

2008年度 卒業論文

3次元入力デバイスを用いた
ガラス細工体験システムの研究

指導教員：渡辺 大地講師

メディア学部 ゲームサイエンス プロジェクト

学籍番号 M0105248

瀬田 陽平

2008年度 卒業論文概要

論文題目

3次元入力デバイスを用いた
ガラス細工体験システムの研究

メディア学部

学籍番号：M0105248

氏名

瀬田 陽平

指導
教員

渡辺 大地講師

キーワード

ガラス細工、ポイントベース法、CUDA、
メタボール、3Dモデリング

近年、アニメーションやビデオゲームといったコンテンツ上で3DCGグラフィックスを用いた様々な表現が開発されている。ゲームセンターのゲーム機や任天堂のWiiを代表とする家庭用ゲームなどでは、擬似的にボーリングやスキージャンプなどのスポーツを体験できるものがある。このような中、仮想空間上において様々な体験を擬似的に可能にするもののコンテンツの需要が高まっているといえる。そこで本研究ではガラス細工に触れた経験のない人でも、ガラス細工の技法の1つであるバーナーワークの操作手法を擬似的に体験可能なシステムを実現した。現実のガラス細工を初心者が行うには事前知識や安全性などの問題がある。これまでバーチャルリアリティを用いた粘土細工システムや6自由度デバイスを用いたガラス細工システムなどの研究がある。本研究では従来法において描画手法の違いから実現していなかった、加熱状態のガラスの持つ引き延ばした際細長く変形するという性質の表現について、メタボールの性質を利用した変形手法を応用して実現した。またGPGPUによる並列計算を行うことによりメタボール計算の高速化を図り、ガラス細工体験システムとして必要とするリアルタイム性を確保し、本システムの有効性を実証した。

目次

第1章	はじめに	1
1.1	研究背景・目的	1
1.2	ガラス細工とは	2
1.3	論文構成	5
第2章	ガラス細工シミュレーション	6
2.1	ガラスシミュレーション	6
2.1.1	ガラス形状の滑らかさの表現	6
2.1.2	ガラスの変形表現	6
2.1.3	ガラスの加熱表現	10
2.1.4	ガラスの質感表現	11
2.1.5	GPGPUによる高速化	13
2.2	本システムの概要	13
2.2.1	入力デバイスについて	15
2.2.2	操作方法について	15
2.2.3	加熱操作	16
2.2.4	変形操作	16
第3章	動作検証	18
3.1	実験結果と考察	18
3.1.1	リアルタイム性	18
3.1.2	メタボールの変形動作	19
3.1.3	フレネル反射	20
第4章	まとめ	22
	謝辞	23
	参考文献	24

目 次

1.1	バーナーワークの様子	3
1.2	吹きガラスで制作したガラス細工の花瓶	4
1.3	切り子ガラス細工のコップ	4
1.4	カットガラスのグラス	4
2.1	メタボール A とメタボール B の融合の様子 1	7
2.2	メタボール A とメタボール B の融合の様子 2	8
2.3	メタボールが 6 個の場合におけるガラスの初期形状	10
2.4	フレネル反射の例	12
2.5	本システムの構成図	14
2.6	3次元入力デバイス「Novint Falcon」	15
2.7	変形操作時の流れ 1	17
2.8	変形操作時の流れ 2	17
3.1	メタボールの変形動作 1-1	19
3.2	メタボールの変形動作 1-2	20
3.3	フレネル適用前	21
3.4	フレネル適応後	21

第 1 章

はじめに

1.1 研究背景・目的

近年、アニメーションやビデオゲームといったコンテンツ上で3DCGグラフィックスを用いた様々な表現が開発されており、アミューズメント施設における擬似的にボクシングを野球、サッカーなどのスポーツ体感できるものや、ダンスや剣術を体験できるようなものまで普及している。またさらに近年家庭用ゲームの分野に体感ゲームは進出しており任天堂のWiiにおけるスノーボードやボーリングなど多岐にわたっている。このように仮想空間上において、現実世界の様々な体験を擬似的に体感可能なシステムの需要は高まっているといえる。そこで今回現実ので行う体験の一つとしてガラス細工 [1][2][3] に着目した。ガラス細工は一般的にはあまり馴染みのないものであると考えられる、その理由として一般家庭でガラス細工を行う機材を用意することが金銭や安全面で困難であるためと考えられる。このようなことがガラス細工が一般的に馴染みのない原因であると考えられる。本研究ではガラス細工の経験のない人でもガラス細工の技法の1つであるバーナーワーク [4][5] について擬似体験可能なシステムを実現する。櫛田ら [6] の研究ではガラス細工の手法のひとつであるバーナーワークについてポリゴンベースでの描画手法によりガラス形状を表現している。また領域内の質点に対して熱パラメータを与えるという仕組みでガラスの加熱状態を表現している。これにより高い熱情報を持つポリゴンの頂点に対する移動を行っている。このような形でバー

ナーワークの熱を与えた部分に対し形状変形を行うというシステムを実現している。しかしこの従来法ではガラスの持つ引き延ばした際細長く変形するという性質の表現を行うことは不可能である。本研究では加熱状態のガラスの持つ引き延ばした際細長く変形する性質をメタボールのもつ性質を応用した変形手法とポイントベースでの描画により実現する。また GPGPU による並列計算を行うことによりメタボール計算の高速化を図りガラス細工体験システムとして必要とするリアルタイム性を確保し本システムの有効性を実証を行う。

1.2 ガラス細工とは

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

ガラスを熱によってガラスを変形する手法としては吹きガラス細工 [7] というものも存在する。これは内部が空洞なパイプの先端にガラスを付着させ、高温の炉の中で加熱し、十分に温まったガラスをパイプの先から空気を送り込むことによって膨張させ、グラスや風鈴などに成形していく技法である。図 1.2 は吹きガラスで制作したガラスの花瓶である。この技法のシミュレーション技法として吹きガラス体験システム「Glassic Park」 [8] が存在する。このシステムは実際にパイプ状の

デバイスに息を吹きこむことでガラス形状を膨張させる変形が行える。またパイプの回転によって生まれた遠心力による変形も可能である。



図 1.2: 吹きガラスで制作したガラス細工の花瓶

また冷めた状態で加工するものでは切り子ガラス [9] やカットガラスなどの手法が存在する。図 1.3 は切り子ガラスのグラスと図 1.4 はカットガラスの画像である。



図 1.3: 切り子ガラス細工のコップ



図 1.4: カットガラスのグラス

この他にもガラス細工には様々な手法が数多く存在している。しかし本研究で

扱うガラス細工技法は先述の高温で成形する技法の1種であるバーナーワーク手法に限定する。バーナーワークはピンセットを用いてガラスをつまみ引き延ばすことや、熱したガラス棒による盛り付けなどの変形により、上級者では動物や昆虫などのある程度複雑な形状も制作可能である。これは形状変形の自由度が他のお皿やグラスなどを主として制作する基本的なガラス細工手法と比較し高いといえる。このことから体験システムとして再現する技法にバーナーワークを選択した。

1.3 論文構成

本論文は全5章で構成する。2章前半ではメタボールを利用した仮想ガラスのシミュレーション手法、また、本システムの全体的な構成と仮想ガラスモデルに対しての変形操作についてバーナーモード時と変形モード時の2種類の変形手法の説明を行う。3章では提案手法を実装したシステムによる形状変形例を示しガラス細工体験が可能なることを描画速度のリアルタイム性の確認とともに実証する。最後に4章で本研究を総括する。

第 2 章

ガラス細工シミュレーション

2.1 ガラスシミュレーション

本項ではバーナーワーク技法の再現に必要なガラスのシミュレーション手法について述べる。

2.1.1 ガラス形状の滑らかさの表現

バーナーワークを対象としたガラス細工のシミュレーションを行うにあたりガラス形状の滑らかさを表現するために、立体モデルの描画をポイントベースレンダリング [10][11][12] の手法を用いる。ポイントベースレンダリングとは物体を多数の点の集合とみなし、各点を個別の描画要素として画像の描き出しを行う。例として、物体の形状が複雑でありポリゴン化を行った際に、個々のポリゴンがピクセルサイズより小さくなってしまいうような場合、ポリゴンベースに比べ、ポイントベースのレンダリングが有効である。今回はガラス細工という複雑に形状が変化しかつ滑らかさが要求されるモデルであるため、ポイントベースレンダリングを採用した。

2.1.2 ガラスの変形表現

バーナーワークにおけるガラスの形状変形を表現するにあたり、ガラスが熔解しハサミなどでつまんで伸ばすといった変形を行う場合その形状は滑らかに細く

なるように伸びて変形する。このような特徴を再現するため本研究ではメタボール [13][14] による形状変形を行う。メタボールとは陰関数曲面の一種であり、濃度分布の数式を定義し、用いることで形状を表現する技法である。具体的に表現できる形状としては水銀のように球体同士が融合する形状等があげられる。このメタボールの性質は熱したガラスの形状変形の性質と合致していることに着目し、ガラス細工の細長く伸びるという表現を行うためメタボールを選択した。固定したメタボールをひとつ配置し、デバイスにより操作するメタボールを設定する。これら 2 つの球の濃度分布により重なり合う地点で球が滑らかに融合していく変形が確認できる。これによりガラスの滑らかに細くなりつつ伸びるという特徴を表現した。図 2.1 と図 2.2 はメタボール A とメタボール B との融合の様子を表したものである。

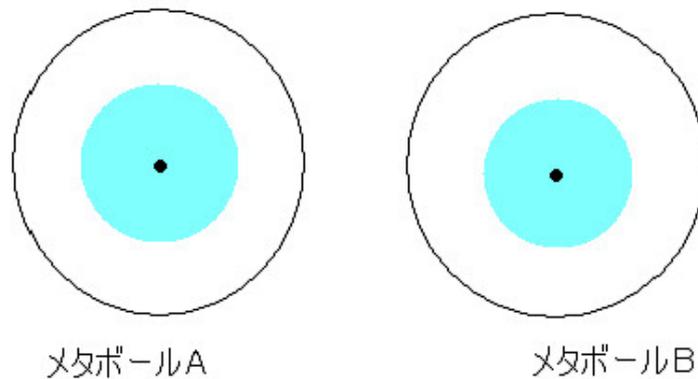


図 2.1: メタボール A とメタボール B の融合の様子 1

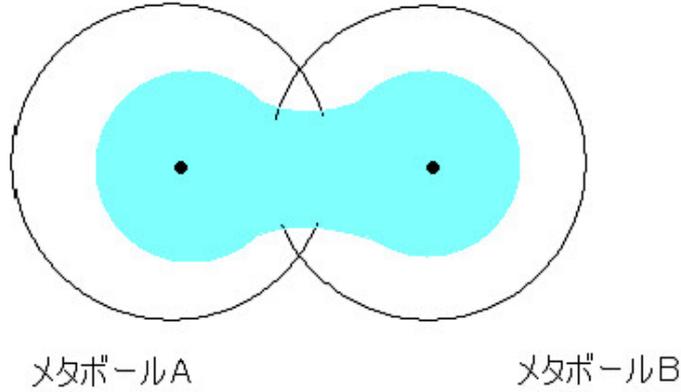


図 2.2: メタボール A とメタボール B の融合の様子 2

2つのメタボールの距離が近づき影響範囲が重なることで、重なり合った部分の濃度を加算し、最終的に閾値を超える濃度を持つ部分を形状の内側と定義する。算出した濃度はボクセルに値を格納する。ボクセルとは形状を3次元の格子点上において小さな立方体の集合で表す1要素である。今回は 128^3 個の濃度ボクセル空間 [15][16] を用意し、3次元空間と対応付けている。濃度ボクセル空間内に対して、メタボールの中心位置からボクセル空間に濃度計算を行う。その後、導出した濃度値と閾値との比較により形状を定義する。本研究で用いるメタボールは式 (2.1) と式 (2.2) により求める。 $M_i(\mathbf{B})$ はボクセル座標 \mathbf{B} での濃度値を表す個別のメタボール関数である。 C_i はメタボール i の中心座標、 D_i はメタボール i の中心座標における濃度値、 b_i はメタボール i の影響範囲の半径をそれぞれ表している。 $M(\mathbf{B})$ は全てのメタボール関数における同座標の濃度値を合計している。 n は全体のメタボールの個数を表している。

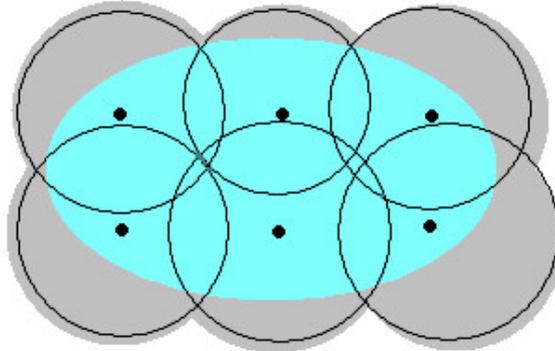
$$M_i(\mathbf{B}) = \begin{cases} 0 & (b_i < |\mathbf{B} - \mathbf{C}_i|) \\ (1 - \frac{|\mathbf{B} - \mathbf{C}_i|^2}{b_i^2})^2 D_i & (b_i \geq |\mathbf{B} - \mathbf{C}_i|) \end{cases} \quad (2.1)$$

$$M(\mathbf{B}) = \sum_{i=0}^n M_i(\mathbf{B}) \quad (2.2)$$

上式により求めた濃度値分布を用いてポイントベースでの描画を行う。ボクセルと対応づけた 3D 座標に濃度分布によって表現した形状に沿って点群モデルを配置し、点群モデルの集合としてガラス細工形状を描画する。点群モデルを配置するボクセル座標 \mathbf{P} は判定式 (2.3) により求める。 e は点群モデルを配置する表面の厚さを定義するごく少数の値であり、 t は濃度値と比較し形状の表面を定義する閾値である。また t はあらかじめ設定した定数である。これによりガラス形状の表面を点群モデルの集合により表現した。

$$t + e > \sum_{i=0}^n M_i(\mathbf{P}) \geq t \quad (2.3)$$

本研究ではこのメタボールの集合を用いガラス形状全体を表現した。図 2.3 はメタボール 6 個の場合におけるガラスの初期形状を表したものである。



メタボールが6個の場合における
ガラスの初期形状

図 2.3: メタボールが 6 個の場合におけるガラスの初期形状

2.1.3 ガラスの加熱表現

ガラス細工において熱した部分を変形するという作業はガラス細工ならではの特徴的なものである。ガラス細工体験システムを実現するためには、ガラスに対して熱を与える操作を可能にする必要がある。ガラスモデルと熱を関連付けるにあたり本研究では熱パラメータを格納する熱ボクセル空間を作成した。熱ボクセル空間は濃度ボクセル空間と対応づけるため同じく 128^3 個のボクセルによって構成している。

ガラス細工では加熱したガラス部分を変形するため、本研究において熱パラメータはガラス形状を構成するメタボールの移動に影響するものとした。ガラス形状を構成するメタボールの中心座標と同位置の熱ボクセル座標に格納されている熱パラメータを個々のメタボールの持つ熱パラメータとした。また、この熱パラメータの最大値は 1.0 とし、最低値は 0.0 とする。この熱パラメータの上下動によって熱を表現した。個々のメタボールの持つ熱パラメータが 0.5 以上のときそのメタボールは加熱状態と定義し、0.5 未満のときメタボールは非加熱状態とした。また

ガラス形状の変形操作つまりメタボールの移動は加熱状態のメタボールのみ行うものとし、非加熱状態のメタボールは移動しない状態とした。またメタボールの形状領域外の熱パラメータは常に0であるとした。

この熱パラメータとメタボールの移動の関係を用いてガラス形状の加熱を表現した。また本研究では熱によるガラスの色の変化は考慮していない。

2.1.4 ガラスの質感表現

ガラス細工をシミュレーションするにあたり、ガラスの視覚的な性質である反射と屈折などの質感を表現するため環境キューブマッピング [17] を用いた。環境キューブマッピングとは周囲の景色の写りこみをテクスチャで表現するための手法である。これによりガラス表面での反射と透過による屈折を同時に扱うことが可能となる。しかし初期状態では屈折率から算出した屈折色と、反射率から算出した反射色の平均から割り出した色で描画している。そのためガラス表面を見る角度を変えても表面の光の反射と屈折が変化しない。現実のガラスでは正面からみるとガラスの透きとおった先の景色がよく見えるが、斜め方向から見ると鏡のように反射し向こう側が見えなくなるフレネル反射という性質がある。図2.4はフレネル反射の例である。

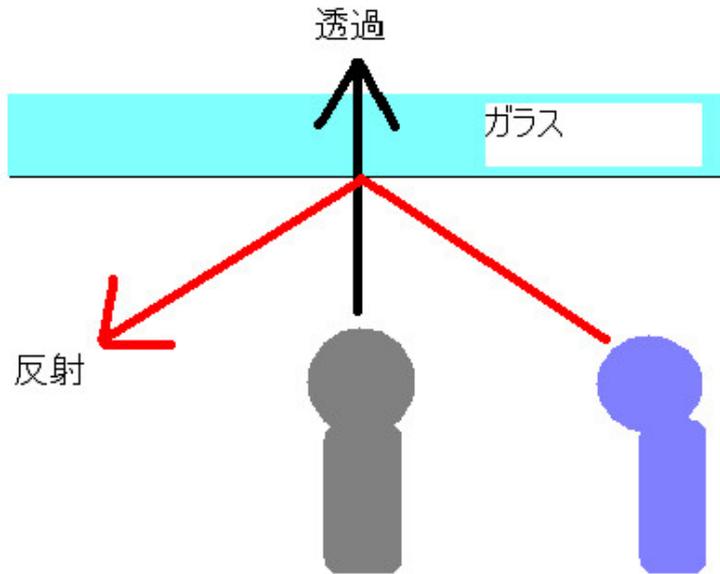


図 2.4: フレネル反射の例

ガラスをより現実に近い表現にするためフレネル反射を実装した。本研究では『The Cg Tutorial 日本語版』[18]に記述しているフレネル反射の近似式を用いて実装を行った。式 2.4 はフレネル反射の近似式である。 f は反射係数、 \mathbf{I} は光の入射ベクトル、 \mathbf{N} はモデル表面の法線ベクトル、 b はフレネルバイアス、 s はフレネル倍率、 p はフレネル指数を表している。

$$f = \max(0, \min(1, b + s(1 + \mathbf{I} \cdot \mathbf{N})^p)) \quad (2.4)$$

上式によって求める反射係数 f は、入射ベクトル \mathbf{I} と法線ベクトル \mathbf{N} がほぼ一致するとき 0、もしくはほぼ 0 に近くなる。この場合反射は起こらず大半の光が屈折する。また \mathbf{I} と \mathbf{N} が離れるにつれ反射係数 f は徐々に増加し、やがて急激に増加して 1 になる。この場合屈折は起こらず大半の光が反射する。フレネルバイアス b とフレネル倍率 s とフレネル指数 p の 3 つの値は上式において \mathbf{I} と \mathbf{N} の距離間から得る結果を調整することに用い、これによりガラスらしくみえる値を設定することが可能になる。式 2.5 は反射係数 f を用いて反射色 A と屈折色 B の混合

を行うものである。 C は最終的な描画色を表している。また色情報 A 、 B 、 C は RGB の 3 つの値として持っており、それにより色を決定している。

$$C = fB + (1 - f)A \quad (2.5)$$

2.1.5 GPGPU による高速化

今回メタボールを実装するにあたり 128^3 個のボクセル空間を定義しメタボールの濃度値と熱パラメータを計算している。これらの計算は空間すべての要素にアクセスするため一般的な CPU 上では 128^3 回という膨大な回数の計算を行う。そのため速度が極端に低下してしまう。しかしガラス細工を体験するという目的に際してリアルタイム性は必要不可欠なものであるといえる。この問題を解消するため GPGPU[19] を用いた並列処理を一部の計算に用い高速化を図った。GPGPU とは GPU という主に描画処理を行うハードウェアの加速度的な性能向上に伴い近年注目されている技術である。通常 CPU 上で行っている計算を GPU 上で並列処理を行うことが可能になる。通常 CPU の計算ユニットの数は 1 つであるのに対し、GPU では計算ユニットを 100 以上持っている。これにより膨大な数の計算を並列に行うことができるので速度向上につながる。GPGPU を利用するために標準の C 言語を採用した GPU 向けプログラミング環境である NVIDIA 社の CUDA[20] を用いた。本研究においては、ボクセル一つのメタボール濃度計算をこの CUDA を用いて行った。

これら上記の技術により形状変形が可能なガラス形状を表現した。

2.2 本システムの概要

3次元入力機能を持つデバイス「Novint Falcon」[21](以下 Falcon) により 3次元座標を取得し仮想空間上の制御点の位置を操作する。前章でのガラス細工シミュレーションに対する変形操作をこの 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、それぞれバーナーワークで用

いるガラス棒とピンセット類と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 2.5 は本システムの構成を図にあらわしたものである。

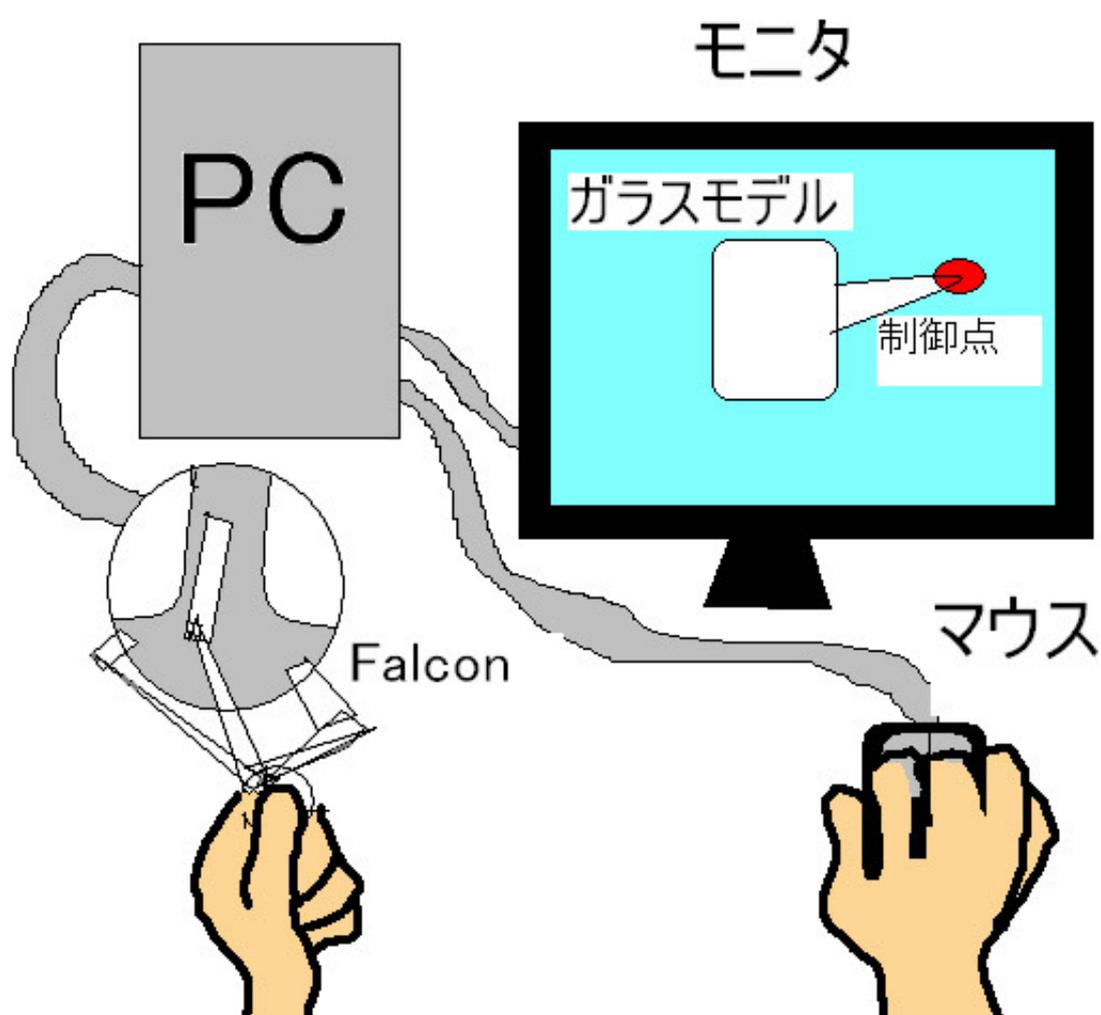


図 2.5: 本システムの構成図

2.2.1 入力デバイスについて

Novint 社の開発した力覚提示機能搭載 3 次元入力デバイスである。本研究ではこれを用い 3 次元位置を取得しバーナーワークのピンセットの操作に見立て物体の変形とバーナーでの加熱操作に用いた。図 2.6 のデバイスが「Falcon」である。

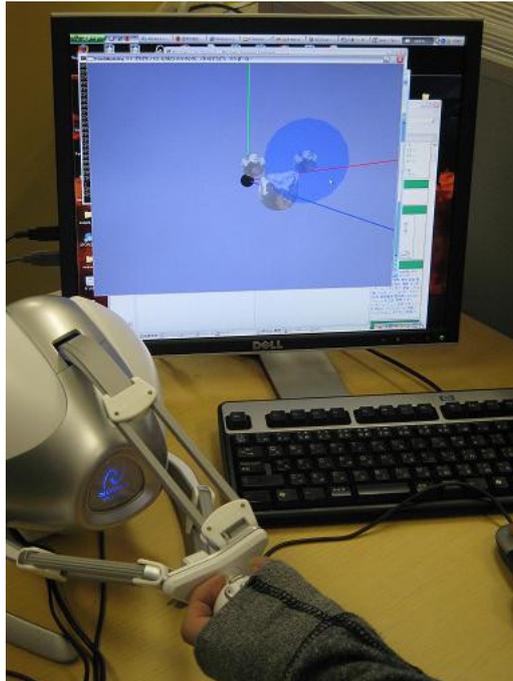


図 2.6: 3 次元入力デバイス「Novint Falcon」

2.2.2 操作方法について

本システムは基本的に Falcon とマウスを用いて操作する。実際にユーザーが手に持ち操作する部分を Falcon アームと呼ぶ。左手に Falcon アームを持ち制御点の操作を行う。Falcon アームには 4 つのボタンが存在しており、それを用い加熱操作と変形操作の切り替えとそれぞれ加熱と変形を行う。右手にマウスをもち視点の操作を行う。右クリックを押しながらドラッグで視点の回転を行い、左右のクリックを押しながら前後移動で拡大縮小することが可能である。

以下には本システムの基本操作である加熱操作と変形操作の処理の流れを示す。

2.2.3 加熱操作

本手法における加熱操作とは、操作領域内に存在するメタボールの熱パラメータを上昇を行うことが可能な状態である。このとき Falcon アームの操作ボタンを押している場合、制御点から制御半径の範囲内にメタボールの形状領域が存在するかの判定を行い、範囲内に存在している位置座標の熱ボクセル空間の持つ熱パラメータを加算する。またその際、制御半径の範囲外の熱ボクセル空間の持つ熱パラメータは減少する。これによりバーナーワークにおけるバーナーを用いたガラスの加熱操作を表現した。

2.2.4 変形操作

本手法における変形操作とは制御領域内に存在するメタボールの中心座標を移動可能な状態である。このとき Falcon アームの操作ボタンを押している場合、制御点から制御半径の範囲内に個々のメタボールの中心座標が存在するかの判定とそのメタボールが加熱状態であるかの判定を行う。加熱状態のメタボールが存在していた場合そのメタボールに対して移動を行うことが可能になる。メタボールの移動は、制御領域内に存在し最も制御点に近いメタボール 1 個の中心座標を制御点の位置に移動させることによって行う。メタボールの移動が行われた場合、ガラスの全体形状を表す濃度計算を行いガラス形状を描画する。

下記の図 2.7 と図 2.8 は変形時のシミュレーションの流れを図に表したものである。

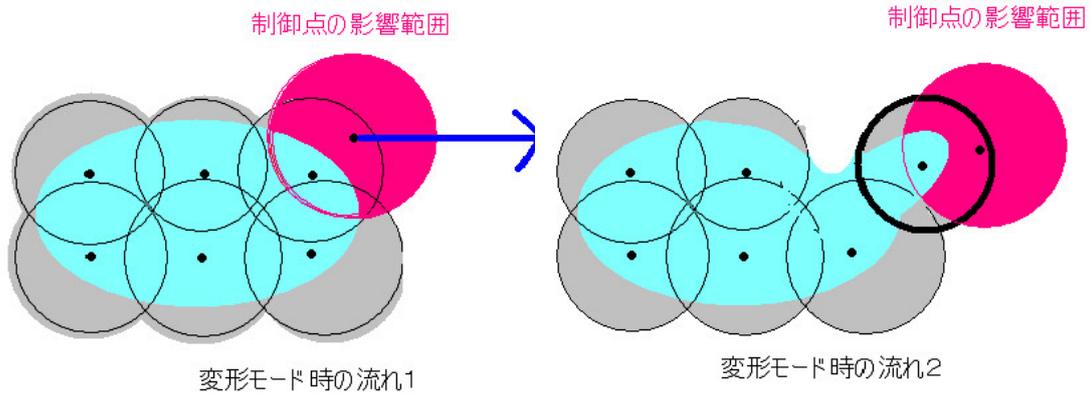


図 2.7: 変形操作時の流れ 1

図 2.8: 変形操作時の流れ 2

これによりバーナーワークにおけるピンセットを用いたガラスの変形操作を表現した。

第 3 章

動作検証

本章では、2 章と 3 章で述べた手法によるガラス細工体験システムの形状変形の
実行例を示し、本手法の有効性を検証し考察する。実装にはグラフィックツール
キットである FKSystem[22]、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。2 章および 3 章で述べた内容により実装を行った。3 次
元入力デバイス「Falcon」を用い形状変形操作を実行した。実験を行った計算機
の環境は表 4.1 の通りである。

表 4.1 実行環境

OS	WindowsXP SP3
CPU	Intel Core2 Duo E8500 3.16GHz
GPU	NVIDIA GeForce 9800 GTX グラフィックスメモリ 512MB
メモリ	3.25GB

3.1 実験結果と考察

本システムを動作し出力結果を以下に示す。

3.1.1 リアルタイム性

システム起動中もっとも計算コストの高いメタボールの濃度値計算部分のルー
プ中におよそ 15FPS 前後の描画速度を記録した。このことから本システムはリア

リアルタイムな処理が可能であるといえる。

3.1.2 メタボールの変形動作

メタボールを移動させガラス形状変形を行った結果、メタボール間の距離が開くほどその間の形状が滑らかに細くなっていく動作を確認した。これは想定した加熱状態のガラス形状変形の性質を満たしていると考えられる。またガラス細工体験システムとしてメタボールの移動により自由に形状変形が行えることを確認した。

以下の図 3.1、図 3.2 はメタボールの変形動作の画像である。これらの動作時のパラメータ設定として、閾値 0.8、影響半径 32 を使用した。

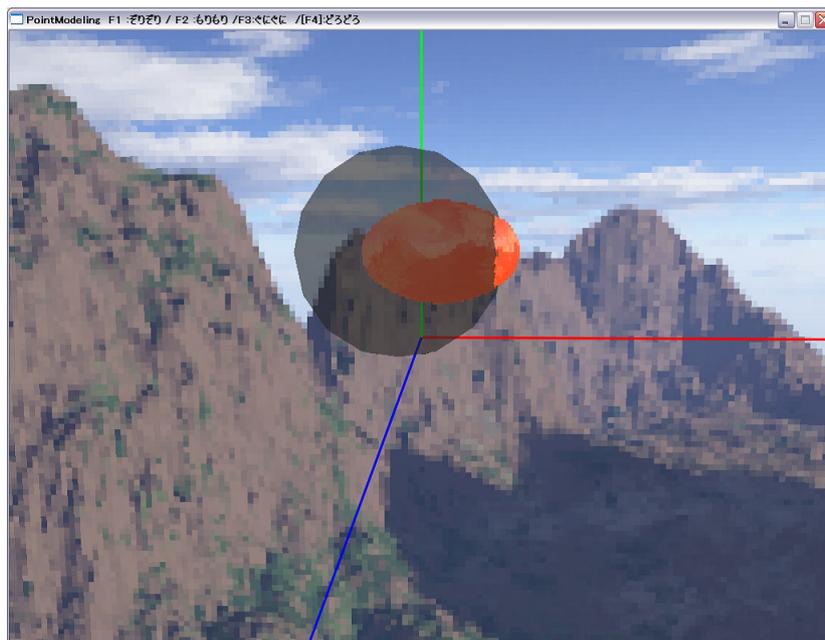


図 3.1: メタボールの変形動作 1-1

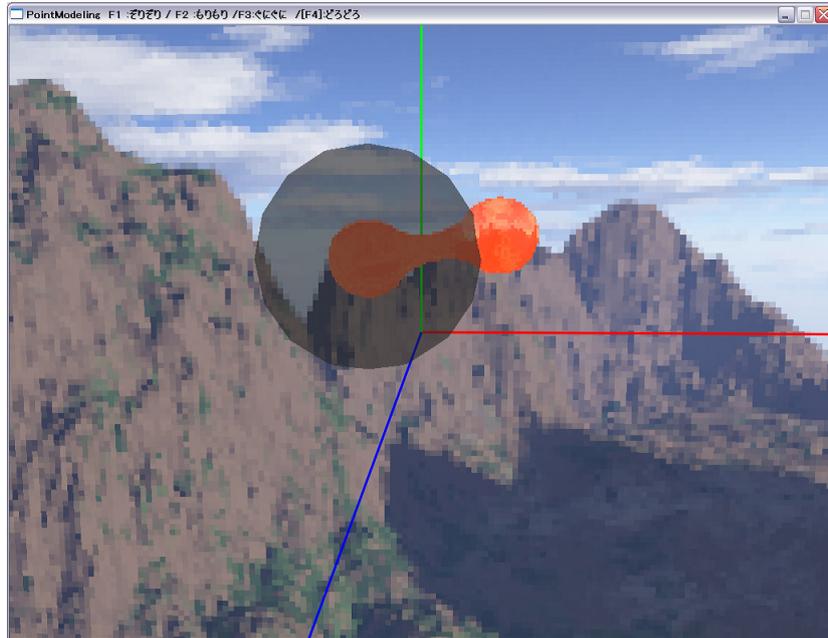


図 3.2: メタボールの変形動作 1-2

3.1.3 フレネル反射

フレネル適応前は全体に等しく反射と透過がかかっているため反射が強くガラスというよりむしろ金属の表現に近い印象を受けた。適応後は球の中心部に存在していた写りこみが透過している背景のみになった。また適応後の球の外周部は鏡のように反射していることがわかった。この結果によりフレネル反射が正しく実装されていることを確認した。

図 3.3 はフレネル適用前の画像であり、図 3.4 はフレネル適応後の画像である。

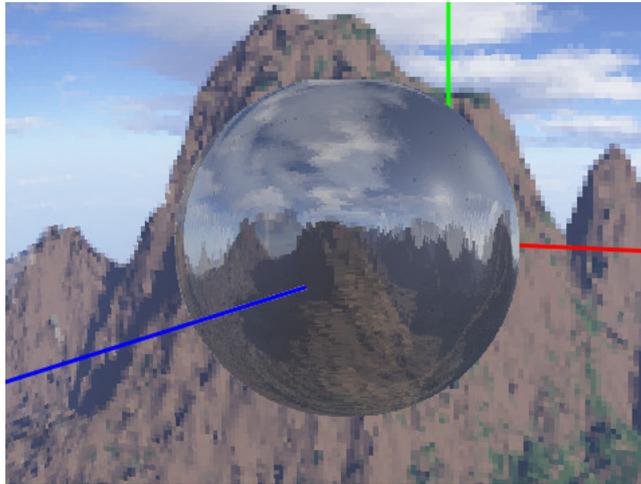


図 3.3: フレネル適用前

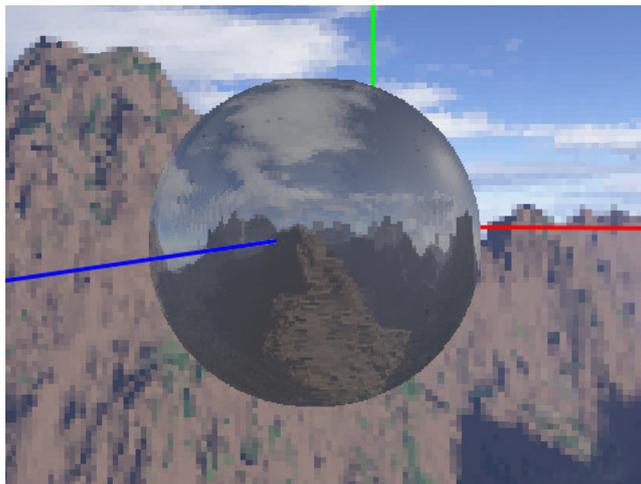


図 3.4: フレネル適応後

第 4 章

まとめ

本論文の総括として、まとめと今後の展望に関して述べる。本研究ではポイントベースレンダリングによる描画と、メタボールによる滑らかな形状変形により従来法で実現不可能であった加熱状態のガラスを引き延ばした際の細長く変形する表現の実現に成功した。また環境キューブマッピングによるガラスのテクスチャ設定の組み合わせによりガラスに近い質感を表現できた。また GPGPU を用いた速度向上によりリアルタイムに形状変形が実現できた。以上のことからバーナーワーク手法におけるガラス体験システムを実現できたといえる。

今回のガラス細工体験システムではガラスの色と現実のガラスに起こるような重力によつての変形については考慮していない。そのため現実のガラス細工の動作とはどうしても異なる部分が生まれてしまっている。また力覚デバイスを用いた変形操作時に起こる力のかかり方を直接手にフィードバックを行う表現によりさらにリアリティが増すと考えられる。

謝辞

本論文を作成するあたり、本当に多くの方のご協力をいただきました。ここに感謝の意を表します。これほど多くの方々の協力のもとで一つのことを成し遂げた経験は、今後の私の人生の大きな糧となると信じています。厳しくも温かいご指導くださった渡辺大地先生、日常の議論を通じて多くの知識や示唆を頂いたゲームサイエンス研究室の皆様本当にありがとうございました。

参考文献

- [1] 作花済夫、由水常雄、伊藤節郎, [ガラスの百科辞典], 朝倉書店 (2007).
- [2] 黒川高明, [ガラスの技術史], アグネ技術センター (2005).
- [3] 由水常雄, [ガラス工芸], 文化出版局 (1981).
- [4] 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).
- [5] 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).
- [6] 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).
- [7] 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).
- [8] 小島勇治 ダン・ミン・グイエ 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).
- [9] 山口勝旦, [江戸切子その流れを支えた人と技], 里文出版 (1993).
- [10] 藤本忠博 今野晃市 千葉則茂, “ポイントグラフィックス概説,” 芸術科学学会論文誌 Vol3(No1), pp.8–21.

- [11] 金井崇, 大竹豊, 川田弘明, 加瀬究, “GPU による点群ベース陰関数曲面の直接的レンダリング,” *グラフィクスと CAD 合同シンポジウム* (2006).
- [12] 竹内亮太 渡辺大地, “異なる物質の特性を表現した点群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.2008(No.80)*, pp. 23–28 (2008).
- [13] 金森由博 西田友是, “GPU を用いたメタボールの高速レンダリング,” *情報処理学会研究報告グラフィクスと CAD 研究会報告 Vol.2007(No.70(20070709))*, pp. 13–18 (2007).
- [14] 谷 寿人 栗山 繁 金子 豊久, “メタボール曲面から細分割曲面への変換手法,” *情報処理学会論文誌 Vol.45(No.3)*, pp.960–968 (2004).
- [15] 原田隆宏 越塚誠一, “GPU を用いた多層ポリゴンモデルの高速ボクセル化手法,” *情報処理学会論文誌* (2006).
- [16] 岡田達夫 C.Teodosiu 牧野内昭武, “ボクセルメッシュを用いた鍛造プロセスのシミュレーション,” *理化学研究所 ものづくり情報技術統合化研究プログラム* (2005).
- [17] “GLSL でキューブ環境マッピング.” http://www.arakin.dyndns.org/glsl_cubeenv.php.
- [18] Randima Fernando MarkJ.Kilgard, [*The Cg Tutorial* 日本語版], ボーンデジタル (2003).
- [19] NVIDIA, [*GPUGems3* 日本語版], ボーンデジタル (2008).
- [20] “CUDA Zone.” http://www.nvidia.co.jp/object/cuda_home_jp.html.
- [21] NOVINT. <http://home.novint.com/>.

[22] 渡辺大地, “FineKernelToolKitSystem.” <http://fktoolkit.sourceforge.jp/>.