

修士論文

平成 22 年度 (2010)

3次元入力デバイスと粒子法を用いた ガラス細工体験システム

東京工科大学大学院
バイオ・情報メディア研究科 メディアサイエンス専攻

瀬田 陽平

修士論文

平成 22 年度 (2010)

3次元入力デバイスと粒子法を用いた ガラス細工体験システム

指導教員 渡辺 大地

東京工科大学大学院
バイオ・情報メディア研究科 メディアサイエンス専攻

瀬田 陽平

論文の要旨

論文題目	3次元入力デバイスと粒子法を用いた ガラス細工体験システム
執筆者氏名	瀬田 陽平
指導教員	渡辺 大地
キーワード	粒子法, 流体, GPGPU, ボリュームレンダリング
[要旨]	<p>近年, 現実の芸術手法を 3DCG 技術を用いて仮想空間上で再現する試みが多く行われている。例として仮想ガラス細工システム, 仮想粘土細工システム, 仮想陶芸システム等が存在する。本研究ではその中でもガラス細工体験システムに着目した。ガラス細工には、ガラスの加工法により様々な手法が存在しているが、本論文では、ガラスを加熱し熔融させ変形を行うバーナーワークの手法を題材とする。既存のガラス細工体験システムにおいて、高温状態のガラスの特徴である動的な位相変化を伴う形状変形については実現されていなかった。しかし、ガラス細工の一手法であるバーナーワークでは引き伸ばし変形に伴う形状の分裂を繰り返すことで様々な形状を作り出す手法であるため、この特徴の表現は不可欠である。そこで本研究ではバーナーワークの体験システムに利用可能な形状の分裂が可能な仮想ガラスモデルの実現を目的とした。本手法では粒子法の一つである SPH 法 (Smoothed Particle Hydrodynamics Method) を用い高温時のガラスの変形挙動を粒子の運動として表現した。また高温時のガラスの特徴である赤熱現象と粘性変化を考慮したモデル化を行いその結果を確認した。ガラスの滑らかな表面の特徴を考慮した描画を行うため粒子の可視化手法としてボリュームレンダリングを用いた。本手法に用いる SPH 法とボリュームレンダリングは一般的に計算コストの高い手法であるため、体験システムとして対話的な操作を可能にするため GPGPU を用いた並列計算を行うことで高速化を図った。これらを用いた提案手法を実装し、その有効性を検証した。</p>

A b s t r a c t

Title	Particle-based glasswork simulation
Author	Youhei Seta
Advisor	Taichi Watanabe
Key Words	Particle based method, Fluid simulation,GPGPU,Volume Rendering
[summary] <p>In recent years, various method of reproducing technique of art of realities in virtual space that uses 3DCG technology are developed. There are virtual glass work system, virtual clay work system, and virtual ceramic art system as an example. In this paper,we especially focused the virtual glass work system. In the glasswork, various techniques exist depending on the processing method of the glass. This paper describes development of virtual experience system of the burner work which is a technique of glasswork. The burner work is a technique for deformation of a heated glass by a flame of burner. Existing virtual glasswork systems not discussed the feature of split in high-temperature state glass. But, the feature of shape split is important expression in deformation technique of burnerwork. Therefore, the aim of this study is to propose new glass model that can be able to split, and a method of considering other feature of heated glass. In this system, the movement of particle using the Smoothed Particle Hydrodynamics Method expressed the deformation of the heated glass. And the modeling of this system can express the feature of red glow effect and viscous change in heated glass. The method of particle visualization and red glow effect used Volume Rendering that can express the feature of smooth surface in the glass. Moreover, processing speed which can secure sufficient interactive nature was realized by performing big calculation processing of load on GPU. Lastly, we builded the system using the proposed method,and verified its effectiveness.</p>	

目次

第1章	はじめに	1
1.1	研究背景	2
1.2	論文構成	7
第2章	提案手法	8
2.1	SPH法によるガラスのモデル化	9
2.1.1	カーネル近似による物理量の補間	9
2.1.2	カーネル関数	10
2.1.3	支配方程式の離散化	12
2.1.4	密度計算	12
2.1.5	圧力項の離散化	13
2.1.6	粘性項の離散化	13
2.1.7	粒子の温度変化	14
2.1.8	表面張力	16
2.2	速度と座標の更新	18
2.3	近傍粒子探索	18
2.4	ボリュームレンダリングによる表面の描画	20
2.5	ガラスの赤熱表現	23
2.6	くびれ補正による突起形状の生成	24
2.7	3次元入力デバイスによる操作	27
2.8	計算手順	28
2.9	GPGPUによる高速化	29
2.10	本システムの概要	30
第3章	検証と考察	32
3.1	速度検証	33
3.2	引き伸ばし変形による突起形成の検証	34
3.3	温度変化に伴う赤熱現象の検証	36
3.4	表面形状の描画の検証	38
第4章	おわりに	40

謝辭	42
参考文献	44

目 次

1.1	バーナーワークの様子	3
2.1	本手法における温度と粘性の関係	14
2.2	くびれ補正処理を用いた引き伸ばし変形-形状変形前	26
2.3	くびれ補正処理を用いた引き伸ばし変形-操作点移動開始	26
2.4	くびれ補正処理を用いた引き伸ばし変形-突起形成後	27
2.5	くびれ補正処理を用いた引き伸ばし変形-分裂後	27
2.6	本システムを用いた形状変形の様子	31
3.1	本システムを用いた形状変形の様子1	34
3.2	本システムを用いた形状変形の様子2	35
3.3	本システムを用いた形状変形の様子3	35
3.4	本システムを用いた形状変形の様子4	35
3.5	外気による温度低下1	36
3.6	外気による温度低下2	37
3.7	外気による温度低下3	37
3.8	外気による温度低下4	37
3.9	光源移動による変化	38
3.10	光源移動による変化2	39

第 1 章

はじめに

1.1 研究背景

近年、3DCG 技術の発展に伴い、現実の芸術手法を仮想空間上で再現する試みが多く行われてきている。例として仮想粘土細工システム [1][2][3]、仮想陶芸システム [4][5]、仮想ガラス細工システム [6][7] 等を挙げることができる。これらはそれぞれ、教育利用やその芸術手法の記録等の様々な目的をもっており、これらの需要は高まっている。この流れにおいて芸術手法としてガラス細工 [8][9][10] に着目した。ガラス細工技法には大別して、ガラスを加熱し変形を行うもの、常温のまま切削切断を行うものに分けることができる。本研究では加熱技法の 1 手法であるバーナーワーク [11][12] を扱う。バーナーワークとはバーナーの炎によってガラスを局所的に加熱し、加熱部分をピンセットでつまみ引き伸ばし変形を繰り返すことで任意の形状を成形するガラス細工技法である。特徴として比較的細かな突起などの複雑な形状を扱うことに適した技法で、熟練者では動物や昆虫等の形成が可能である。本研究ではこのバーナーワークを対象としたガラス細工体験システムの構築を目的とする。

ガラス細工の技法はガラスが冷めた状態で成形する加工技法やガラスを高温で熔融しやわらかい状態で成形する加工技法など様々な技法が存在する。熱して変形する手法の 1 種としてバーナーワークがある。バーナーワークとはバーナーの炎によってガラスを熱し成型を行う技法である。主に左手にガラス棒をもち右手でピンセット等で熱したガラスをつまみ変形を行っていく。図 (1.1) はバーナーワークの様子である。



図 1.1: バーナーワークの様子

ここでガラス細工に関するガラスの性質について述べる。ガラスは温度変化により急激な粘性変化が起こる物質であり、ガラスの温度がある点以上では液体のような振る舞いをみせ、温度が低下するにつれ粘性が増し、最終的に固体となる性質をもつ。ガラス細工ではその性質を用いて液体と固体の中間の状態を維持しながら形状変形を行う。高温状態のガラスについて、引き伸ばし変形時の速度と粘性がある一定で釣り合った際、非常に細長くまでちぎれず伸びる曳糸性という性質も存在する。ガラスを高温に加熱した際にはガラスが赤く発光する赤熱現象が起こる。ガラスの赤熱現象は約 800℃あたりから赤く変わり始め、1000℃を超えたあたりから黄色みがかかったオレンジ色になり 1500℃では黄色がかかった白色にその温度の変化にしたがって変色するという性質がある。これによりガラスの温度状態を目で見ることで確認できる。また低温時のガラスは表面が滑らかでかつ無色透明であり、周囲の景色を反射屈折するという特徴がある。これらの特徴

をまとめると次の通りである。

- 温度変化による粘性変化
- 高温時の赤熱現象
- 引き伸ばした際の形状分裂
- 無色透明
- 光の反射・屈折

加熱手法であるバーナーワークでは、ガラスを軟化させるため非常に高温のバーナーの炎を用いガラスを加熱する。仮想空間ではその際の火傷などの危険性がなくなるという利点がある、また現実のバーナーワークでは一度失敗したガラスを再利用することは難しいが、仮想空間ではガラスを無駄にすることなくガラス細工を練習可能であるといえる。これらから仮想空間でガラス細工を行うことで、これまでガラス細工を体験することが難しかった年少者や初心者が気軽にガラス細工体験を行うことが可能になるといえる。

このような背景をもつ既存の加熱技法のガラス細工体験システムとして小島らはガラス細工技法の一種、吹きガラス技法についての体験システムを構築した。吹きガラス技法とは高温の炉の内部で加熱したガラスを吹き竿に巻き取り、息を吹き込み膨張させ成形するガラス細工技法である。小島らは吹き竿形状の操作デバイスを作成した。デバイスに息を吹き込みガラスモデルが膨張変形するシミュレーションを行った。またペンタブレットを用い画面上のガラスモデルに対して凹ませる変形、または膨らませる変形を可能にした。これらにより仮想空間上でインタラクティブに吹きガラス技法を体験することを可能にした。また櫛田らは、ガラスを局所加熱することで変形する部分を決定するというガラス細工の特徴を考慮した3次元モデリングシステムを構築した。変形対象の3次元モデルの頂点に対して温度情報を付加することで、変形操作を行った際の影響の度合いを変化さ

せ、ガラスの粘性を表現した。また6自由度の3次元入力デバイスを用い直感的な変形操作を可能にした。

既存手法ではそれぞれのガラス細工技法を再現するために必要な特徴を考慮したガラスのモデル化を行っているが、これらのガラスモデルでは変形に伴う分裂や合体を考慮していない。既存手法のガラスモデルを用いてバーナーワークの表現を試みた場合、引き伸ばし変形を行った際に形状の分裂を考慮していないため、変形後の形状が現実と異なることが予想できる。しかしバーナーワークのガラス細工では引き伸ばし変形に伴う分裂により、突起形状を形成することは任意の形状表現を行う際に重要な要素といえる。

そこで本研究では新たにバーナーワーク体験システムのための形状の分裂とガラスの温度による色と粘性変化の特徴を考慮したガラスモデルを提案した。分裂を伴う動的な形状変形が可能なモデルを表現するため本手法では、粒子法の一つであるSPH(Smoothed Particle Hydrodynamics)法を用いた流体シミュレーションを用いた形状表現を行った。高温時のガラスの変形を表現するため温度により粘性の変化する流体として仮想ガラスモデルを表現した。またガラスの性質である赤熱現象と滑らかな表面の表現のため粒子の集合から濃度値分布と温度分布のボリュームデータを生成しボリュームレンダリングを用いた可視化を行った。またボリュームデータをGPU内のテクスチャメモリに展開し、高速な数値の補間を行うことで計算速度を損なうことなく描画の質を向上させた。バーナーワークの引き伸ばし変形と局所的な加熱操作を可能にするため3次元入力デバイスを用いた操作により、体験システムを構築した。粒子法シミュレーションとボリュームレンダリングはいずれも計算コストが高い手法であり、対話的な操作が難しい手法であったが、本手法ではGPU(Graphics Processing Unit)を汎用計算に用いるGPGPUを用いることでシミュレーション中の処理を並列化し処理の高速化を図った。提案手法による体験システムを実装し、その形状変形について検証を行った。検証では引き伸ばし変形時のくびれを伴う突起形成を確認し、本手法のくびれ補正が有効であることを確認した。また粒子の温度変化の検証のため全粒子の温度を1500°C

とした状態での時間変化の様子を確認した。この検証により温度変化による粘性変化と赤熱現象を確認した。計算速度については粒子のみ描画においては163(ステップ数/秒)でのシミュレーションを確認し、5ステップに一度のボリュームレンダリング時には27(ステップ数/秒)の結果を得た。この描画では操作時に画面更新の遅延が発生するためボリュームレンダリング処理を高速化する技術を用いることでこの遅延を解消した上で対話的な操作が可能になるという考察を行った。以上のような検証により提案手法によるガラス細工体験システムは引き伸ばし変形によるくびれを伴う突起形成が可能であることを確認し、バーナーワーク体験システムに用いるガラスモデルとしての有効性を確認した。

本研究では新たにバーナーワーク体験システムのための形状の分裂を考慮したガラスモデルを提案し、分裂を伴う動的な形状変形を確認した。なお本研究はガラス細工体験システムの表現の向上を目的としたものであり、ガラスの物理的性質の解析等を目的としたものではないため、定性的なモデル化を行う。

本手法ではガラス細工の特徴を表現するために考慮するガラスの性質を以下で定義する。

- 温度変化による粘性変化
- 高温時の赤熱現象
- 引き伸ばした際の形状分裂
- 無色透明
- 光の反射

現実のバーナーワークではガラスを加熱することで粘性を低下させ、ガラスを成形しやすい状態におく、その際赤く発光している。その状態のガラスに対しピンセットでつまみ引き伸ばすことで任意の形状を形成する。そして、その突起は引き伸ばし操作により受けた外力がなくなったときにも形状を維持し、またその変形時には全体の体積は変化しない。その性質によりガラス細工職人は任意の形状の

生成が可能になる。そのためこれらの性質を表現する。また見た目の特徴としての透明感と、光の反射を表現する。ガラス細工体験システムにおいて、これらの特徴を考慮するため粒子法の一つである SPH 法 (Smoothed Particle Hydrodynamics Method)[13] を用い粒子の運動としてガラスの変形を考慮し、形状を定義するモデル化を行った。

1.2 論文構成

本論文の構成を以下で述べる。第 2 章では、提案手法のガラス細工のモデル化について述べる。第 3 章では、本手法を用いたガラス細工体験システムによる、結果の検証と考察を行う。第 4 章では、本研究を総括し今後の展望について述べる。

第 2 章

提案手法

本章から本研究のガラス細工のモデル化について述べる。

2.1 SPH法によるガラスのモデル化

SPH法は物体の挙動を粒子の運動によって近似する粒子法 [13] の一手法である。SPH法は粒子の運動を記述する支配方程式を変えることで汎用的な物体の物理シミュレーション [14][15] に利用可能な手法である。さらに計算コストの高い粒子法の中では比較的高速にシミュレーションが可能なることから、近年映像コンテンツやゲームコンテンツの制作現場において水、煙、炎などの人の手で作るには難しい流体の表現に利用が増えている。また運動する粒子の周囲に対し陰関数曲面を定義することで、連続体の自由表面の分裂、融合を伴う変形を扱うことが容易である。これらの特徴から本手法で考慮するガラスの特徴を表現することに適していると考えSPH法を用いることとした。本手法ではこのSPH法を用いてリアルタイムに自由表面流れのシミュレーションとして水の挙動を表現した天田ら [16] の手法を参考にガラスのモデル化を行った。以下にSPH法によるガラスのモデル化の具体的な手法について述べる。

2.1.1 カーネル近似による物理量の補間

SPH法は粒子の集合を用いて、個々の粒子の運動により、流体や剛体の運動を表現する手法である。本手法で用いる粒子は速度 \mathbf{v} 、密度 ρ 、圧力 p 、質量 m 、また温度 T の物理量を保持し空間中を運動するものとする。本手法では空間中の点 \mathbf{r} のある物理量 A を以下の式 (2.1) により求める。

$$A(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} W(\mathbf{r} - \mathbf{r}_j, h) \quad (2.1)$$

ここで \mathbf{r}_j は点 \mathbf{r} の周囲に存在する粒子の位置座標であり、関数 W は粒子の間の空間の物理量を補間するカーネル関数を表し、 h はカーネル関数の影響半径を表す。SPH法ではカーネル関数を粒子間の距離による影響力の違いを考慮するため

に用い、粒子同士との距離が離れているほど減少する関数を用い、影響範囲外の粒子の影響を受けない関数を用いる。SPH法では粒子の保持する離散的な物理量の値をカーネル関数を用いることで空間中の滑らかな分布として近似することが可能であり、本手法ではこれをカーネル近似と呼ぶ。

またSPH法には物理量 A の勾配 ∇A やラプラシアン $\nabla^2 A$ を、カーネル関数の勾配やラプラシアンと物理量の積との積分により表すことが可能であるという特徴がある。これにより空間中の点 \mathbf{r} の物理量 $A(\mathbf{r})$ の勾配 $\nabla A(\mathbf{r})$ は以下の式 (2.2) で表す。

$$\nabla A(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla W(\mathbf{r} - \mathbf{r}_j, h) \quad (2.2)$$

また空間中の点 \mathbf{r} の物理量 $A(\mathbf{r})$ のラプラシアン $\nabla^2 A(\mathbf{r})$ は以下の式 (2.3) で表す。

$$\nabla^2 A(\mathbf{r}) = \sum_j m_j \frac{A_j}{\rho_j} \nabla^2 W(\mathbf{r} - \mathbf{r}_j, h) \quad (2.3)$$

これらを用い粒子の保持する離散的な物理量を空間の分布として補間し、粒子の運動に影響する物理量を求めることが可能となる。本手法では求める物理量により異なる3種類のカーネル関数を用いた。

2.1.2 カーネル関数

本手法では Müller ら [17] が提案した3種類のカーネル関数を用いる。カーネル関数は物理量の補間に用いるため、安定でかつ計算量が少ないことを条件とする。以下で挙げるカーネルは全て有効範囲内と外との境界上でそのカーネルの値と傾きが0となっており、有効範囲外の計算を行う必要がないため効率的に計算可能である。以下の式 (2.4)、(2.5)(2.6) はそれぞれ密度と温度変化の計算に用いる poly6 カーネル W_p と表面張力の計算に用いる poly6 カーネルの勾配とラプラシアンである。また $r = |\mathbf{r}|$ である。

$$W_p(\mathbf{r}, h) = \begin{cases} \frac{315}{64\pi h^9}(h^2 - r^2)^3 & (0 \leq r \leq h) \\ 0 & (\text{上記以外}) \end{cases} \quad (2.4)$$

$$\nabla W_p(\mathbf{r}, h) = \begin{cases} \frac{945}{32\pi h^9}(h^2 - r^2)^2 \cdot \mathbf{r} & (0 \leq r \leq h) \\ 0 & (\text{上記以外}) \end{cases} \quad (2.5)$$

$$\nabla^2 W_p(\mathbf{r}, h) = \begin{cases} \frac{945}{32\pi h^9}(r - h)(r + h)(5r^2 - h^2) & (0 \leq r \leq h) \\ 0 & (\text{上記以外}) \end{cases} \quad (2.6)$$

式(2.4)の poly6 カーネルは距離の2乗を用い値を出すため、計算コストの高い平方根の計算の必要がない。

以下の式(2.7)、(2.8)はそれぞれ圧力の計算に用いる Spiky カーネル W_s と実際に用いるその勾配である。

$$W_s(\mathbf{r}, h) = \begin{cases} \frac{15}{\pi h^6}(h - r)^3 & (0 \leq r \leq h) \\ 0 & (\text{上記以外}) \end{cases} \quad (2.7)$$

$$\nabla W_s(\mathbf{r}, h) = \begin{cases} \frac{45}{\pi h^6} \frac{(h-r)^2}{r} \cdot \mathbf{r} & (0 \leq r \leq h) \\ 0 & (\text{上記以外}) \end{cases} \quad (2.8)$$

Spiky カーネルの勾配は粒子間の距離が小さくなるにつれ急激に大きくなる性質をもち、圧力計算に用いることで粒子間が極端に近付くことを防ぐ効果がある。

以下の式(2.9)、(2.10)はそれぞれ粘性力の計算に用いる Viscosity カーネル W_v と実際に用いるラプラシアンである。

$$W_v(\mathbf{r}, h) = \begin{cases} \frac{15}{2\pi h^3} \left(-\frac{r^3}{2h^3} + \frac{r^2}{h^2} + \frac{h}{2r} - 1 \right) & (0 \leq r \leq h) \\ 0 & (\text{上記以外}) \end{cases} \quad (2.9)$$

$$\nabla^2 W_v(\mathbf{r}, h) = \begin{cases} \frac{45}{\pi h^6}(h - r) & (0 \leq r \leq h) \\ 0 & (\text{上記以外}) \end{cases} \quad (2.10)$$

これらのカーネルの有効範囲 h は定数であるため、それぞれのカーネルの係数部分をあらかじめ計算しておくことで全体の処理を軽減することができる。

2.1.3 支配方程式の離散化

粒子の運動を定義する支配方程式として、本手法では以下の2つの式(2.11)(2.12)を用いる。式(2.11)は質量保存則を表す連続の式であり、式(2.12)は流体の運動を表す Navier-Stokes 方程式である。

$$\nabla \cdot \mathbf{v} = 0 \quad (2.11)$$

$$\rho \frac{D\mathbf{v}}{Dt} = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f} \quad (2.12)$$

ここで ρ 、 \mathbf{v} 、 p 、 μ 、 \mathbf{f} はそれぞれ密度、速度ベクトル、圧力、粘性係数、外力を表し、本手法の外力には重力、デバイス操作による力、表面張力を含む。連続の式(2.11)の質量保存則は、本手法においてはシミュレーション中の粒子の質量を一定とし、粒子の数はシミュレーション中は変化しない物として満たした。Navier-Stokes 方程式(2.12)内の $\frac{D\mathbf{v}}{Dt}$ は流体要素の速度変化を表し、速度の時間微分となる。右辺の $-\nabla p$ 、 $\mu \nabla^2 \mathbf{v}$ 、 \mathbf{f} はそれぞれ圧力項、粘性項、外力項と呼び、SPH 法では圧力項、粘性項、外力項を一度に解くのではなく、それぞれについての力を解き、最終的にそれらの力の和をとり加速度を求める。式(2.1)、(2.2)、(2.3)を用い Müller らが提案した方法をもとに式(2.12)を離散化する。以下で密度計算、圧力項、粘性項の離散化について述べる。

2.1.4 密度計算

SPH 法の計算ではまず粒子の密度を粒子の質量を用いてカーネル近似で求める。式(2.1)を用い空間中の点 \mathbf{r} の密度 $\rho(\mathbf{r})$ は以下の式(2.13)で表す。

$$\begin{aligned} \rho(\mathbf{r}) &= \sum_j m_j \frac{\rho_j}{\rho_j} W_p(\mathbf{r} - \mathbf{r}_j, h) \\ &= \sum_j m_j W_p(\mathbf{r} - \mathbf{r}_j, h) \end{aligned} \quad (2.13)$$

ここで m_j 、 ρ_j はそれぞれ、周囲に存在する粒子 j の質量、周囲に存在する粒子 j の密度である。密度はシミュレーションのステップごとに粒子の位置で更新し、粒子間で作用する力の計算に用いる。

2.1.5 圧力項の離散化

粒子間に働く圧力の影響を示す圧力項 $-\nabla p$ は Müller らの手法により以下の式 (2.14) ように表す。

$$\mathbf{f}_i^p = -\nabla p = -\sum_j m_j \frac{p_j + p_i}{2\rho_j} \nabla W_s(\mathbf{r}_i - \mathbf{r}_j, h) \quad (2.14)$$

圧力項の計算時に必要となる圧力 p は、Desbrun ら [18] が提案した手法を用い以下の式 (2.15) より求める。

$$p = k_p(\rho - \rho_0) \quad (2.15)$$

k_p 、 ρ_0 はそれぞれ粒子の位置における圧力の大きさを調節するパラメータと初期粒子配置における密度である。この式により粒子の密度が ρ_0 になるような方向へ圧力が働く。

2.1.6 粘性項の離散化

粘性とは流体内の速度を一様にする方向に働く力のことであり、本手法では周囲の粒子との速度の差分の総和をとることで考慮する。粒子間に働く粘性力を示す粘性項 $\mu \nabla^2 \mathbf{v}$ は Müller らの手法を参考にし、以下の式 (2.16) で表す。

$$\mathbf{f}_i^v = \mu \nabla^2 \mathbf{v} = \sum_j \mu_j m_j \frac{\mathbf{v}_j - \mathbf{v}_i}{\rho_j} \nabla^2 W_v(\mathbf{r}_i - \mathbf{r}_j, h) \quad (2.16)$$

式 (2.16) における μ_j は粒子 j の粘性係数を表す。本手法では μ_j をガラスの温度による粘性変化の性質を表現するため個々の粒子に温度 T を与え、温度に依存す

るものとして定義する。ガラスの温度変化と粘性の関係は VTF(Vogel-Tamman-Fulcher) 式 [19] により表すことができる。式 (2.17) は VTF 式を基にした本手法で用いる温度と粘性の関係式である。

$$\log \mu_j = A + \frac{B}{T_j - T_0} \quad (2.17)$$

式中の A 、 B 、 T_0 は物体の性質により決定する定数であり、 T_j は粒子 j の温度である。ガラスの粘性係数と温度の実測値を用いることで A 、 B 、 T_0 を決定し、ガラスの温度による粘性変化を近似する。本手法では A に -2.09、 B に 7071、 T_0 に 91 を用いた。図 (2.1) は本手法で用いた温度と粘性の関係を表すグラフである。

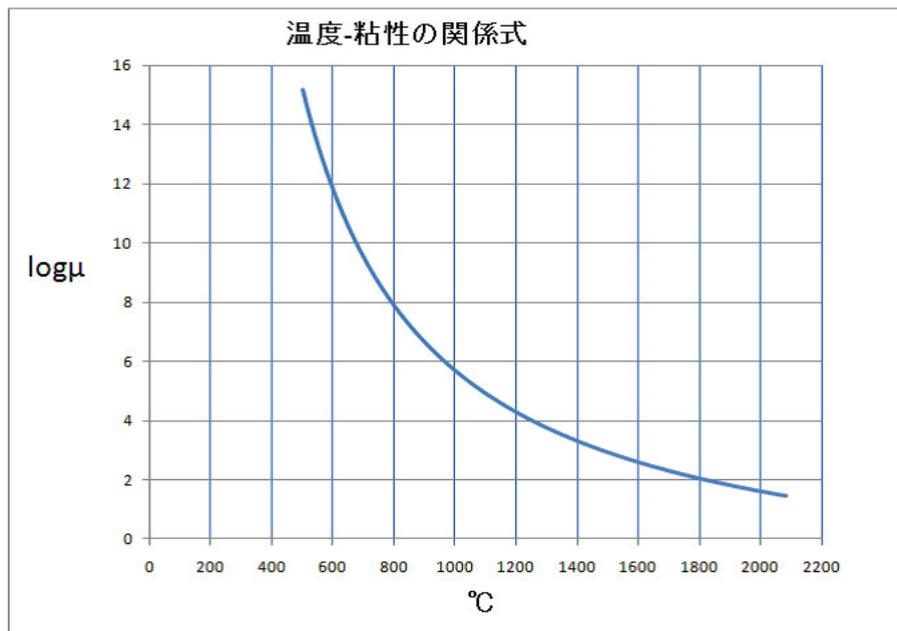


図 2.1: 本手法における温度と粘性の関係

2.1.7 粒子の温度変化

本手法ではガラス細工の粘性変化の性質を表現するため、粒子の温度変化の条件を以下のように定義する。

- 隣接する粒子間での熱移動
- 外気による温度低下
- バーナーの炎による加熱

隣接する粒子との熱移動は、小田ら [20] の手法を参考にし、ある粒子 i の温度変化量 ΔT_i を以下の式 (2.18) により表現する。

$$\Delta T_i = \sum_j \lambda(T_j - T_i)W_p(\mathbf{r}_i - \mathbf{r}_j, h) \quad (2.18)$$

T_i 、 T_j 、 λ はそれぞれある粒子 i の温度、粒子 i の近傍粒子の温度、熱伝導率を表す。本手法では熱伝導率として 0.0002 を使用した。以上により隣接する粒子間の熱移動を行う。

次に外気による温度低下について考慮する。本手法では計算量を少なくするために空気を粒子として考慮していない、そのため式 (2.18) を用いることはできない。そこで外気による温度低下は以下のように表現する。まず表面付近に存在する粒子は空気に触れているものとし、表面付近の粒子に対し温度変化処理を行う。表面付近の粒子を求める。表面付近に存在する粒子は物体の外部と接する位置に存在する以上、表面付近に存在する粒子の周囲に存在する粒子の数は物体内部の粒子と比べ少なくなっている。これは表面付近の粒子の密度と物体内部の密度に差が生じることを示している。そのため粒子の密度が低い粒子を表面に接している粒子として求めることができる。本手法では密度が初期配置粒子密度 ρ_0 の 0.98 倍以下の粒子は表面付近の粒子とした。表面付近の粒子 s の温度変化量 ΔT_s は以下の式 (2.19) により求める。

$$\Delta T_s = \alpha(T_r - T_s) \quad (2.19)$$

T_s 、 T_r 、 α はそれぞれ、表面付近のある粒子 s の温度、外気温、熱伝達係数を表す。本手法では外気温 T_r は一定として扱い 40 °C とした、また熱伝達係数 α は粒

子から空気への熱の伝わり易さを定義する係数であり、本手法では 0.0001 を用いた。これにより空気への放熱を表現する。

バーナーの炎による加熱は、後述のデバイス操作によって加熱領域を選択し、加熱領域内に存在する粒子かつ表面付近の粒子に対し温度上昇を行う。バーナーによる温度上昇は以下の式 (2.20) により行う。

$$\Delta T_s = \alpha_b (T_b - T_s) \left(1.0 - \frac{r_t}{r_e} \right) \quad (2.20)$$

ここで T_s はバーナーの加熱領域内の表面付近の粒子 s の温度であり、 T_b はバーナーの炎の温度を表し本手法では 1500 °C とした。 α_b はバーナーの炎からガラスへの熱の伝わりやすさを定義する係数であり、本手法では 0.001 を用いた。また r_t は操作点と粒子との距離を表し、 r_e はバーナーの影響半径を表す。これによりバーナーの炎の中心部に近づくほど熱が高くなるという特徴を表現している。

上記のようにして、粒子の温度変化を表現した。

2.1.8 表面張力

流体が気体と接している表面をもつ状態では、流体の運動は表面張力の影響を受ける。表面張力とは、流体の表面積をできる限り小さくする方向へ働く力であり、流体は表面積が最小である球形に近づく力が働く。ガラス細工には引き伸ばし変形の際、曳糸性により細長く伸びる変形が起こるが、その際に変形中のガラスの中心部分がくびれるという特徴が確認できる。このくびれを起こす要素として、流体の表面張力が関係していると考えられることができる。ガラス細工において、このくびれ形状による突起の形成は鋭利な突起を生成するために必要な要素であるといえる。そのため本手法では表面張力を考慮する。本手法では表面張力を考慮するため、Morris ら [21] の手法を用いた。前節で求めた表面付近の粒子に対し以下で求める表面張力を与えることで表現する。まず Morris らの手法では式 (2.21) により空間中の値 $C(\mathbf{r})$ を求めその値を粒子に保持する。

$$C(\mathbf{r}) = \sum_j m_j \frac{1}{\rho_j} W_p(\mathbf{r} - \mathbf{r}_j, h) \quad (2.21)$$

次に 2.1.7 節と同様に粒子の密度が初期配置粒子密度より密度が低い粒子を検出し、表面付近の粒子とする。表面張力は流体の表面積をできる限り小さくする方向へ働く力であり、その力の向かう方向は流体の内部に向かい、流体の表面に対して垂直な物となる。そこで表面付近の粒子では式 (2.22) を用いて C 値の勾配を求める。ここで求めた C 値の勾配は同時に流体表面のある点での表面に対して垂直な法線ベクトルを表す。ある表面の付近の粒子における法線ベクトル \mathbf{n} は式 (2.22) のように表す。

$$\mathbf{n}(\mathbf{r}) = \nabla C(\mathbf{r}) = \sum_j m_j \frac{C_j}{\rho_j} \nabla W_p(\mathbf{r} - \mathbf{r}_j, h) \quad (2.22)$$

ここで C_j は空間中のある点 \mathbf{r} の周囲に存在する粒子 j の保持する、式 (2.21) により定義した C の値である。また、表面張力の強さには表面の曲率が影響する。曲率を得るためには、まず表面付近の粒子において式 (2.23) により C のラプラシアン $\nabla^2 C$ を求める。そして求めた $\nabla^2 C$ を用い式 (2.24) により曲率 k を得る。

$$\nabla^2 C(\mathbf{r}) = \sum_j m_j \frac{C_j}{\rho_j} \nabla W_p^2(\mathbf{r} - \mathbf{r}_j, h) \quad (2.23)$$

$$k(\mathbf{r}) = \frac{-\nabla^2 C(\mathbf{r})}{|\mathbf{n}(\mathbf{r})|} \quad (2.24)$$

これらを用いてある表面付近の粒子にかかる表面張力を \mathbf{f}_s と定義すると、表面張力 \mathbf{f}_s は以下のように表現できる。

$$\mathbf{f}_s = \sigma k(\mathbf{r}) \mathbf{n}(\mathbf{r}) = -\sigma \nabla^2 C(\mathbf{r}) \frac{\mathbf{n}(\mathbf{r})}{|\mathbf{n}(\mathbf{r})|} \quad (2.25)$$

ここで σ は表面張力の強さを表す係数である。

以上のようにして表面張力 \mathbf{f}_s を得る。検出した表面付近の粒子の作用力計算式(2.12)の外力項に求めた表面張力 \mathbf{f}_s を加算することで、粒子の運動に表面張力を考慮することができる。

2.2 速度と座標の更新

以上のようにして求めた力を用い、粒子の速度を以下のように更新する。

$$\mathbf{v}_i(t + \Delta t) = \mathbf{v}_i(t) + \frac{\Delta t}{\rho_i}(\mathbf{f}_i^p + \mathbf{f}_i^v + \mathbf{f}_i) \quad (2.26)$$

ここで $\mathbf{v}_i(t)$ 、 \mathbf{f}_i^p 、 \mathbf{f}_i^v 、 \mathbf{f}_i 、 Δt はそれぞれ時刻 t における粒子 i の速度、粒子 i の圧力、粒子 i の粘性力、粒子 i の外力、タイムステップを表す。式(2.26)により求めた時刻 $t + \Delta t$ における粒子 i の速度 $\mathbf{v}_i(t + \Delta t)$ を用いて、時刻 $t + \Delta t$ における粒子 i の座標 $\mathbf{x}_i(t + \Delta t)$ を以下で求める。

$$\mathbf{x}_i(t + \Delta t) = \mathbf{x}_i(t) + \mathbf{v}_i(t + \Delta t)\Delta t \quad (2.27)$$

このように1タイムステップにおける計算を行い、粒子に働く力を求め、次のステップの速度を計算し、粒子の座標を更新する、以上の処理を繰り返すことで粒子の運動を表現する。

2.3 近傍粒子探索

本手法で用いる粒子法のモデルは、粒子間に働く力の計算を行う際に何の工夫も行わない場合、粒子ごとにその他全ての粒子の影響範囲との内外判定を行う必要がある。この処理は粒子の数が増えるにつれ計算量は指数関数的に増加するため、多数の粒子を扱いながら対話的な処理を目的とする本手法には不向きである。そ

ここで本手法では、空間をボクセル格子状に分割し、粒子が影響し合う可能性のある格子領域内の粒子間でのみ内外判定を行う近傍探索を行うことで処理の高速化を図る。まず、本手法における全粒子はある限定した空間内でのみ移動可能とする。次に全粒子の存在する空間を、一辺を粒子の影響半径と等しくする格子領域に等間隔に分割する。本手法では一辺の長さを粒子の影響半径と等しくする $32 \times 32 \times 32$ 個のボクセル空間を近傍探索の格子領域とし、粒子を粒子が存在する位置座標に対応するボクセルに割り当てる。本手法では原田ら [22] のスライスデータ構造を参考にしボクセルに粒子を割り当てた。本手法で用いたスライスデータ構造は上で定義した近傍探索の格子領域をボクセルの y 軸に直行するスライス状を分割し、ある一つのスライスにはボクセルの Y 軸方向にボクセルを 1 つ持つとし、 XZ 軸方向に広がりをもつ 2 次元の格子とする。スライス分解後、それぞれのスライスにおいて XZ 軸方向のバウンディングボックスを定義する。 XZ 軸方向における最大と最小のボクセル開始座標を計算し、 X 軸方向においては X_i^b 、 X_i^s をそれぞれ最大、最小の開始座標とし、 Z 軸方向においては Z_i^b 、 Z_i^s とする。またここで i はスライスの番号を表す。これを用いてスライス i 上の X 軸方向のボクセル数 n_i^x 、 Z 軸方向のボクセル数 n_i^z はそれぞれ式 (2.28)、(2.29) により求まる。

$$n_i^x = \frac{X_i^b - X_i^s}{d} + 1 \quad (2.28)$$

$$n_i^z = \frac{Z_i^b - Z_i^s}{d} + 1 \quad (2.29)$$

ここで d はボクセル一辺の長さを表す。これらを用い、スライス i 上のボクセル数 n_i を式 (2.30) により求める。

$$n_i = n_i^x n_i^z \quad (2.30)$$

本手法ではスライスのバウンディングボックス内のボクセルのみボクセルを確保する。ここで計算領域を n 個のスライス分解するとして、あるスライスを S_i と

表す。スライス S_i での先頭ボクセル番号 b_i はスライス S_0 から $S_i - 1$ にあるボクセル数の和として、式 (2.31) により求まる。

$$b_i = \sum_{j < i} n_j \quad (2.31)$$

ここである点 x, y, z が存在するボクセルの番号を計算するため、Y 軸方向の最小ボクセルの開始座標 Y_s を用いて、その点を含むスライス番号 i を式 (2.32) により求める。

$$i = \left(\frac{y - Y_s}{d} \right) \quad (2.32)$$

そしてボクセル番号 $v(x, y, z)$ は式 (2.33) により求まる。

$$v(x, y, z) = b_i + \left(\left(\frac{x - X_i^s}{d} \right) + \left(\frac{z - Z_i^s}{d} \right) n_i^x \right) \quad (2.33)$$

式 (2.33) よりある粒子の位置座標から属するボクセルの番号を求め、近傍格子に割り当てることができる。このときある粒子と影響し合う可能性のある粒子は、粒子自身が属するボクセルとその周囲に隣接しているボクセルに割り当てた粒子のみとなる。したがってある粒子の力の計算を行う可能性のある粒子は、ある粒子が属するボクセルを中心とした 27 ボクセルに属する粒子のみとなり、その他のボクセルに割り当てた粒子と影響し合う可能性はない。そのためその他のボクセルに属する粒子との距離の判定処理を省略可能である。これにより粒子間に働く力の計算処理を効率化する。

2.4 ボリュームレンダリングによる表面の描画

本手法では粒子モデルからガラスの滑らかな表面形状を可視化するため、粒子からの距離に応じ濃度値が減少する距離場関数を用いてボリュームデータを生成し、

その等値面をガラス表面として可視化する。まず、全ての粒子の中心位置から空間中のある点 \mathbf{r} に対し以下の式 (2.34) により濃度値 $D(\mathbf{r})$ を与える。ここで $r = |\mathbf{r}|$ である。

$$D(\mathbf{r}) = \begin{cases} D_0 \left(1 - \left(\frac{r}{h}\right)^2\right)^2 & (0 \leq r \leq h) \\ 0 & (h < r) \end{cases} \quad (2.34)$$

このとき D_0, r, h はそれぞれ粒子中心での濃度値、粒子中心からの距離、距離場関数の影響範囲である。次に、距離場関数が存在する範囲を覆う形でボクセルに分割する。本手法では $128 \times 128 \times 128$ 個のボクセルに分割した。ここでボクセルの中心位置での濃度をボクセルの濃度値とすることでボリュームデータを生成する。

粒子の位置座標から生成したボリュームデータを可視化するため本手法では、ボリュームレンダリング [23] による可視化を行う。ボリュームレンダリングは、空間の密度等を表すボリュームデータを直接表示する手法であり、本手法では式 (2.34) により求めた濃度値のボリュームデータよりガラス形状の表面を可視化し、式 (2.37) より求めた温度値のボリュームデータより、加熱部分の赤熱現象を可視化する。一般的なボリュームレンダリングは視点からスクリーン上のピクセルに視線を伸ばしその視線にあるボクセルの濃度値を合計することによりピクセルの描画色を決定する手法である。これをスクリーン上にピクセル解像度分の視線を伸ばし対応するピクセルの描画色を決定することで、内部情報を反映したボリュームデータの可視化を行うことが可能である。以下で本手法で用いたボリュームレンダリングを述べていく。まずスクリーン上のピクセルに対し視点位置から視線ベクトルをピクセル解像度分求める。次に視点からスクリーン上の点の間を等間隔に区切った視線ベクトルのサンプリング点においてのボクセルの値を参照する。このサンプリング点は必ずしもボクセルの中心を通過するわけではない。そのためボクセルの値を求めたいサンプリング点の周囲に存在するボクセルの濃度値を参照し、線形補間を行う。これによりサンプリング点においての濃度値を求める。本手法ではボリュームデータを GPU のテクスチャメモリ上に展開しテクスチャユニットの

補間機能を用いることで、視線ベクトル上のサンプリング点での補間値を求めた。ここで求めた濃度補間値を表面を定義する閾値と比較し、表面の検出を行う。表面を検出した際にはガラス表面での光の反射を表現するため、法線を求める。サンプリング点 $\mathbf{B} = (B_x, B_y, B_z)$ 上の法線 $\mathbf{N}(B_x, B_y, B_z)$ はボクセル通過点に隣接するボクセルの濃度値を参照することで求めることができ、本手法では中心差分法 [24] により法線を求める。式 (2.35), (2.36) は本手法で用いた法線 $\mathbf{N}(B_x, B_y, B_z)$ を求める式である。

$$\mathbf{N}(B_x, B_y, B_z) = \frac{\nabla D(B_x, B_y, B_z)}{|\nabla D(B_x, B_y, B_z)|} \quad (2.35)$$

$$\nabla D(B_x, B_y, B_z) = \left(\begin{array}{c} \frac{D(B_{x+1}, B_y, B_z) - D(B_{x-1}, B_y, B_z)}{2}, \\ \frac{D(B_x, B_{y+1}, B_z) - D(B_x, B_{y-1}, B_z)}{2}, \\ \frac{D(B_x, B_y, B_{z+1}) - D(B_x, B_y, B_{z-1})}{2} \end{array} \right) \quad (2.36)$$

ここで $D(B_x, B_y, B_z)$, $\nabla D(B_x, B_y, B_z)$ はそれぞれサンプリング点上の補間濃度値、サンプリング点上の補間濃度値の勾配ベクトルを表す。これにより求めた法線を用い、鏡面反射光計算と拡散反射光計算を行い、光の影響を考慮した色値を対応するピクセルの描画色に加算する。上記の処理を視線ベクトルがボクセルデータを通るまで繰り返しピクセル解像度分の描画色を決定する。これにより表面部分のみの可視化を行いガラスの透明な質感を表現した。また本手法では体験システム用いるため対話的な速度で操作可能である必要があるといえる。しかし、ボリュームデータの生成と視線計算の処理が高コストな処理であるためボリュームレンダリングを毎ステップ行うことで速度低下が起こる。そこで本手法ではボリュームレンダリング処理を数ステップ間隔で行うことを可能にしている。

2.5 ガラスの赤熱表現

粒子の温度値に対してボリュームデータを生成し、ガラスの赤熱表現に用いる。温度値のボリュームデータ $T(r)$ の生成には式 (2.37) を用いた。

$$T(\mathbf{r}) = \begin{cases} T_j \left(1 - \left(\frac{r}{h}\right)^2\right)^2 & (0 \leq r \leq h) \\ 0 & (h < r) \end{cases} \quad (2.37)$$

ここで T_j はある地点 \mathbf{r} の周囲存在する粒子の温度値である。

式 (2.37) により生成した温度のボリュームデータを用い、ボリュームレンダリングにより高温時のガラスの特徴である温度による色変化を考慮した可視化を行う。まず 2.4 節と同様にして、視線ベクトルとボクセルの通過点を求め、温度の補間値を得る。ガラスの赤熱現象はガラス内部においても発生するため表面の検出は行わず、温度補間値を得た時点でピクセルの描画色計算を行う。ガラスの赤熱現象は約 800 °C あたりから赤く変わり始め、1000 °C を超えたあたりから黄色みがかかったオレンジ色になり 1500 °C では黄色がかかった白色にその温度の変化にしたがって変色するという特徴がある。本手法では色と温度の関係を表すことが可能なプランクの公式を用いボクセル通過点での色情報 $B_c = (R, G, B)$ を計算した。式 (2.38) はプランクの公式 [25] を用いた色情報計算式である。

$$\begin{aligned} R_B &= \left(\frac{8\pi hc}{R_p^5} \left(\exp\left(\frac{hc}{kT_k(\mathbf{B}) R_p}\right) - 1 \right) \right) q \\ G_B &= \left(\frac{8\pi hc}{G_p^5} \left(\exp\left(\frac{hc}{kT_k(\mathbf{B}) G_p}\right) - 1 \right) \right) q \\ B_B &= \left(\frac{8\pi hc}{B_p^5} \left(\exp\left(\frac{hc}{kT_k(\mathbf{B}) B_p}\right) - 1 \right) \right) q \end{aligned} \quad (2.38)$$

ここで h, k, c は定数値であり、 $h = 6.6260755E - 34$, $k = 1.380658E - 23$, $c = 2.99792458E + 08$ である。また R_p, G_p, B_p はそれぞれ可視光線の赤色波長、緑色

波長、青色波長を表す。またここで $T_k(\mathbf{B})$ はボクセル通過点での温度補間値の絶対温度 (K) を表し、ここで求めた色情報 B_c を視線ベクトルが通過したボクセル数分加算することで、ピクセルの描画色を決定する。また q は本手法で用いた描画色の濃度を調節する係数であり、本手法では $q = 0.0005$ とした。このようにしてガラスの温度変化による色変化の特徴を考慮した赤熱現象を表現した。ここで求めた表面の描画と赤熱表現の描画を足し合わせることで最終的な描画としガラス粒子の可視化を行った。

2.6 くびれ補正による突起形状の生成

本手法ではバーナーワークの引き伸ばし変形を表現するため3次元入力デバイスをを用い3次元位置を取得し、仮想空間中でのデバイス位置をデバイス操作点 M とする。任意の位置にデバイス操作点を移動し、変形を行いたい場所を選択した際にデバイス操作点から半径 M_r の範囲を変形影響範囲としバーナーワークのピンセットで掴む範囲を表す。その際に操作点の影響範囲内に存在する十分な加熱状態の粒子を変形中の粒子 m と定義する。変形操作中は全ての粒子 m はデバイス操作点に追従して移動するとする。ここで引き伸ばし変形行うためデバイス操作点を移動することで粒子を移動しガラス形状に変形を行うことが可能となるが、本手法の粒子法ではデバイス操作の早さに対して粒子同士の影響範囲が非常に小さいため、デバイス操作により粒子を移動した場合粘性力による影響を受けるより早く、変形中の粒子が粒子間の影響範囲を抜け出てしまうという問題が生じた。ガラスの突起形状は変形する部分に周囲のガラスが粘性力によって伸びることで形成する。そのためこの変形手法ではガラス細工のようなくびれを伴う突起を形成することができない。そこで本手法では粘性力の代わりに強制的に変形時に周囲に存在する粒子を移動させる引力を導入することでくびれを伴う突起を形成を行う。これを本手法ではくびれ補正処理と呼ぶ。

くびれ補正処理ではデバイス操作点から半径 M_r 以内の粒子に存在する粒子を変形中の粒子と定義する。またデバイス操作点から半径 M_r より遠く、半径 M_r の

2 倍の距離以内の領域を変形隣接範囲とする。そして変形隣接範囲内の粒子を変形隣接粒子 n とする。デバイス操作により変形中の粒子は追従して移動する。このとき変形を開始した地点を記憶しその地点を変形開始点 S とする。ここで変形開始点から半径 M_r の 2 倍の距離以内の領域を変形開始点の影響範囲とする。この変形開始点は一度の引き伸ばし変形が継続している間は、移動せず同じ位置に存在し続ける。変形中の粒子がデバイス操作に追従して移動した後はガラス形状内に空洞が生じてしまう。これを防ぐため、変形開始点の影響範囲内の全ての変形隣接粒子は変形開始点への引力を受けることとした。また変形隣接範囲に存在する変形隣接粒子は変形開始点への引力と共にデバイス操作点への引力を受ける。これにより引き伸ばし方向へ伸びる様子を表現する。さらに変形隣接範囲に存在する変形中の粒子はデバイス操作点に追従して移動するとともに変形開始点への引力を受ける。また引力を受けた変形中の粒子がデバイス操作点の影響範囲から出た場合、その粒子は変形隣接粒子に変化しその後デバイス操作には追従しなくなる。式 (2.39) は本手法で用いた引力 \mathbf{f}_x を求める式である。

$$\mathbf{f}_x = \left(\frac{\mathbf{p}_t - \mathbf{p}_n}{|\mathbf{p}_t - \mathbf{p}_n|} \right) k_p \quad (2.39)$$

ここで $\mathbf{p}_t, \mathbf{p}_n, k_p$ はそれぞれ引力の向かう点の位置座標、引力の働く粒子の位置座標、本手法で定めた引力の強さを定義する係数である。これにより変形操作中は粒子に対しそれぞれの状況に応じた変形操作点と変形開始点の方向へ移動する力が加わる。図 (2.2)(2.3)(2.4)(2.5) はくびれ補正処理を用いた引き伸ばし変形時の粒子の動きを 2 次元で表したものである。

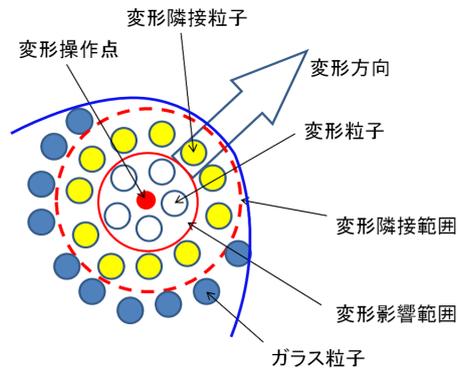


図 2.2: くびれ補正処理を用いた引き伸ばし変形-形状変形前

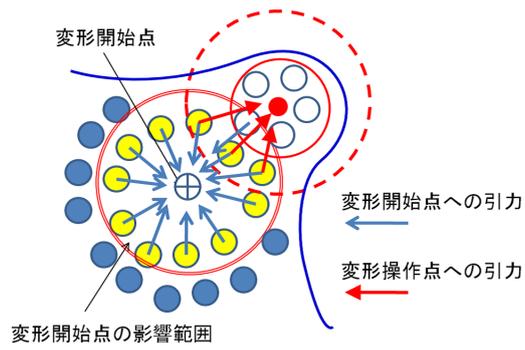


図 2.3: くびれ補正処理を用いた引き伸ばし変形-操作点移動開始

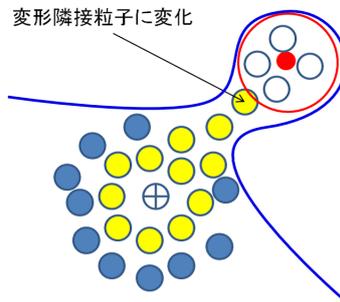


図 2.4: くびれ補正処理を用いた引き伸ばし変形-突起形成後

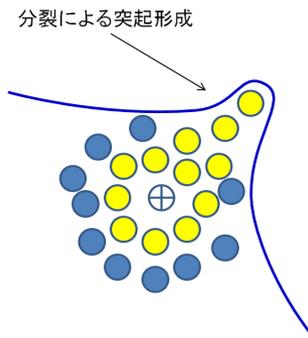


図 2.5: くびれ補正処理を用いた引き伸ばし変形-分裂後

2.7 3次元入力デバイスによる操作

本手法ではガラス細工の変形操作を表現するため、3次元の位置入力可能なデバイスを用いる。これによりバーナーワークの引き伸ばし操作と、バーナーによる加熱操作を表現する。なお本手法では引き伸ばし操作と加熱操作を同一デバイスを用いて行うため、利用者はこの2つの操作をシミュレーション中任意に切り替えることを可能とする。

バーナーワークの引き伸ばし変形ではピンセットによりつまみ引き伸ばす操作

を行う。またバーナーワークのくびれを伴う突起形成を考慮するため、引き伸ばし変形時に 2.6 節で述べたくびれ補正処理を行うことでその形状を表現する。

バーナーワークの加熱操作は引き伸ばし操作と同様に操作点を表示し、加熱操作を行う任意の点を選択する。その際、本手法では操作点より半径 5.0 の領域を加熱領域と定義する。加熱領域内かつ表面付近の粒子に対し式 (2.20) により温度上昇処理を行う。これによりバーナーワークの加熱操作を表現する。

2.8 計算手順

本手法の計算手順を以下で述べる。

まず、ガラス形状表現する粒子を初期位置へ配置する。次に全ての粒子に対し 2.3 節で述べた近傍粒子探索を行い、全ての粒子を近傍格子に割り当てる。割り当てたのち全粒子の近傍粒子を自身が属する近傍ボクセルから探索し、式 (2.13) を行い粒子の現在位置での密度を更新する。また 2.1.7 節で述べた温度計算を行い温度を更新する。その後 2.7 節で述べたデバイスによる操作により、速度を外力に加算、バーナーによる加熱処理を行う。次に密度を用いて式 (2.14) により圧力計算、更新した温度を用いて式 (2.16) により粘性項を計算する。また 2.1.8 節で述べた、表面張力を外力に加算し、圧力項と粘性項と外力項を合計し粒子の速度を求める。求めた速度に従い 2.2 節で述べたように粒子座標を更新する。粒子の移動が終わった時点で 2.4 節と 2.5 節で述べた濃度値と温度値に関するボリュームデータを生成する。描画用のボリュームデータを用いボリュームレンダリング処理を行い画像を生成し最終的に出力する。そして近傍粒子探索からここまでの一連の処理をシミュレーション中繰り返し続けることで、ガラス細工体験システムを実現した。以下のチャートは本手法の計算手順を表したものである。

1. 粒子をそれぞれの初期位置へ配置
2. 全ての粒子に対し 2.3 節で述べた近傍粒子探索を行う

3. 全ての粒子に対し近傍粒子間で式 (2.13) により密度、2.1.7 節で述べた温度を計算
4. 全ての粒子に対し 2.7 節で述べたデバイス操作による粒子の移動と温度変化を与える
5. 全ての粒子に対し式 (2.14) により圧力項と式 (2.16) により粘性項と外力による影響を計算し速度を求める
6. 全ての粒子の位置を 2.2 節で述べたように速度に従い更新
7. 全ての粒子の位置から 2.4 節で述べた描画用のボリュームデータを生成
8. 描画用のボリュームデータを用い 2.4 節と 2.5 節で述べたボリュームレンダリングを行い画像を生成
9. ボリュームレンダリングによって生成した画像を表示
10. 2~9 を繰り返す

2.9 GPGPU による高速化

本研究が目的とする体験システムでは、対話的な操作が可能である必要がある。そのため対話的に操作可能な実行速度を得るため、GPU(Graphics Processing Unit)を汎用計算に用いる GPGPU を用いることで高速化を図った。本手法では GPGPU のプログラミング環境として Nvidia 社 [26] の CUDA[27] を使用した。本手法の粒子法モデルは粒子間での逐一的な処理を必要としないため個々の粒子についての処理を独立して行うことが可能であり、GPU 内の計算ユニット毎に 1 粒子の処理を割り当てることで並列化し、粒子にかかる作用力の計算を行い粒子移動処理を高速化した。粒子法に用いる近傍粒子探索の GPGPU 実装については原田らの手法を参考にし、近傍ボクセルへの粒子の割り当て処理を並列化した。またボリュームレンダリングに関しては粒子位置からボリュームデータの生成にボリュームデー

タを構成するボクセル毎に計算ユニットを割り当てることで並列化した。画面の画素毎に計算ユニットを割り当て視線計算を行い描画色計算を行った。また2.4節で述べた、濃度値の補間処理にはCUDAでサポートしているテクスチャメモリ機能を使用することで、高速な濃度値の補間を行った。以上のようにしてシステム全体の計算を並列化し高速化を行った。

2.10 本システムの概要

本手法では3次元入力機能を持つデバイス「Novint Falcon」[28](以下Falcon)により3次元座標を取得し2.7節で述べた仮想空間上の操作点と加熱点の位置を移動し、粒子モデルに対する変形操作と加熱操作をFalconを用いて行う。バーナーワークを再現するにあたり右手にマウス、左手にFalconを持ち、マウス側では視点移動を行いfalcon側ではバーナーワークで用いるピンセットの動作と対応させる変形操作とガラスに対して加熱を行う加熱操作の2種類の基本操作を行う。以上からなるシステムを提案し、ガラス細工体験システムの実装を行った。

図(2.6)は本システムを用いた形状変形の様子である。

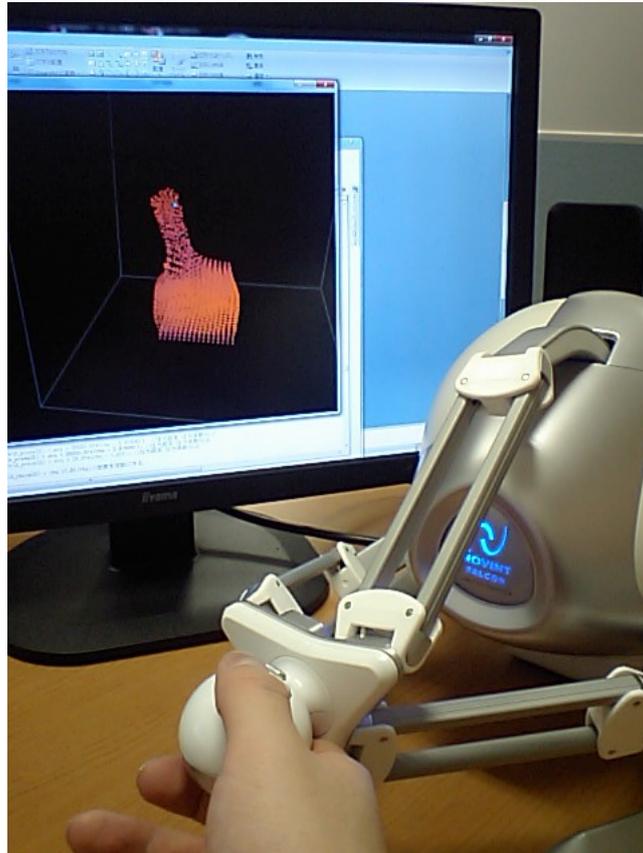


図 2.6: 本システムを用いた形状変形の様子

第 3 章

検証と考察

本章では、2章で述べた手法によるガラス細工体験システムの実装を行い、検証により本手法の有効性を考察する。実装にはグラフィックツールキットであるFKSystem[29]、GPU用のグラフィックス言語にnVidia社のCg言語とCUDAを使用した。実験を行った計算機の環境は表1の通りである。

表1 実行環境

OS	Windows7 64bit
CPU	AMD PhenomII X6 1090T 3.2GHz
GPU	NVIDIA GeForce GTX470
ビデオメモリ	1280MB
メインメモリ	4GB

今回の検証では、一辺を16個とする立方体形状に粒子を初期配置し、合計4096個の粒子の集合によりガラス形状を表現する。以降の検証はすべて上記の初期配置により行っている。

3.1 速度検証

本システムが体験システムとして必要な対話的な速度で実行可能であるか、検証を行う。表2は本システムの実行速度を一秒間に実行可能なステップ回数として表し、本システムの描画状態の違いによる結果を比較したものである。

表2 実行速度

描画の状態	実行速度(ステップ数/秒)
粒子のみの描画	163
ボリュームレンダリング	6
ボリュームレンダリング(3ステップ毎に1度)	18
ボリュームレンダリング(5ステップ毎に1度)	27

粒子の位置座標にポイントスプライトを表示することのみにより粒子を可視化した場合の実行速度は、平均163(ステップ数/秒)であり十分対話的な速度を得ているが、本手法で用いたボリュームレンダリングを有効にした場合の実行速度は平均6(ステップ数/秒)ほどに低下した。これは表面抽出のためのボリュームデータの生成と温度分布のボリュームデータの生成が高コストな処理であることと、視

線上のボリュームをサンプリングする計算を逐一的に行っていることにより、このような速度低下が発生していると考えられる。これに対して本手法ではシミュレーションが数ステップ進む毎に一度だけボリュームレンダリングを行うことで、実行速度の向上を図った、今回の検証では3ステップに一度のボリュームレンダリングを行った場合18(ステップ数/秒)、また5ステップに一度のボリュームレンダリングを行った場合27(ステップ数/秒)という結果を得た。ここの状態での実質のFPS程になるため画面の描画について考えると描画間隔を大きくすることは変形操作による粒子の急な移動が画面に反映されるまで若干の遅延が発生し対話的な操作が可能になっているとは言えない。このようにボリュームレンダリングによって速度が低下する原因としては、高コストはボリュームデータ生成を温度と濃度において2種類で行っていることが挙げることができる。粒子の可視化に関する研究においてボリュームデータの生成を高速に行うような研究 [30] が存在し、今後そのような技術を用いることで高速化を図ることが可能であると言える。

3.2 引き伸ばし変形による突起形成の検証

引き伸ばし変形を行った際の分裂と突起の形成を確認する。図(3.1)(3.2)(3.3)(3.4)は本手法における形状変形の様子である。

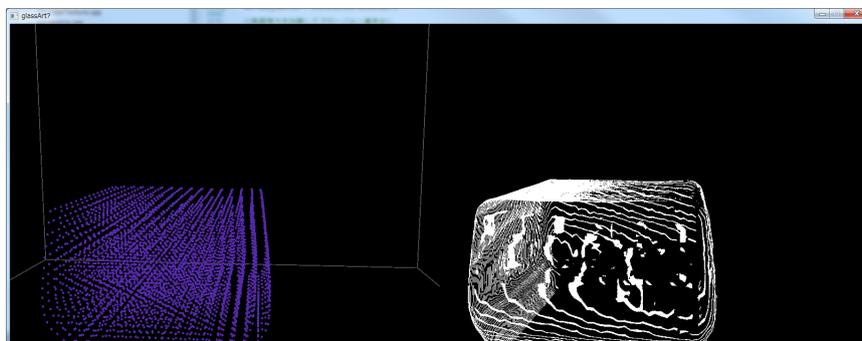


図 3.1: 本システムを用いた形状変形の様子 1

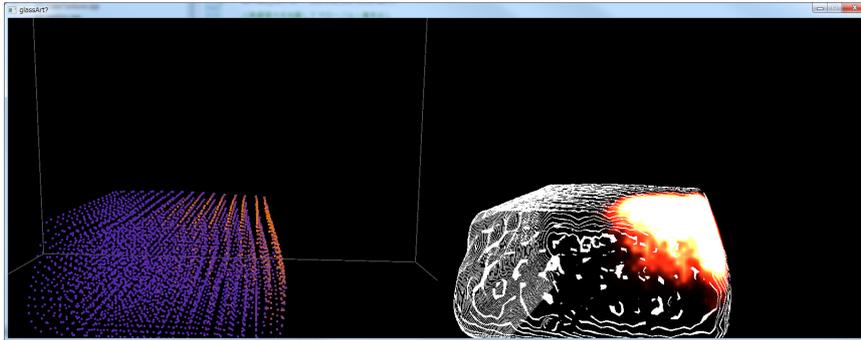


図 3.2: 本システムを用いた形状変形の様子 2

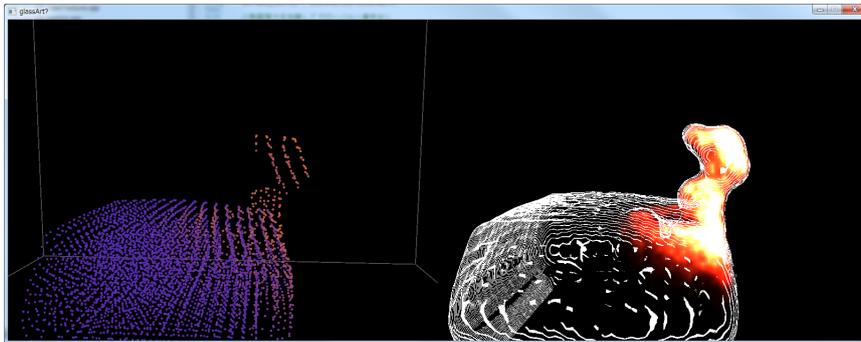


図 3.3: 本システムを用いた形状変形の様子 3

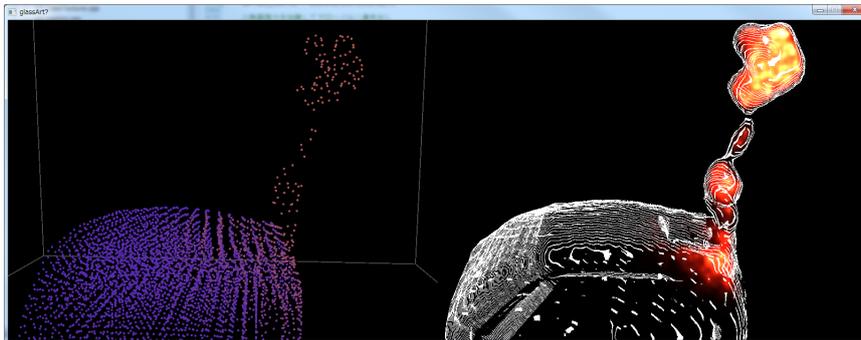


図 3.4: 本システムを用いた形状変形の様子 4

加熱操作により十分な加熱を確認し、引き伸ばし変形を行った。変形開始後、変

形開始点へ向かって周囲の粒子がくびれ補正の引力により徐々に移動していく様子が確認できた。そのままデバイス操作点をゆっくりと上方向へ移動することで周囲の粒子のデバイス操作点への移動も確認できた。また操作点側の変形中の粒子が変形開始点の引力により移動し、最も外側にあった粒子がデバイスに追従しなくなるのを確認した。さらに変形を続けそれまで引力により操作点方向へ移動していた粒子の移動が収まり、そのまま引き伸ばしたところ形状の分裂を確認した。最終的に細長い突起を確認することができた。形状の下部にはところどころ粒子が固まりになっている部分が確認できこれは操作途中で、操作点が存在していた時間が長かった場所に引力により粒子が集まった結果であると考えられることができる。

3.3 温度変化に伴う赤熱現象の検証

全粒子に初期温度として1500℃を与え、時間経過による温度変化と色の変化を確認し、外気による温度低下の効果を検証する。図(3.5)(3.6)(3.7)(3.8)は本手法における温度低下の様子である。

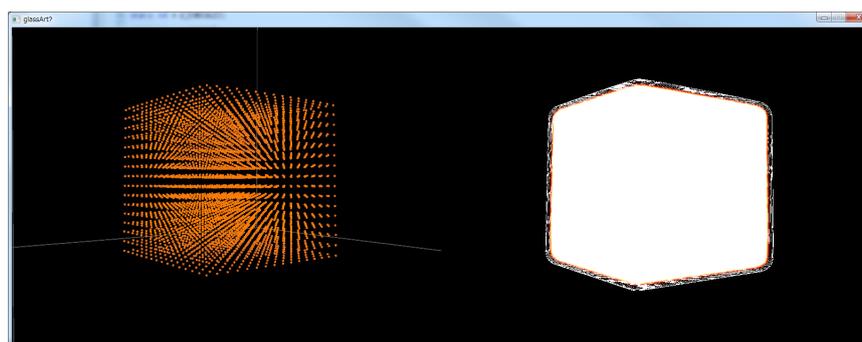


図 3.5: 外気による温度低下 1

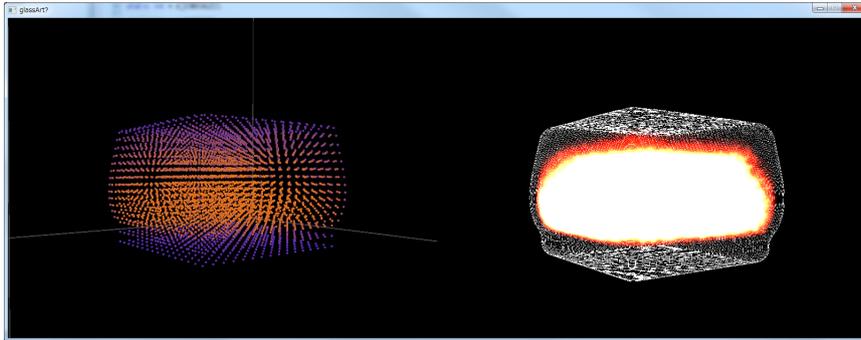


図 3.6: 外気による温度低下 2

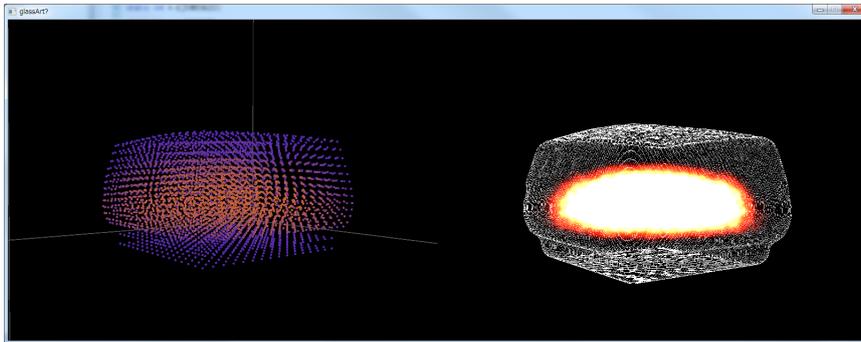


図 3.7: 外気による温度低下 3

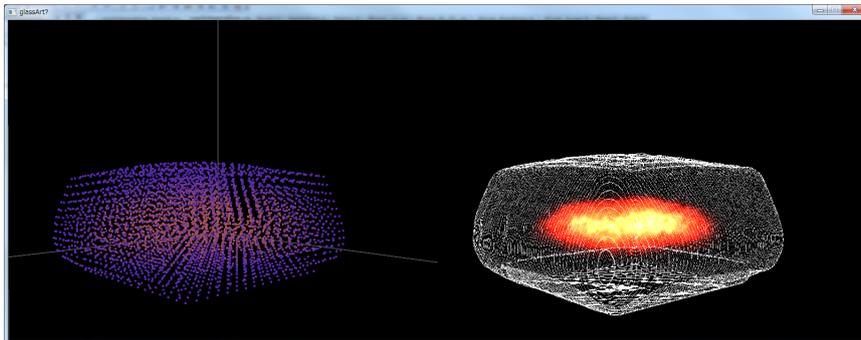


図 3.8: 外気による温度低下 4

初期状態では全体がほぼ白色を示し表面のみ部分がうっすらとオレンジ色がかつ

ている。時間経過に従い徐々に上部と底部からオレンジから薄暗い赤へ変化し、やがて背景の黒へ変化した。全体的に上部から早く熱が下がっていったのは、本手法では表面判定を粒子の密度によって行っているため、重力によって下部分の圧縮している部分が密度が高くなり表面付近の粒子として抽出しなかったためであると考えることができる。これにより粒子の温度変化に対応した赤熱表現による色の変化を確認した。

3.4 表面形状の描画の検証

透明で滑らかな質感と光源の移動による反射効果の変化を確認する。図(3.9)(3.10)は本研究における光源移動による変化の様子であり、光源が移動した際の表面の色の変化を表す。

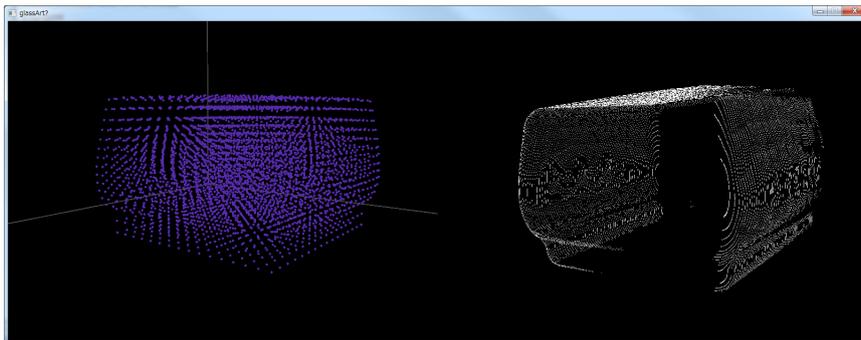


図 3.9: 光源移動による変化

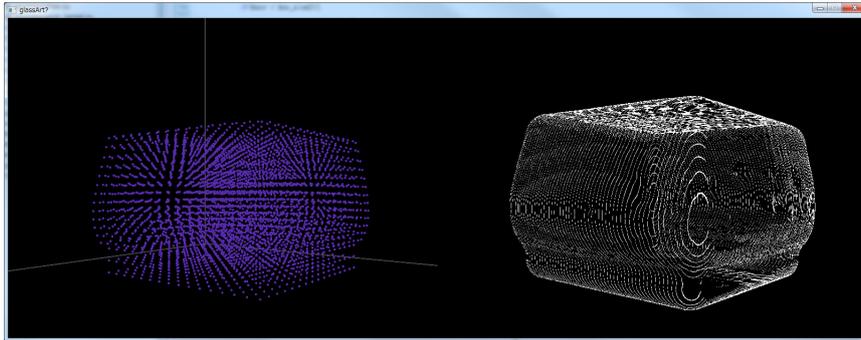


図 3.10: 光源移動による変化 2

光があたっている部分の表面を白色に表示可能なことを確認した。この白色部分により粒子集合の周囲に滑らかで連続的な表面形状が生成できていることが分かった。白色部分の表示結果からは視線方向に対して垂直なスライス状に縞模様が発現しているのを確認した。これはボリュームレンダリングの計算を軽減するためサンプリング間隔を広くとり視線ベクトル上のボリュームデータのサンプリング回数を少なくしているため、このような縞模様が発生していると考えられる。視線ベクトル上のサンプリング点で濃度ボリュームのボクセル値を参照し表面を定義する閾値と比較する際、あるサンプリング点とあるサンプリング点の間にガラス表面が存在した場合は表面は検出できない。これを回避するためにはサンプリング間隔を狭くし、表面検出の精度を向上することにより可能であるが、その分計算コストの増大を招く。そのため本手法では縞模様は発生するが対話的な操作に影響が出ない程度のサンプリング間隔による描画によって表面形状を可視化した。本手法ではガラス形状の周囲の背景などについては考慮していないため背景色の黒が透過した部分の色となっているが、あらかじめ用意した背景画像を表面の反射や屈折を考慮した描画が可能な環境キューブマッピング等の技術を用いることで周囲の写りこみや反射を表現することが可能になるため体験システムとしてのリアルさをさらに向上するにはこのような技術の使用を考慮することができる。

第 4 章

おわりに

提案手法の実装により、ガラスの赤熱現象と温度変化による粘性変化の性質を確認した。またデバイスを用いた操作による対話的な形状変形とくびれ補正処理を用いた引き伸ばし変形による分裂と温度操作を確認した。外気による温度変化と粒子間の熱移動を確認した。

これにより本手法により既存の仮想ガラスにない分裂を伴う引き伸ばし変形に対し有効性があることを示した。またさらに本手法では考慮していなかった力覚デバイスを用いた力覚フィードバックを行うことで、より良いガラス細工体験システムを実現し、多くの人にガラス細工の体験を可能にしていきたい。

謝辭

本研究を締めくくるにあたり、学部時代から続けての3年間の研究生活を見守って下さり、日常生活から研究方針、論文執筆、学会発表の準備等、多大なご指導とご協力を頂きました本校メディア学部の渡辺大地講師、並びに副査の三上浩二講師、桑原明栄子助教に深く感謝いたします。またプログラムの実装方法や研究方針、論文の執筆に多くのアドバイスを頂いた研究室の先輩方、苦楽を共にした、同輩達の励ましがあってこそ今ここに立っていられているということを実感しています、皆様本当にありがとうございます。そして研究の困難に諦めかけた時にも、いつでも温かく精神的にも経済的にも支え続けてくれた自分の両親の協力に、心より感謝いたします。

参考文献

- [1] 松宮雅俊, “陰関数曲面とパーティクルシステムを用いた仮想粘土細工による自由形状モデリングに関する研究,” **奈良先端科学技術大学院大学情報科学研究科博士論文** (2002).
- [2] YASUYUKI, O., KAZUNOBU, M., and NORISHIGE, C., “Particle-based visual simulation of virtual clay (special issue; new advances in computer graphics),” *Transactions of Information Processing Society of Japan* **42**(5), 1142–1150 (2001-05-15).
- [3] 中野幸一, 漢野救泰, 米., “仮想現実による伝統工芸品作製支援システムの開発,” **石川県工業試験場研究報告** (47), 31–38 (1998).
- [4] Kazuyoshi, K., Hiroaki, N., and Kouichi, U., “The design and implementation of an interactive 3d virtual ceramic work environment,” *Bulletin of Oita prefectural College of Arts and Culture* **35**, 129–138 (1997-12-31).
- [5] 永井亮一, 坂口正道, 荒. 藤., “力覚を伴うろくろ成形シミュレータの開発,” **日本機械学会ロボティクス・メカトロニクス講演会講演論文集 (CD-ROM) 2007**, 1A2–C08 (2007).
- [6] 小島勇治, ダン・ミン・グィエン, 下平勝紀, 塚田孝一, 中澤利顕, “ガラス工芸体験システム「GlassicPark」.” (<http://www.ei.nagano-nct.ac.jp/Labs/horiuchi/1998/procon/index.html>.) (1998).
- [7] 櫛田英功 石氏雅博 小池康晴 佐藤誠, “ガラス細工メタファーによる自由形状モデリング手法の提案,” **情報処理学会研究報告グラフィクスと CAD 研究会報告 Vol.2000(No.22(20000225))**, pp. 31–36 (2000).
- [8] 視覚デザイン研究所 編, [**ガラス工芸ノート**], 視覚デザイン研究所 (1998).
- [9] 黒川高明, [**ガラスの技術史**], アグネ技術センター (2005).
- [10] 由水常雄, [**ガラス工芸**], 文化出版局 (1981).

- [11] 松村潔, [バーナーワーク 酸素バーナーを使った耐熱ガラス工房], ほるぷ出版 (2007).
- [12] 船越美智代, [バーナーを使って (家庭ガラス工房)], ほるぷ出版 (1998).
- [13] 越塚誠一, [粒子法シミュレーション物理ベース CG 入門], 培風館 (2008).
- [14] Jos Stam and Eugene Fiume, “Depicting fire and other gaseous phenomena using diffusion processes,” *In SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, 129–136 (1995).
- [15] Dan Stora, Pierre-Olivier Agliati, Marie-Paule Cani, Fabrice Neyret, and Jean-Dominique Gascuel, “Animating lava flows,” *In Proceedings of the 1999 conference on Graphics interface '99*, 203–210 (1999).
- [16] 天田崇, “水の実時間アニメーション,” 奈良先端科学技術大学院大学修士論文 (2005).
- [17] Matthias Müller, David Charypar, and Markus Gross, “Particle-based fluid simulation for interactive applications,” *In Proceedings of the 2003 ACM SIGGRAPH/ Eurographics Symposium on Computer animation* (2003).
- [18] Mathieu Desbrun and Marie-Paule Gascuel, “Smoothed particles: a new paradigm for animating highly deformable bodies,” *In Proceedings of the Eurographics workshop on Computer animation and simulation '96*, 61–76 (1996).
- [19] 浜田敏, 笠置和幸, 阿. 作. 堀., “Vogel-tamman-fulcher の式における 3 つのパラメータの物理的意味,” **日本物理学会講演概要集** 56(270), 2-2 (2001).
- [20] 小田泰行, 藤本忠博, 村岡一信, 千葉則茂, “溶岩流の粒子ベース・ビジュアルシミュレーション,” **芸術科学会論文誌** 2(1), 51–60 (2003).

- [21] J.P.Morris, “Simulating surface tension with smoothed particle hydrodynamics,” *International Journal for Numerical Methods in Fluids* **33** (2000).
- [22] 原田隆宏, 政家一誠, 越塚 誠一, 河口洋一郎, “Gpu 上での粒子法シミュレーションの空間局所性を用いた高速化,” *日本計算工学会論文集 2008*, 20080016 (2008).
- [23] YUKI, S., SHINOBU, M., HAJIME, S., SHINICHIRO, M., YASUHIKO, N., and SHINJI, T., “Consideration for speculative rendering in pvr,” *IPSJ SIG Notes* **2005**(80), 145–150 (2005-08-03).
- [24] AKIRA, S. and TOSHIO, K., “On th smoothing of the numerical solution of the convection-diffusion equation by the cantral differnce method,” *Transactions of Information Processing Society of Japan* **27**(4), 397–400 (1986-04-15).
- [25] 澤見英男, “色温度とは.” (<http://cafe.mis.ous.ac.jp/sawami/%E9%BB%92%E4%BD%93%E8%BC%BB%E5%B0%84.PDF>.)
- [26] NVIDIA. (<http://jp.nvidia.com/page/home.html>.)
- [27] “CUDA Zone.” (http://www.nvidia.co.jp/object/cuda_home_jp.html.)
- [28] NOVINT. (<http://home.novint.com/>.)
- [29] 渡辺大地, “FineKernelToolKitSystem.” (<http://fktoolkit.sourceforge.jp/>.)
- [30] REN, Y., TAKAHIRO, H., and YOICHIRO, K., “Effective visualization of particle-based fluid with multiple refractions,” *IPSJ SIG Notes* **2009**(12), 73–78 (2009-02-09).