

流体シミュレーションを統合した対話的な形状設計手法

A new approach for interactive shape modeling combined with fluid simulation

三谷 純 五十嵐 健夫*

Summary. 近年のCG分野では、物理シミュレーションの研究が盛んに行われており、現実世界の剛体および流体の挙動をリアルにそして高速に表現できるようになりつつある。しかしながら、これらのシミュレーションはユーザーが記述した初期状態に基づいて物体の動きを計算するものであるため、意図した挙動を得るためには初期状態の変更とシミュレーション結果の観察、という時間のかかる試行の繰り返しが必要である。そこで本研究ではユーザーが対話的に初期状態を変更しつつ、その変更に伴う物体の挙動を容易に把握できるようなインタフェースの提案を行う。このようなインタフェースにより、製品設計の上流工程に物理シミュレーションを統合することが可能となれば、目的の機能を果たす形状の試行錯誤を軽減できると考えられる。具体例としてティーポットの形状設計を挙げ、ユーザーが対話的にティーポットの形状を変形でき、その操作に伴って注ぎ口から流出する水の挙動が変化する様子を効果的に提示するシステムの開発を行った。提案システムではティーポットの断面形状の設計を対象とし、形状操作にAs Rigid As Possible Curve Editingの技術を用いた。水の挙動解析にはレベルセット法に基づく流体シミュレーションを採用し、シミュレーションの高速化のためにGPUを援用した。挙動の提示には時系列で変化する流体の流れをアルファ混合した1枚の静止画を用いた。これにより、対話的な形状操作とその操作が及ぼす流体の挙動の変化を容易に確認できるようになった。

1 はじめに

近年のCG分野では、物理ベースのシミュレーションの研究が盛んに行われており、現実世界の剛体および流体の挙動をリアルにそして高速に表現できるようになりつつある。これらは衝突を伴う複数の剛物体の挙動や流体の挙動など、人が手作業で動作を記述することが困難なシーンを効率的に構築する上で必要不可欠な技術である。しかしながら、これら多くのシミュレーションはユーザーが記述した初期状態に基づいて物体の動きを計算するものであるため、各物体がシーンの中で意図した通りの挙動をするように制御することは極めて困難である。ユーザーは目的のアニメーションを得るために、初期状態の変更（場合によっては物理パラメータの変更）とシミュレーション結果の観察、という時間のかかる試行錯誤を繰り返し行う必要がある。そこで本研究では、形状の設計を支援することを目的に、ユーザーが対話的に初期状態を変更しつつ、その変更に伴う物体の物理的な挙動の変化を容易に把握できるようなインタフェースの提案を行う。

本研究では具体例としてティーポットの形状を設計することを取り上げ、ユーザーが対話的に形状を変形し、その注ぎ口から流出する水の挙動を確認できるアプリケーションの開発を行った。本システム

ではプロトタイプ版という位置づけで2次元平面上での断面形状を対象とした。既存のアプローチでは、形状設計が完了した後にシミュレーションを行うことが一般的であるが、本研究で提案するシステムでは、形状設計の操作に連動してシミュレーションが高速に行われ、その結果が1枚の図として提示される。このような形状設計と物理シミュレーションを統合した設計インタフェースを提案することで、製品設計の上流工程における試行錯誤のコストが低減できる可能性があることを示すことが本研究の目的である。

本稿の2章で関連研究を示し、3章では提案手法の詳細を記す。4章で結果を示し、最後に5章でまとめと今後の展望について述べる。

2 関連研究

CGの分野に物理シミュレーションを取り入れる研究は古くから行われ、近年では目覚ましい成果が挙げられている。例えば、[7]の研究では物体の破壊が組み込まれるなど、よりリアルな結果が得られるようになってきている。一方で、望んだ結果を得るためにはユーザーが試行錯誤でパラメータを調整する必要があり、労力を要する作業が伴う。そこで、パラメータの調整による間接的な制御でなく、より直接的にシミュレーション結果を制御可能なインタフェースについて、いくつかの手法が提案されている。[11]は、シミュレーション実行中に物体の任意の点を選択し、それを対話的に操作しシミュレーショ

Copyright is held by the author(s).

* Jun Mitani, 筑波大学大学院 システム情報工学研究科, JST ERATO, Takeo Igarashi, 東京大学大学院 情報理工学系研究科, JST ERATO

ン結果を編集することが可能なインタフェースを提案している．このシステムでは初期パラメータを自動調整することで目的のアニメーションを生成している．文献 [13] では、複数の異なる物理パラメータおよびランダムな要素を含む剛体シミュレーションを高速に行い、それらによって得られる動きの軌跡を同一画面上に表示し、ユーザーが目的とする挙動に近いものが得られるように、特定の軌跡を選択するインタフェースを提供している．これにより、目的とする物体の挙動を選択的に取得できるようになっている．また、より大規模で衝突が何千回も発生するようなシーンにも適用可能とした手法が [13] によって提案されている．しかし、この手法ではシミュレーション結果の正確性よりも映像制作者の希望とする表現を実現することを重視しており、実際にはありえない結果を生成してしまうため、現実世界の製品の形状設計に用いることはできない．また、これらのシステムではシミュレーション結果を単純な数本の軌跡として表現している．本稿で提案するインタフェースでは、シミュレーション結果を1枚の画像に重ねて描画することで、ユーザは流体の挙動をより容易に把握できるようになっている．VRにおける仮想世界の中に物理シミュレーションを取り入れる研究も行われている．[4] は物理エンジン CORIOLIS を内蔵したワークベンチと専用の三次元入力装置によるシステムを提案している．このシステムを用いることで、ユーザは機械部品の組み立て作業を仮想空間上でシミュレーションすることができる．しかし、個々のパーツの形状を変化させることはできないため、形状の設計段階で用いることはできない．[6] はぬいぐるみの制作を支援するシステムの中に物理シミュレーションを取り入れている．布のふくらみを設計段階でユーザーに提示することで、実際に布を縫い合わせた後にできる形を事前に把握することができる．本稿で提案するインタフェースの目的は、この研究に近く、形状設計の段階に物理シミュレーションを組み合わせることで、目的の形の設計を支援可能であると考えている．

3 提案手法

本研究ではティーポットの形状を設計することを取り上げ、ユーザーが対話的に形状を変形し、その注ぎ口から流出する水の挙動を確認できるようにする．これらは2次元平面上で行うこととし、ティーポットの形状は断面形状を対象とし、流体シミュレーションは2次元で行う．シミュレーション結果は1枚の図として視覚的に確認できるような提示を行う．以降で、断面形状を対話的に変形するためのインタフェース、入力された形状から水が流出する様子を得るための流体シミュレーション、その結果を提示するための手法、およびこれらを統合したシステムの実装について述べる．

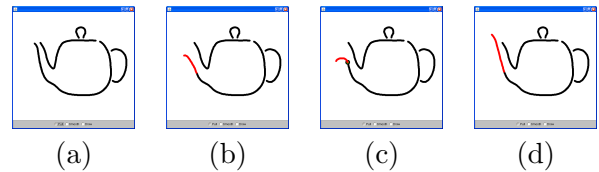


図 1. ティーポットの断面形状の対話的な操作．(a) 初期形状，(b),(c),(d) ユーザーによる注ぎ口の変形

3.1 断面形状の変形インタフェース

形状の編集は対話的にかつ直観的に行えることが望ましい．本研究では、ティーポットの断面形状を2次元平面上で操作するためのインタフェースとして、文献 [5] で提案されている曲線編集用のインタフェースを使用することとした．この方法では、ユーザーが掴んで動かしている点を制約として、曲線の局所的なゆがみが最小になるように形状を瞬時に決定することが可能である．変更する領域の指定、および変形操作がマウスカーソルの簡単な操作で行えるため、例えばティーポットの注ぎ口の形状を直感的な操作で変更することができる．実装システムでは、図 1(a) のようなティーポットの初期形状をユーザーに提示し、ユーザーが目的の場所をマウスカーソルでドラッグして変形可能にした(図 (b) ~ (d))．このようにして変形した形状に対して、水の流出シミュレーションを行う．

3.2 流体のシミュレーション

本研究で用いる流体シミュレータでは、以下に示すオイラー方程式と連続の式を計算することによって、流体の挙動をシミュレーションする．

$$\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \nabla) \mathbf{u} = -\frac{\nabla p}{\rho} + \mathbf{f}$$

$$\nabla \cdot \mathbf{u} = 0$$

ここで \mathbf{u} は速度、 p は圧力、 ρ は密度、 \mathbf{f} は外力である．本稿では外力は重力のみとし、表面張力や粘性は考慮しないものとした．なお流体シミュレーションの詳細は、文献 [1] を参照されたい．

3.2.1 レベルセット法

本稿では、流体の領域の表現にレベルセット法 [9, 8, 12] を用いる．レベルセット法は界面を陰関数として表現する方法で、流体のシミュレーションでよく使用される．レベルセット関数 $\phi(\mathbf{x}, t)$ を、 $\phi(\mathbf{x}, t) < 0$ となる領域を水、 $\phi(\mathbf{x}, t) > 0$ となる領域を空気、 $\phi(\mathbf{x}, t) = 0$ となる領域を境界と定義する．(以下、レベルセット関数 $\phi(\mathbf{x}, t)$ を、 ϕ と略記する．) レベルセット関数 ϕ の時間発展は、次の式によって得られる．

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0$$

3.2.2 流体シミュレータの実装

本稿では速度をセル境界の中心に、圧力などその他の変数をセルの中心に配置する staggered 格子を用い、移流の計算には無条件安定でかつ実装が容易という利点がある一次精度の Semi-Lagrangian 法を使用した。二次元の圧力のポアソン方程式を離散化した連立一次方程式は、GPU で実装しやすい Jacobi 法を用いて計算した [2]。また、レベルセット関数が符号付き距離関数の性質を保つよう、定期的にレベルセット関数の再初期化を実行した。流体の運動方程式は、 $\phi \leq 0$ の領域のみ計算される。しかしながら、境界付近の移流の計算には、 $\phi > 0$ の領域の速度も必要になる。よって、 $\phi > 0$ となる領域の速度は、 $\phi \leq 0$ の領域の値を外挿した [3, 10]。

以上に述べた計算手法を GPU 上で動作するように実装した。実装にあたっては文献 [2] を参考とした。

3.3 シミュレーション結果の提示手法

シミュレーション結果は、水の挙動を把握しやすくするために、アニメーションを要約した 1 枚の静止画で提示することとした。前節で述べたシミュレーションをバックグラウンドで実行し、一定時間間隔でレンダリングを行い、再生された N 枚の画像に対して、それぞれの α 値を $1/N$ に設定して混合を行う。これにより、シミュレーション終了後に 1 枚の要約画像が得られる。なお、要約画像が得られるまでの間は、逐次要約画像が更新されながら表示されることになる。

3.4 システムの統合

上記で述べた各要素を統合し、ユーザーが行う形状操作に同期して、シミュレーション結果が表示されるようにする。具体的には以下のようにシステムを構成した。

- 3.1 節で述べたインタフェースを実装した形状操作のウィンドウで、ユーザーが対話的にティーポットの形状を操作すると、その結果が自動的にファイルに保存される。ファイルには形状を表す線分の集合が格納され、このファイルを経由して形状データがシミュレータに渡される。
- シミュレータでは形状が更新される時点でシミュレーションを初期化し、まず最初に渡されたティーポットの形状に対して水を満たす操作を行う。今回の実装では、満たされた状態を作るために内部の点（今回は全頂点の平均座標を使用）から一定の時間 (t_1) だけ水を流入させることを行った。その後シミュレーション結果の要約画像生成を開始し、一定時間 (t_2) 経過後に、要約画像の生成およびシミュレーションを終了する。シミュレーシ

ョンの間、画面右下部からは常に水を排出するようにする (t_1 および t_2 の値は事前の試行により決定しておくものとする)。

- シミュレーション終了後も、生成された画像は提示を続け、ユーザーの操作によって形状に対する変更があった時点で、上記の処理を再び実行し直して画像の更新を行う。

この実装により、ユーザーは形状操作のウィンドウで形状デザインのみを行えばよく、シミュレーションの開始および終了を明示的に指定せずに結果を確認することができる。

4 実装結果

これまでに述べたシステムを、CPU:Core 2 Duo 2.33GHz, GPU:Quadro NVS 290, RAM:1GByte の PC 上に実装した。3.1 節で述べたシステムによってユーザーが形状設計を行うウィンドウ領域は 512×512 で、図 1 に示すような雛型となるティーポットの断面形状が配置されている。ユーザーが形状の変形を行うと、その形状に基づく流体の挙動がバックグラウンドで実行され、その要約結果が別ウィンドウに提示される。流体シミュレーションは入力されたティーポットの形状を 60 度傾けた状態で行われるようにした。流体シミュレーションに用いた空間は 50cm 四方であり 128×128 の格子を用いた。各形状に対し、シミュレーション時間で 1 秒間だけ水を流入させ、内部が水で満たされた状態から約 2 秒間分の流出シミュレーションを行った。要約画像は約 0.08 秒間のシミュレーション毎に取得したものをアルファ混合して生成した。ユーザーが形状操作を行ってから最終的な要約画像が出力されるまでの時間はおよそ 20 秒であった。ユーザーが形状操作を行ってから要約画像が出力されるまでの時間は、その進行に応じて画像を更新することでシミュレーションが終わるまで待たずに変化の様子を見ることができる。図 2(a) ~ (f) に 6 種類の異なる形状に対して実行した結果を示す。下段はシミュレーション終了後に要約画像として提示されたものである。ティーポットの注ぎ口の形状が流出する水の挙動に与える影響を確認することができる。

5 まとめと展望

本稿では、流体シミュレーションを統合した対話的な形状設計手法を提案した。具体例として、ティーポットの形状設計を挙げ、対話的な形状操作によって変化する水の挙動を静止画として確認できるシステムの開発を行った。形状を変形してからシミュレーション結果の画像が提示されるまでに若干の時間を要したが、今後のハードウェア性能の向上などでシミュレーション速度がさらに高速になれば形状操作と水の挙動の確認を、よりスムーズに行えるように

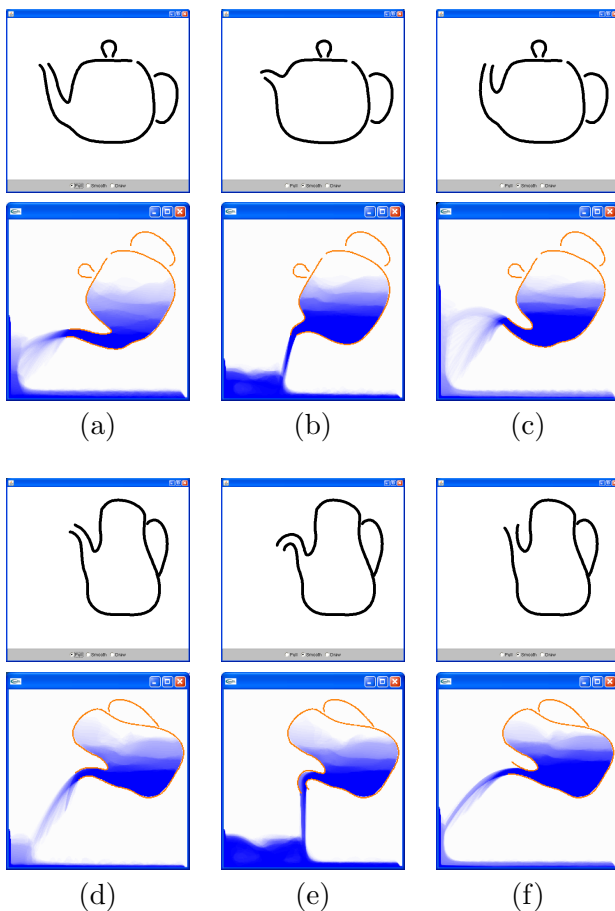


図 2. 提案システムの実行結果．上段:入力形状，下段:シミュレーション結果

なることが可能と考えられる．本稿で提案したインタフェースでは，ユーザーは形状の変形を行うだけでよく，シミュレータに対する，開始・終了の操作を行う必要がないという利点がある．これにより，形状を設計の段階から対象物の挙動を確認できるため試行錯誤の時間を大幅に短縮することが可能であると考えられる．CGの分野では表示されるシミュレーション結果が現実と大幅に異なっている場合も多く、人が見て違和感を感じなければ良いとする場合が多く、目的のアニメーションを得るために物理パラメータを変更することも多いが、本稿で使用した流体シミュレーションは (CGの分野でも使われている手法であるが) 物理ベースの手法であるため、実世界での挙動と大幅に異なることはないと考えられる．実世界での再現性については、実際にものを製造して比較評価する必要があるが、本システムを実際のものづくりの意匠設計や教育用途のアプリケーションなどに活用できるのではないかと考えている．本研究では、新しいインタフェースとして流体シミュレーションを統合した形状設計手法を提案したが、今後の課題としては、さらなるシミュレーションの高速

化およびシステムの3次元への拡張、他の例題への応用、実物との比較評価などが挙げられる．

謝辞

流体シミュレータの実装に際し八木雄一氏の助力を得た．

参考文献

- [1] R. Bridson, R. Fedkiw, and M. Muller-Fischer. Fluid simulation: SIGGRAPH 2006 course notes. In *ACM SIGGRAPH 2006 Courses*, pp. 1–87, 2006.
- [2] K. Crane, I. Llamas, and S. Tariq. Real Time Simulation and Rendering of 3D Fluids. In *GPU Gems 3*. Addison-Wesley, 2007.
- [3] D. Enright, S. Marschner, and R. Fedkiw. Animation and rendering of complex water surfaces. *ACM Trans. Graph.*, 21(3):736–744, 2002.
- [4] B. Fröhlich, H. Tramberend, A. Beers, M. Agrawala, and D. Baraff. Physically-Based Manipulation on the Responsive Workbench. In *Proceedings of the IEEE Virtual Reality 2000 Conference*, pp. 5–11, 2000.
- [5] T. Igarashi, T. Moscovich, and J. F. Hughes. As-rigid-as-possible shape manipulation. *ACM Trans. Graph.*, 24(3):1134–1141, 2005.
- [6] Y. Mori and T. Igarashi. Plushie: An Interactive Design System for Plush Toys. *ACM Trans. Graph.*, 26(3):articleNumber45, 2007.
- [7] J. F. O'Brien and J. K. Hodgins. Graphical modeling and animation of brittle fracture. In *Proceedings of ACM SIGGRAPH 1999*, pp. 137–146, 1999.
- [8] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Springer, 2003.
- [9] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.
- [10] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local level set method. *J. Comput. Phys.*, 155(2):410–438, 1999.
- [11] J. Popović, S. M. Seitz, M. Erdmann, Z. Popović, and A. Witkin. Interactive manipulation of rigid body simulations. In *Proceedings of ACM SIGGRAPH 2000*, pp. 209–217, 2000.
- [12] J. A. Sethian. *Level Set Methods and Fast Marching Methods*. Cambridge University Press, 1999.
- [13] C. D. Twigg and D. L. James. Many-worlds browsing for control of multibody dynamics. *ACM Trans. Graph.*, 26(3):articleNumber14, 2007.