタイムシミュレーション可能な粒子数は20,000から80,000までである。開発言語はJavaScriptを用いて、JavaScriptのグラフィックスAPIであるWebGLを用いて描画した。図4～図9は、上記の環境、粒子数60,000で実行している。

8. 結論と今後の課題

本論文では、直接操作を用いた直感的な流体挙動の制御を提案した。本システムにより、流体シミュレーションに知識のないユーザが容易に3次元流体を制御できるシステムを提供できた。

今後の課題として、実装が完了していないコマンドとモーションの対応付け(6.3.5項, 6.5.3項)が必要である。表1のコマンドがすべて実行可能な状態で流体を制御する上で最低限の項目が揃っているかを検討する必要がある。

また、本論文では直接操作により直感的に流体を制御することを目的としているため、流体の制御に対して、どのようなジェスチャを当てはめれば直感的であるといえるかを客観的に決める必要がある。複数のユーザに流体の制御の様子のみを見せ、どのようなハンドモーションを当てはめるかをアンケートし、これらのアンケート結果の共通点から、ジェスチャを決める必要がある。

謝辞

本研究の一部は、基盤研究(A)26240015および基盤研究(A)17H00737の支援により実施された。

参考文献

- [1] A. Emilien, P. Poulin, M. Cani, and U. Vimont: "Interactive Procedural Modelling of Coherent Waterfall Scenes," *Computer Graphics Forum*, Volume 34, Issue 6, pp. 22-35, 2015.
- [2] Y. Hayakawa and I. Fujishiro: "2D Fluid Shape Design by Direct Manipulation," in *Proceedings of the 5th IEEE International Workshop on Image Electronics and Visual Computing*, 3B-2, 2017.
- [3] A. McNamara, A. Treuille, Z. Popovi, and J. Stam: "Fluid Control Using the Adjoint Method," *ACM Transactions on Graphics*, Volume 23, Issue 3, pp. 449-456, 2004.
- [4] Z. Pan, J. Huang, Y. Tong, C. Zheng, and H. Bao: "Interactive localized liquid motion editing," *ACM Trans-*

- actions on Graphics*, Volume 32, Issue 6, Article No. 184, 2013.
- [5] K. Raveendran, N. Thuerey, C. Wojtan, and G. Turk: "Controlling Liquids Using Meshes," in *Proceedings of the 2012 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 255-264, 2012.
- [6] L. Shi and Y. Yu: "Taming Liquids for Rapidly Changing Targets," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pp. 229-236, 2005.
- [7] A. Stomakhin and A. Selle: "Fluxed Animated Boundary Method," *ACM Transactions on Graphics*, Volume 36, Issue 4, Article No. 68, 2017.
- [8] N. Thuerey, R. Keiser, M. Pauly, and U. Rude: "Detail-preserving fluid control," in *Proceedings of the 2006 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 7-12, 2006.
- [9] K. Ueda and I. Fujishiro: "Adsorptive SPH for directable bleeding simulation," in *Proceedings of the 14th ACM SIGGRAPH International Conference on Virtual Reality Continuum and its Applications in Industry*, pp. 9-16, 2015.
- [10] J. Yoshino and R. Tsuruno: "Interactive fluid control by shape matching," *芸術科学会論文誌*, Volume 13, Article No. 3, pp. 169-176, 2014.
- [11] S. Zhang, X. Yang, Z. Wu, and H. Liu: "Position-based fluid control," in *Proceedings of the 19th Symposium on Interactive 3D Graphics and Games*, pp. 61-68, 2015.
- [12] Y. Zhu, D. Charypar, and R. Bridson: "Animating sand as a fluid," in *Proceedings SIGGRAPH '05 ACM SIGGRAPH*, pp. 965-972, 2005.
- [13] 早川雄登, 藤代一成: "直接操作による2次元流体の簡易形状設計," in *NICOGRAPH2016*, P-19, 2016.
- [14] *blender*, <https://blender.jp/> (最終アクセス日: 2017年10月17日).
- [15] *Houdini*, <http://sidefx.jp/> (最終アクセス日: 2017年10月17日).
- [16] *NVIDIA Flex*, <https://developer.nvidia.com/flex> (最終アクセス日: 2017年10月17日).
- [17] *RealFlow*, <http://www.nextlimit.com/realflow/> (最終アクセス日: 2017年10月17日).
- [18] *WebGL*, <https://www.khronos.org/registry/webgl/> (最終アクセス日: 2017年10月17日).