

アーティスト制御可能な
オーロラシミュレーションに関する研究

筑波大学
図書館情報メディア研究科
2018年3月
関口智大

目次

第1章	序論	1
1.1	研究背景	1
1.2	研究目的	2
1.3	関連研究	2
1.3.1	オーロラのシミュレーションに関する先行研究	2
1.3.2	形状制御に関する先行研究	4
第2章	提案手法	6
2.1	全体の流れ	6
2.2	電場分布生成処理	8
2.2.1	アーティストによる形状指定	8
2.2.2	中間目標形状生成	10
2.2.3	中間目標形状の電場分布の算出	11
2.3	荷電粒子の運動計算処理	15
2.3.1	荷電粒子に働くローレンツ力の計算	15
2.3.2	荷電粒子間に働くクーロン力の計算	16
2.3.3	荷電粒子の位置の更新	18
2.3.4	荷電粒子の再配置	18
2.4	レンダリング	20
2.4.1	荷電粒子の落下シミュレーション	20
2.4.2	大気粒子の発光シミュレーション	22
2.4.3	カラーマップ生成	23
2.5	GPUによる並列化	23
2.5.1	荷電粒子間に働くクーロン力の計算処理の並列化	24
2.5.2	荷電粒子の再配置処理の並列化	25
2.5.3	レンダリング処理の並列化	28
第3章	結果と考察	29
3.1	編集画面	29
3.2	形状制御検証	31
3.3	考察	40
第4章	結論	43
4.1	まとめ	43
4.2	今後の課題	43

目 次

1.1	S 字状オーロラ [5]	2
1.2	渦状オーロラ [6]	2
1.3	分断・結合によって多重なオーロラが生成される様子	3
2.1	提案手法の流れ	7
2.2	荷電粒子を配置する 2 次元平面	8
2.3	オーロラの形状指定	9
2.4	分断が発生する際の接続情報の更新	9
2.5	結合が発生する際の接続情報の更新	10
2.6	中間目標形状における分断の表現	11
2.7	中間目標形状における結合の表現	11
2.8	荷電粒子のランダムサンプリング	12
2.9	生成した中間目標形状	13
2.10	生成した荷電粒子数分布	13
2.11	パーリンノイズの例	15
2.12	生成した電位・電場分布	16
2.13	縞模様のレイを含んだオーロラ [6]	18
2.14	再配置処理の有無による荷電粒子数分布の比較	19
2.15	再配置処理比較の際に設定した目標形状	19
2.16	スクリーン空間上での落下シミュレーションの模式図	21
2.17	空間分割法におけるグリッドと配置された荷電粒子の例	24
2.18	クーロン力計算の並列化の流れ	26
2.19	荷電粒子の再配置処理の並列化の流れ	27
2.20	スキャン処理	27
2.21	レンダリング処理の並列化の流れ	28
3.1	オーロラ形状の編集画面	30
3.2	分断の設定	30
3.3	結合の設定	30
3.4	シーン 1 で指定したオーロラ形状	31
3.5	シーン 2 で指定したオーロラ形状	32
3.6	シーン 3 で指定したオーロラ形状	32
3.7	シーン 4 で指定したオーロラ形状	33
3.8	シーン 5 で指定したオーロラ形状	34
3.9	シーン 1 のレンダリング結果	35

3.10 シーン 2 のレンダリング結果	36
3.11 シーン 3 のレンダリング結果	37
3.12 シーン 4 のレンダリング結果	38
3.13 シーン 5 のレンダリング結果	39
3.14 折り返し後の形状 (図 3.9(h)) における電位分布	40

表 目 次

2.1	等色関数 [29]	23
2.2	配列に格納した荷電粒子番号とセル番号	25
2.3	ソート後の荷電粒子番号とセル番号	25
2.4	荷電粒子番号の開始位置と終了位置	25
3.1	実装・実験環境	29
3.2	実験に用いた各パラメータ	31
3.3	再配置処理に関する各パラメータ	31

第 1 章

序論

1.1 研究背景

近年, 物理シミュレーションによって生成された自然現象のコンピュータグラフィックス (CG) は, 主にゲームや映画といったエンターテインメント分野で幅広く利用されている. 従来の CG 作成手法はアーティストの技量に依存する部分が多く, 非常に膨大な作業量が必要であったが, 物理シミュレーションの登場により, 高品質な CG アニメーションを効率よく大量に生産することが可能となった. 物理シミュレーションを用いた CG の生成では, 自然環境に則したパラメータを設定し, 現実の物理法則に基づいた計算を行うことで現実的なアニメーション生成を可能としている. 例えば, 水や炎や煙などに代表される流体の物理シミュレーションでは, 密度や質量, 粘性や初期速度などを設定する必要があり, このパラメータの設定によって, 最終的に生成されるアニメーションが大きく変化する. そのため, アーティストが望んだ色や形状を得るためには, パラメータを細かく調整する必要があり, また, パラメータを変えても望んだ結果が得られるかわからず, 多くの試行錯誤が必要となる. これが物理シミュレーションの使用を難しくしている. このような細かな調整や試行錯誤を自動化するために, ペンなどを用いた直感的な入力に基づき, シミュレーションをインタラクティブに制御する手法の研究が, 炎や煙など様々な自然現象に対して行われている [1, 2].

CG で表現される自然現象の一つとして, 極地周辺で観測されるオーロラ現象がある [3]. オーロラの CG は, その神秘的な見た目や物理的な特性から, ゲームや映画といったエンターテインメント分野での利用の他に, プラネタリウムなど学術的な分野でも利用されている. オーロラが発生する原因は, 太陽から放射された荷電粒子が, 地球上の酸素や窒素といった大気粒子と衝突することで起こる発光現象が原因であると言われており, 電磁場から受けるローレンツ力によって, カーテンが揺らめくような動きをしたり, S 字状 (図 1.1) や渦状 (図 1.2) といった形状を形成することが判明している [4]. このような挙動がどんなメカニズムによって生じるのかは未だ完全に解明されていない. そのため, 物理的な正確さを追及するのではなく, 物理的な特性を加味した上で実際のオーロラとの視覚的な妥当性を追求するビジュアルシミュレーションに関する研究 [7, 8] が多く行われている.

オーロラのビジュアルシミュレーションでは, オーロラを磁場方向から見た 2 次元平面上で, 電場や磁場による力を考慮したパーティクル法によるシミュレーションを行う事で荷電粒子の動きを表現する. その後, 計算した荷電粒子の位置から, 磁場による影響を考慮した疑似的な落下シミュレーションを行い, 荷電粒子と大気粒子との衝突地点, つまり発光地点を算出し, 最終的なレンダリング結果を生成する. このような手法を用いることで, 実際のオーロラのような CG を作成することが可能となるが, 他の自然現象の物理シミュレーシ

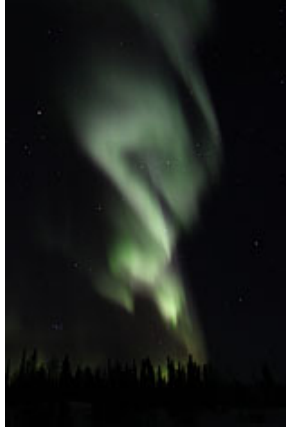


図 1.1: S 字状オーロラ [5]



図 1.2: 渦状オーロラ [6]

ンと同様に、アーティストが望む形状のオーロラアニメーションを作成するためには、初速度や電荷などの複雑なパラメータ設定が必要となる。

1.2 研究目的

複雑なパラメータ設定を行わずに、オーロラの複雑な挙動をシミュレーションで再現するためには、インタラクティブに形状を制御する手法が必要となる。本研究では小島らの手法 [7] をベースに、オーロラの形状を制御可能なシミュレーション手法を提案する。オーロラが形状を変化させる原因は、電磁場から受けるローレンツ力であると考えられている。そのため、アーティストによって指定された形状に近づくような力を発生させる電場分布を生成することで、オーロラアニメーションの形状制御を実現する。またオーロラ現象では、図 1.3 に示すようにシート状の構造が分断して多重のオーロラになったり、複数のオーロラが結合して一つながりのオーロラを形成するといった、分断と結合現象が発生することがある。電場分布を生成する際に、オーロラの接続情報も考慮することでこれらの現象を表現する手法も提案する。

1.3 関連研究

1.3.1 オーロラのシミュレーションに関する先行研究

オーロラの分布を数式で表される曲線によって表す研究として、井上ら [9] は一葉双曲面を用いてオーロラの形状を表現する手法を提案した。高度が高くなるほどオーロラの色が薄くなる点を考慮し、観測データから連続的な近似モデルを作成することで、オーロラの色を表現した。実験結果においてはオーロラの静止画のみを生成しているが、今後の展望として、タイムステップごとに一葉双曲面のパラメータを変化させることでアニメーション生成が可能であると述べている。また、Baranoski ら [3] は、正弦曲線近傍に配置した荷電粒子を一定間隔ごとに落下させ、荷電粒子と大気粒子との衝突地点を求め、その点を発光点としてスクリーンに投影することでオーロラの静止画を生成した。これらの手法では、オーロラ

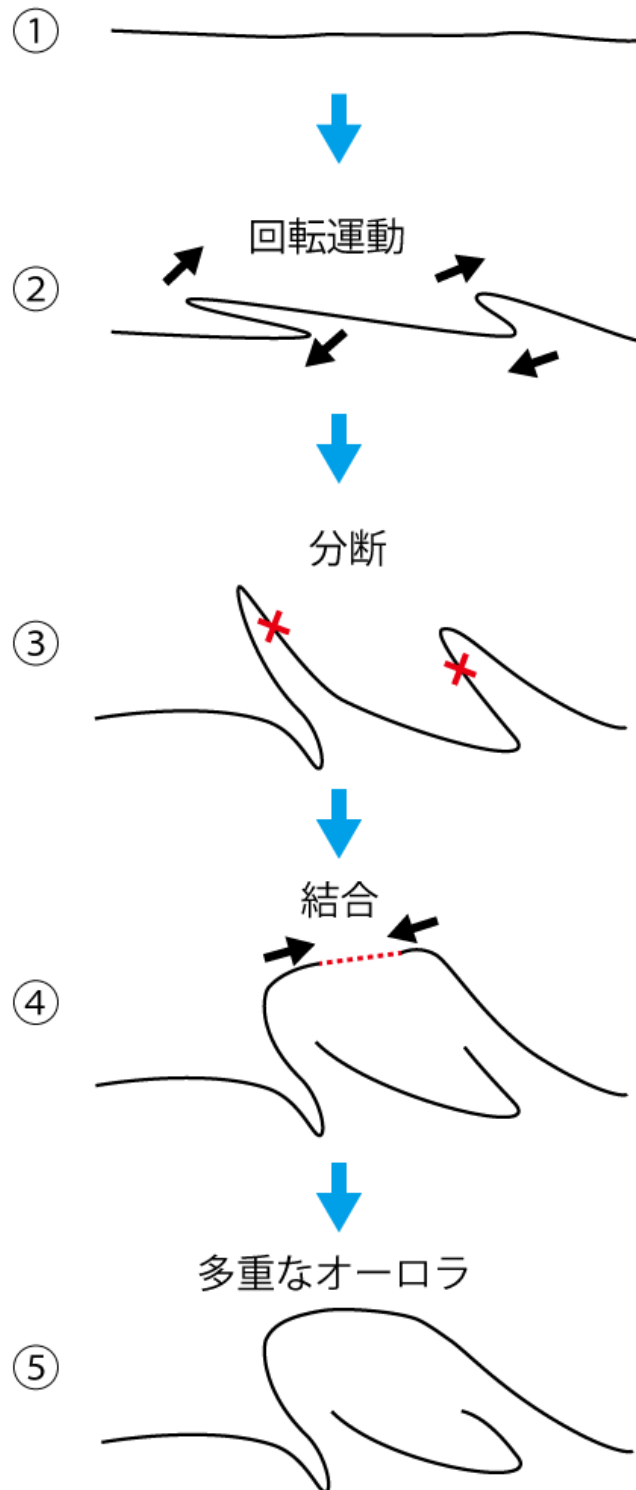


図 1.3: 分断・結合によって多重なオーロラが生成される様子

の色を決定する処理に観測データを用いているため、実際のオーロラに基づいた色しか生成できず、アーティストが独自に色を調整することができない。また、オーロラ形状の変化が一葉双曲面や正弦曲線のパラメータに依存するため、複雑なオーロラの揺らめきに対応することも難しい。

米山ら [8] は、オーロラの形状を複数の正弦曲線の重ね合わせとして表現し、オーロラが発光する物理的な過程を考慮したレンダリング手法を提案することで、オーロラの様々な色を表現することを可能とした。また、正弦曲線の振幅や周波数などのパラメータを調整することで、オーロラがカーテンのように揺れる形状を表現することができる。しかし、オーロラはS字状や渦状の形状をなすことがあり、正弦曲線ではこのような形状を表現することができない。Baranoskiら [10] は、オーロラをカーテンのような細長いシートで表現し、その中にランダムに荷電粒子を配置したモデルを提案した。電場や磁場から配置した荷電粒子にかかる力を求め、その力を基にカーテン状のシートの動きを計算することで、オーロラの動きを表現している。1枚のシートを用いてオーロラを表現しているため、シートが折れ曲がるようにしてS字状や渦状構造を作り出すことはできるが、実際のオーロラに見られるような、オーロラが裂けてS字状や渦状構造を生成する動きは表現することができない。津郷ら [11] はオーロラの形状を荷電粒子の集合として扱い、電磁場シミュレーションを行う事でオーロラが裂けるような動きを表現した。しかし、荷電粒子と大気粒子との衝突判定に高度ごとの大気粒子の比率を用いているため、高度が低くなるほど衝突回数が増えてしまい、結果として実際のオーロラとかけ離れたアニメーションを生成してしまうことがある。

小島ら [7] は、S字状や渦状といったオーロラ特有の複雑な動きを表現するモデルを提案した。また、その後の研究 [12] では、オーロラ分布を疑似的な非圧縮性流体として表現し、荷電粒子同士の接続関係を考慮に入れることで、図 1.3 のようにシート状のオーロラが分断・結合を繰り返して多重なオーロラを形成するアニメーションの生成を可能とした。しかし、どちらの手法も電場や磁場、電流から荷電粒子にかかる力を計算して運動計算を行っているため、アーティストの望むアニメーションを生成するためには、複雑なパラメータ設定が必要となる。また、荷電粒子間の距離に依存して接続関係の更新を行っているため、アーティストが望んだ位置での分断・結合を発生させることが難しい。

本研究では、小島ら [12] の手法で表現可能であったS字状・渦状の形状や、オーロラの方断・結合に対応しつつ、具体的な形状や分断・結合が発生する位置をアーティストのマウス操作によって制御可能な手法を提案する。

1.3.2 形状制御に関する先行研究

物理シミュレーションでのパラメータ調整を自動化するため、様々な自然現象のシミュレーションをインタラクティブに制御する手法の研究が行われている。Kawadaら [2] は、爆発の流れを制御パスによって指定し、パスに従って密度・速度・圧力・温度場を手続き的に考慮することで、物理的特性を考慮しながら爆発の形状制御を行う手法を提案した。Shiら [13] は、フィードバック制御を用いて水の形状制御を行う手法を示し、流体制御の評価基準として(1)制御性、(2)使いやすさ、(3)流体らしさ、(4)安定性の4つを提示した。Fattalら [14] は、通常のナビエ・ストークス方程式を用いた流体シミュレーションにDriving forceという外力とSmoke gatheringという処理を加えることで、煙の形状制御を可能とした。Driving

force は、現在の煙の密度分布を目標形状に近づけるような力のことであり、目標形状の密度勾配と現在の密度分布から力の方向と大きさを求め、ナビエ・ストークス方程式の外力項として加える。また、Smoke gathering という処理に従って密度を直接変化させることで、目標形状への変形をより鮮明なものにしている。また、水谷ら [15] は現在の形状と目標形状の差分から計算される累積分布を Driving force の係数調整に用いることで、煙よりも強い浮力の影響を受ける炎の形状制御を可能とした。

これらの手法は、爆発、水、煙、及び炎を対象とした形状制御手法であり、この他にも雲 [16] や花火 [17] や弾性体 [18] など、様々な現象の形状制御手法が提案されているが、オーロラの形状制御を行う手法は未だ提案されていない。本研究では、アーティストの入力したオーロラ形状に従って、それに近づくような力を発生させる電位・電場分布を生成することで目標形状への変形を行う。

第 2 章

提案手法

2.1 全体の流れ

提案手法は大きく分けて以下の 3 つの処理から構成される。

1. アーティストが設定した目標形状から、形状を制御するための電場分布を生成する処理
2. 生成した電場分布を用いて荷電粒子の運動計算を行う処理
3. 荷電粒子の落下シミュレーションによるレンダリング処理

提案手法の全体的な流れを図 2.1 に示す。なお、本手法では図 2.2 のように高さとして y 軸を磁場方向にとり、それに垂直な水平面を xz 平面とする。形状制御や運動計算 (1 と 2 の処理) は図 2.2 赤枠で示された xz 平面上で行う。

最初の電場分布生成処理では、まずアーティストのマウスドラッグ操作によって、生成したいオーロラの変形前の元形状と変形後の目標形状を指定する。この際、図 1.3 で示されるような分断・結合現象を考慮するため、オーロラの接続情報も設定する。次に、設定した元形状と目標形状の間の中間目標形状を 3 次のスプライン補間によって生成し、その近傍に荷電粒子を配置する。その後、平面をグリッドセルで分割し、配置された荷電粒子を基に各セルにおける電荷密度を求め、電位・電場分布を反復法によって計算する。

次に、生成した電場分布を用いて実際に荷電粒子の運動を計算する。提案手法では荷電粒子に働く力として、電磁場から受けるローレンツ力と、荷電粒子の間で働く疑似的なクーロン力の 2 つを扱う。計算した 2 つの力を用いて荷電粒子の位置の更新を行い、その後荷電粒子の過度な密集や目標形状からの離反を防ぐための削除・再配置処理を行う。

最後に、位置を更新させた荷電粒子を疑似的に落下させ、荷電粒子と大気粒子との衝突位置を計算する発光シミュレーションを行う事で、最終的なレンダリング結果を生成する。

本章では、提案手法を各ステップごとに詳しく説明していく。まず、2.2 節ではアーティストによる形状指定から、指定された形状に基づく電場分布生成処理について述べる。2.3 節では生成した電場分布を用いて荷電粒子にかかる力を計算し、運動計算を行う処理について述べ、2.4 節で物理的な発光過程を考慮した最終的なレンダリング手法について述べる。最後に、2.5 節で各処理の GPU による並列化を含めた実装方法について述べる。

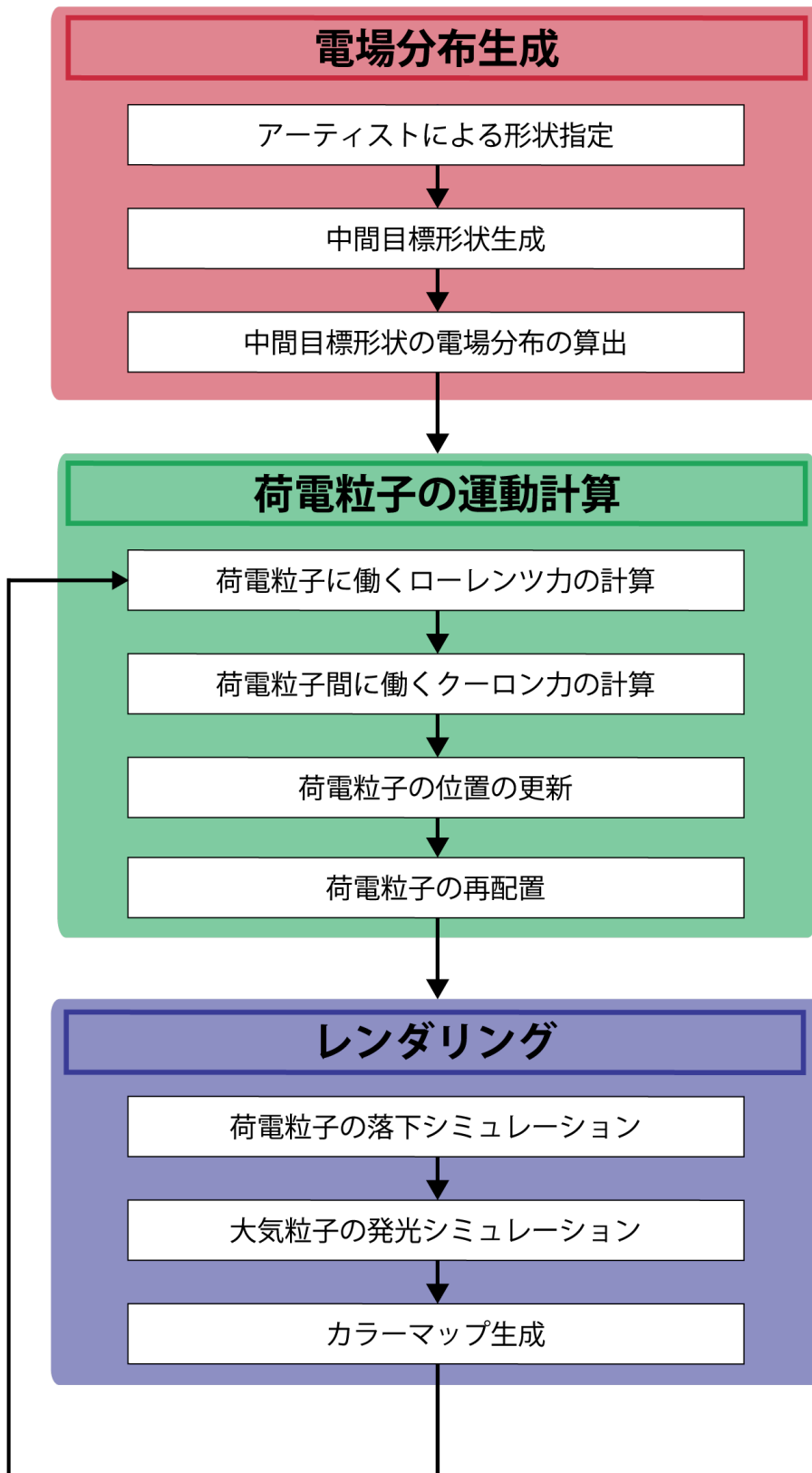


図 2.1: 提案手法の流れ

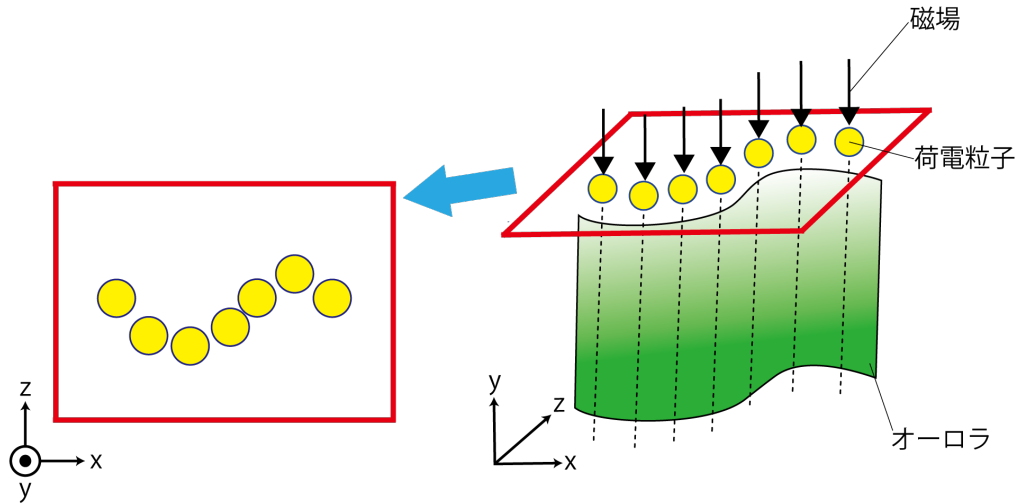


図 2.2: 荷電粒子を配置する 2 次元平面

2.2 電場分布生成処理

2.2.1 アーティストによる形状指定

初めに、アーティストが生成したいオーロラの形状をマウスドラッグ操作によって与える。マウスをクリックしてから離すまでに描いた曲線を一本のシート状構造として扱い、得られた曲線を線分で離散化する。この際、図 2.3 に示すように、離散化に使用する各線分の端点を制御点と呼称する。 i 番目の制御点の位置 P_i は、ドラッグ操作によって取得された j 番目のマウス位置を M_j とすると次の線形補間式 (2.1) で求めることができる。

$$P_i = (1 - p)M_q + pM_{q+1} \quad (2.1)$$

ここで、アーティストが予め設定した制御点の総数を n 個、ドラッグ操作によって得られたマウス位置の総数を m 個とすると、補間係数 p は $i \times \frac{m}{n}$ の小数部分、補間に用いるマウス位置座標番号 q は $i \times \frac{m}{n}$ の整数部分となる。一度のドラッグ操作で描いた曲線を一本のシート状構造として扱うため、その離散化に用いた各制御点には、前後の制御点との接続情報を格納しておく。また、オーロラの複数のシート状構造は互いに交差しないことが観測上知られている [19] ため、ドラッグ操作によって新しく指定した曲線が、既存の制御点を結んだ線分と交差してしまう場合はその曲線の追加を行わない。これらの処理を、変形前の元形状と変形後の目標形状についてそれぞれ行う。複雑な変形アニメーションを作成する場合には、途中過程での形状を含めた、複数個の目標形状を入力として与える。

オーロラ現象では、図 1.3 に示されるように一本のシート状構造が分断して複数のシート状構造を形成する分断現象や、逆に複数のシート状構造が結合して一本のシート状構造を形成する結合現象が発生することがある。元形状と目標形状の間での分断現象と結合現象は、各制御点が保持している前後の制御点との接続情報を更新することで表現する。その際、元形状の接続関係を単純に変更してしまうと、元々接続していた線分が不連続に消失したり、逆に元々分離していた部分の不連続に接続する現象が生じてしまう。これを防ぐため、元形

①ドラッグ操作による形状指定 ②制御点を結ぶ線分で離散化



図 2.3: オーロラの形状指定

状の分断・結合が発生する線分上に新しく制御点を2つ配置する．一方目標形状では，分断・結合が発生した後の位置に新しく制御点を配置し，同様にその制御点との接続関係を追加する．この処理によって，2.2.2節で説明する中間目標形状生成処理において，図2.6と2.7に示されるように線分が徐々に発生・消滅していくような中間目標形状を生成することができる．分断・結合が発生する場合の制御点の接続関係の更新を図2.4と2.5に示す．

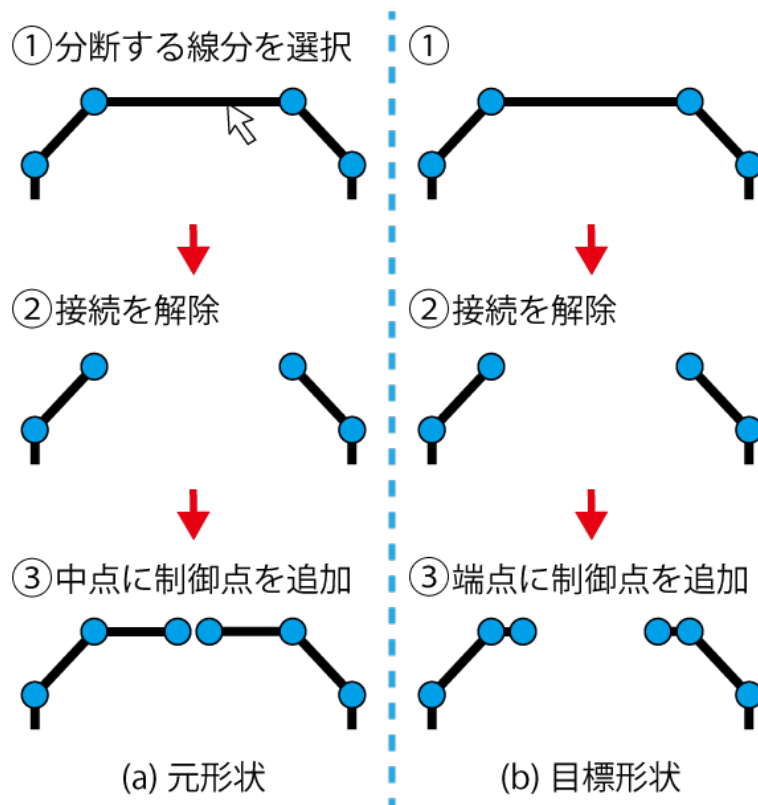


図 2.4: 分断が発生する際の接続情報の更新

分断が発生する場合には，元形状内で分断現象が発生する線分をアーティストが選択する（図2.4(a)の①）．その後，元形状で選択された線分の端点にあたる制御点の接続情報を削除し（図2.4(a)の②），新たに線分の midpoint に生成した制御点との接続情報を追加する（図2.4(a)の③）．目標形状では，元形状で選択された線分に対応する制御点の接続情報を同様

に削除し、新たにその線分の端点に生成した制御点との接続情報を追加する (図 2.4(b)).

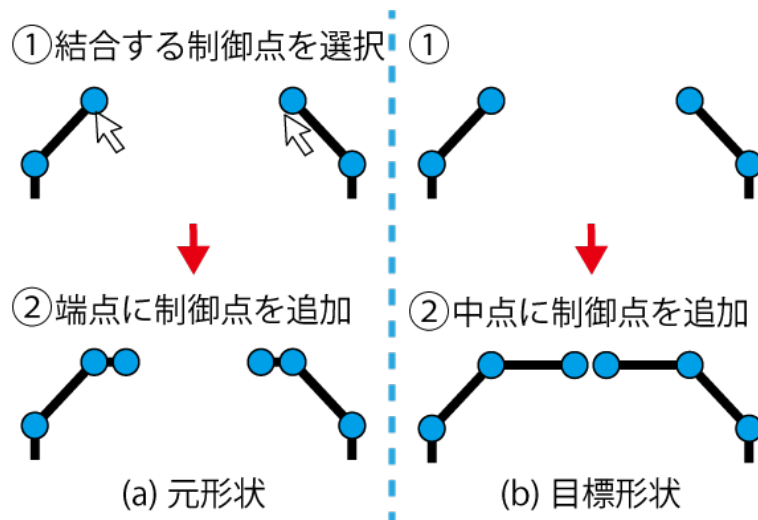


図 2.5: 結合が発生する際の接続情報の更新

結合が発生する場合には、元形状内で結合したい2つの制御点をアーティストが選択する (図 2.5(a) の①). その後、元形状では選択された制御点の位置に新たに制御点を追加し、その制御点との接続情報をそれぞれ追加する (図 2.5(a) の②). 目標形状では、選択された制御点の midpoint に新たに制御点を追加し、同様にその制御点との接続情報を追加する (図 2.5(b)). また、小口 [19] によるとオーロラは枝分かれするような構造を形成することはないため、結合の際に選択できる制御点は接続情報を一つしか持たない点、つまり曲線の端点にあたる制御点のみに限定する.

2.2.2 中間目標形状生成

本手法では、元形状と目標形状の間を単純に補間し、その近傍に荷電粒子を配置する手法ではなく、目標形状を基に電場分布を生成し、それを基に荷電粒子を移流させる。目標形状から生成した電場分布によって荷電粒子を移流させることで、オーロラの揺らめくような動きを再現しつつ、荷電粒子の分布を目標形状へ近づけることができる。しかし、アーティストが指定した目標形状から直接電場分布を生成すると、元形状と目標形状が大きく異なっていた場合に、生成した電場分布が荷電粒子に対して及ぼす影響が小さくなってしまい、元形状から目標形状への形状変化が上手く実行されない。そのため、変形前の元形状と変形後の目標形状の間の形状にあたる、中間目標形状を生成する。中間目標形状から生成した電場分布によって荷電粒子の移流を行い、次の中間目標形状を新しく生成するといった処理を繰り返すことで、元形状と目標形状が大きく異なっていた場合にも形状変化を可能とする。中間目標形状は、元形状と目標形状と同様に制御点とそれを結ぶ線分によって表現される。各フレームにおける中間目標形状の制御点位置は、元形状と目標形状の対応する各制御点の位置を3次スプライン補間を用いて補間することで求める。元形状と目標形状の制御点を、アーティストがドラッグ操作を行った順番に一つ一つ対応させ、対応する制御点間で補間を行う。また、分断・結合を指定した際に元形状と目標形状に追加した制御点もそれぞれ対応させる

ことで、図 2.6 と 2.7 に示されるように、元形状から目標形状に近づいていくほど徐々に線分が削除・追加されていく様子を表現することができる。

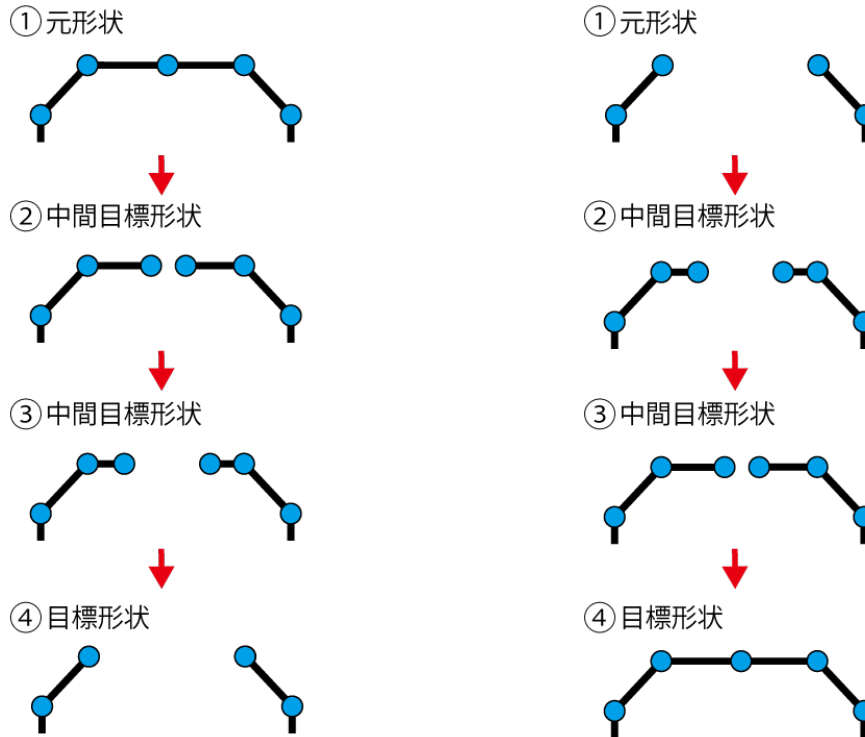


図 2.6: 中間目標形状における分断の表現 図 2.7: 中間目標形状における結合の表現

中間目標形状は、元形状から目標形状へのシミュレーションを行っている間、全てのフレームで生成するのではなく一定フレームごとに生成する。ここで、中間目標形状を生成するフレーム間隔が大きすぎた場合、前後の中間目標形状が大きく異なってしまい、形状変化が上手く実行されない。逆に、中間目標形状を生成するフレーム間隔が小さすぎた場合、毎フレーム中間目標形状を生成し電場分布算出を行う必要があるため計算コストが高くなる。また、2.3.4 節で説明する荷電粒子の再配置処理の影響で、荷電粒子分布が均一なものになってしまう。元形状から目標形状への荷電粒子の運動が自然なものとなるフレーム間隔を実験的に求めた結果、本論文では中間目標形状を 5~10 フレームごとに生成する。

2.2.3 中間目標形状の電場分布の算出

オーロラの原因である荷電粒子は、高さ数千 km 付近で薄いシート状構造に従って加速し、エネルギーを持った状態で地球に落下していく。荷電粒子がシート状構造に従って落下する原因は未だ物理的に解明されていないが、落下する際にかかる電圧は数千ボルトと非常に強力なものであり、地球の電位分布に影響を与えていると考えられている [4]。そのため本手法では、中間目標形状の形をしたオーロラが出現する際には、中間目標形状の付近に落下してくる荷電粒子の影響を受けた電位分布が生成されているという仮定のもと、荷電粒子に働く電場分布のモデル化を行った。まず、図 2.2 赤枠で示された 2 次元平面を直交グリッドで

分割する．その後，中間目標形状の近傍にエネルギーを持った荷電粒子を配置する．配置された荷電粒子の影響を考慮して電位分布を算出することで，中間目標形状のオーロラが出現する際の電位分布を生成する．以下は具体的な処理の説明である．

初めに，中間目標形状の近傍に荷電粒子を配置することで，各セル (i, j) に属する荷電粒子数 $N(i, j)$ を得る．中間目標形状との近傍判定は，各制御点を接続情報に従って結んだ線分と荷電粒子との距離が閾値 L_{thr} 以下であることを条件として行う．処理の全体図を図 2.8 に示す．まず，荷電粒子を配置するために，中間目標形状に含まれる制御点の xz 方向それぞれの最小値と最大値を取得し AABB(Axis-Aligned Bounding Box) を生成する (図 2.8 緑破線)．生成された AABB を L_{thr} だけ各方向に拡大したものを，荷電粒子の配置領域として定義する (図 2.8 緑太線)．配置領域内に荷電粒子をランダムに散布する．このとき，荷電粒子の位置が中間目標形状の近傍 (図 2.8 赤色の領域) であれば配置，近傍でなければその点を破棄する．これを繰り返すことで，中間目標形状近傍のみにランダムに粒子をサンプリングできる．

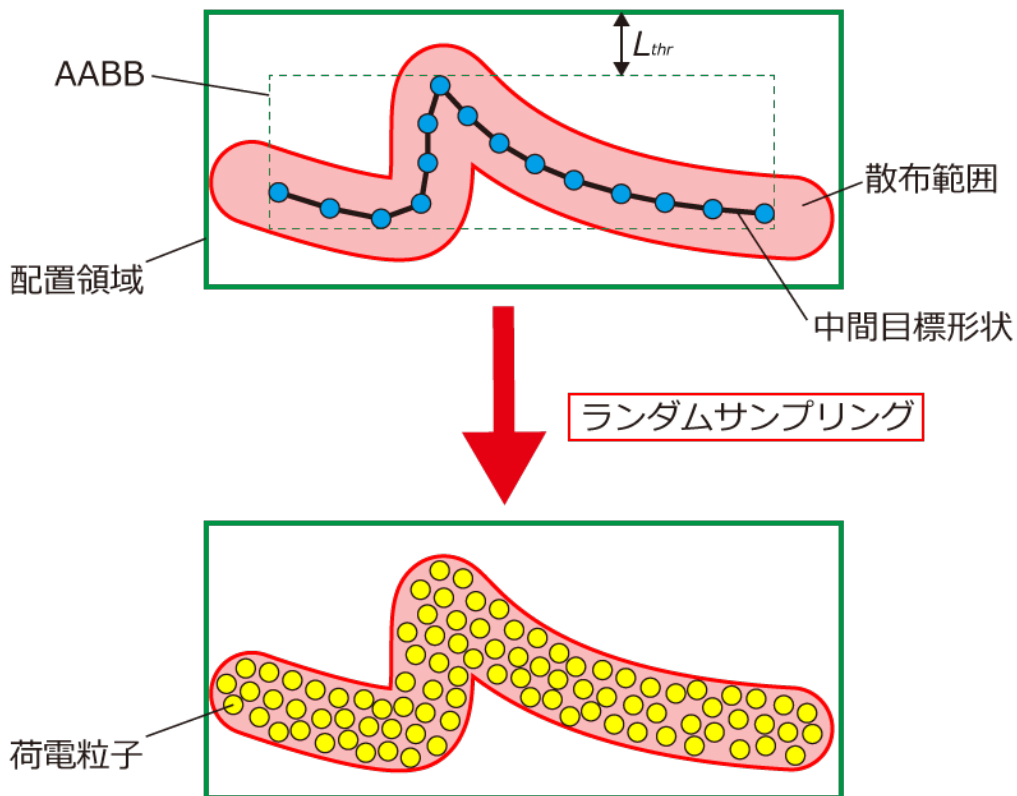


図 2.8: 荷電粒子のランダムサンプリング

図 2.9 は実際に生成した中間目標形状であり，黒色の点が制御点，制御点を結んだ緑色の線分が中間目標形状の概形を表す．図 2.9 の中間目標形状から生成した荷電粒子数の分布を図 2.10 に示す．青い部分が荷電粒子のないセルを表し，赤色に近づくほど荷電粒子数が多いセルを表す．中間目標形状の近傍に，荷電粒子をほぼ均等に配置できたことが観測された．

次に，各セルの電荷密度を求める．電荷密度とは，単位面積当たりの電荷の分布量のことを指す．荷電粒子数 $N(i, j)$ から式 (2.2) を用いて各セルの電荷密度 $\rho(i, j)$ を計算する式を

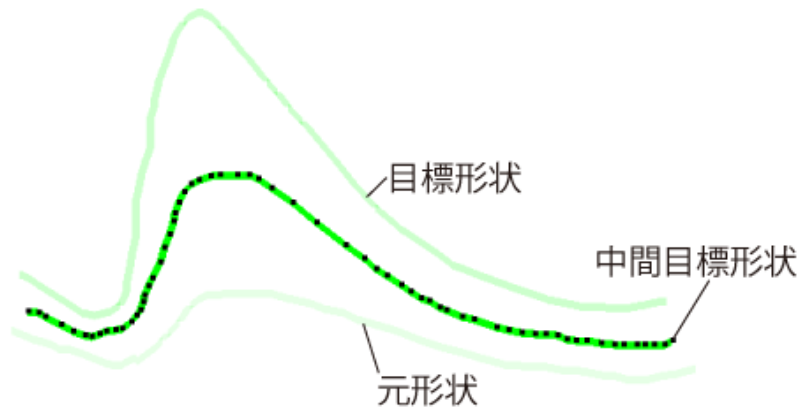


図 2.9: 生成した中間目標形状

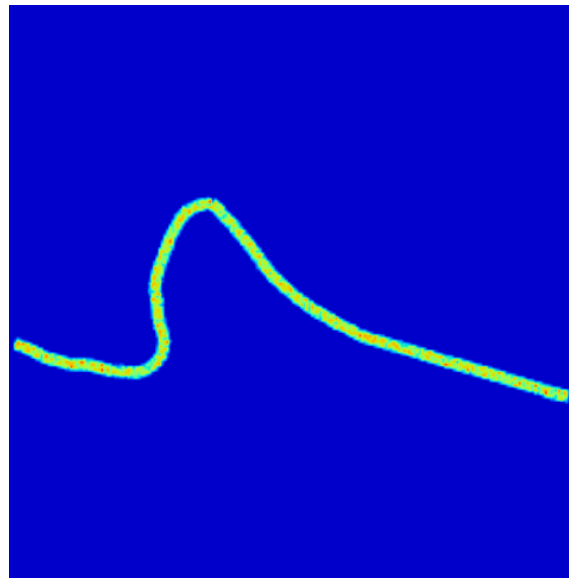


図 2.10: 生成した荷電粒子数分布

次の式 (2.2) に示す.

$$\rho(i, j) = \frac{q_0 N(i, j)}{\Delta S} \quad (2.2)$$

q_0 は電気素量, ΔS は単位面積を表す. 電気素量とは, 陽子あるいは電子がもつ電荷の大きさを表す定数のことを指し, 本手法では 1.6×10^{-19} クーロンを用いた.

次に, 求めた電荷密度 $\rho(i, j)$ から, 各セルの電位 $\phi(i, j)$ を求める. 真空の誘電率を ϵ_0 とすると, 2次元平面上における電位と電荷密度の関係を表すポアソン方程式は次の式 (2.3) で表される [20].

$$\nabla^2 \phi = \frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial z^2} = -\frac{\rho}{\epsilon_0} \quad (2.3)$$

反復法としてガウス・ザイデル法を用いてこのポアソン方程式を解くことで, 各セルの電位 $\phi(i, j)$ を求める. 式 (2.3) の電位の2階微分は, 各セル間の距離を Δd とすると中心差分近似を用いて次の式 (2.4) と (2.5) で表すことができる.

$$\frac{\partial^2 \phi}{\partial x^2} = \frac{\phi(i+1, j) - 2\phi(i, j) + \phi(i-1, j)}{\Delta d^2} \quad (2.4)$$

$$\frac{\partial^2 \phi}{\partial z^2} = \frac{\phi(i, j+1) - 2\phi(i, j) + \phi(i, j-1)}{\Delta d^2} \quad (2.5)$$

式 (2.3), (2.4), (2.5) から, セル (i, j) における電位 $\phi(i, j)$ を次の式 (2.6) で求める.

$$\phi(i, j) = \frac{1}{4} \left\{ -\rho(i, j) \frac{\Delta S}{\epsilon_0} + \gamma(i, j) \right\} \quad (2.6)$$

ここで, $\gamma(i, j)$ はセル (i, j) の4近傍の電位和を表し, $\gamma(i, j) = \phi(i+1, j) + \phi(i-1, j) + \phi(i, j+1) + \phi(i, j-1)$ と表される. この式をガウス・ザイデル法を用いて, 繰り返し回数が一定回数に達するか, 前ステップと比較した誤差の最大値が閾値以下になるまで反復する. 境界条件はディリクレ境界条件として $\phi = 0$ を与えた. ここで, 反復時に電荷密度の反数である $-\rho(i, j)$ を用いることで, 中間目標形状の近傍セルほど電位が低くなるような電位分布が生成される.

また, オーロラの高周波な動きを表現するために, 算出した電位分布にパーリンノイズ [21] を加える. パーリンノイズは炎や煙や雲などの自然物を表現する際によく用いられるノイズである. 単純な乱数に比べて滑らかに変化する自然なノイズを生成でき, 複数周波数のノイズを統合することで, 自然界に見られる自己相似性を再現できるといった特徴がある. パーリンノイズの引数には xz 座標に加えてタイムステップ t を与え, 時間方向にも滑らかに変化させる. パーリンノイズの例を図 2.11 に示す. 青色の部分が値が低く, 赤色の部分が値が高い部分を表す.

最後にセル (i, j) における電場 $E(i, j)$ を求める. 電場 \mathbf{E} は電位 ϕ の勾配を用いて次の式 (2.7) で表すことができる.

$$\mathbf{E} = (E_x, E_z) = \left(-\frac{\partial \phi}{\partial x}, -\frac{\partial \phi}{\partial z} \right) \quad (2.7)$$

対象の格子点の周囲の電位における中心差分近似を用いて, 電場 $\mathbf{E}(i, j)$ の x 成分 $E_x(i, j)$ を式 (2.8), z 成分 E_z を式 (2.9) でそれぞれ求める.

$$E_x = -\frac{\phi(i+1, j) - \phi(i-1, j)}{2\Delta d} \quad (2.8)$$

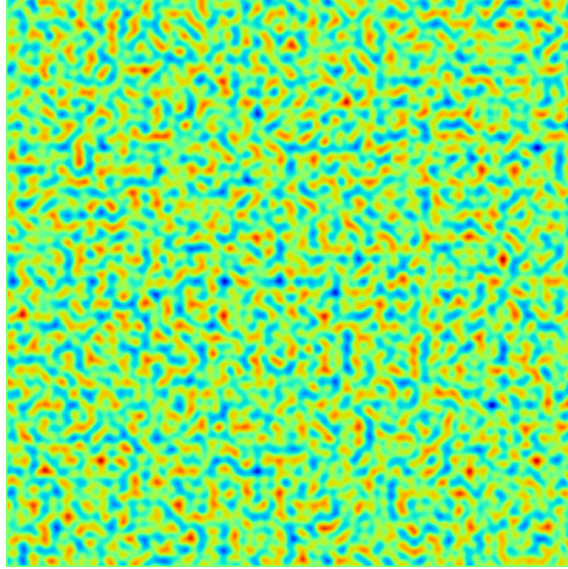


図 2.11: パーリンノイズの例

$$E_z = -\frac{\phi(i, j+1) - \phi(i, j-1)}{2\Delta d} \quad (2.9)$$

中間目標形状の近傍セルほど電位が低い分布となっているため，生成される電場は中間目標形状の近傍に向かって働くような分布となる。

図 2.9 の中間目標形状と，図 2.10 の荷電粒子数分布および図 2.11 のノイズから生成した電位・電場分布を図 2.12 に示す．各画素の色は対応するセルの電位値を表し，青色に近いほど電位が低く，赤色に近いほど電位が高いことを表す．また，白色の矢印は電場を表し， xz 方向それぞれ 10 セルおきに表示している．中間目標形状近傍の電位値が他の部分に比べて低くなっており，中間目標形状に向かって力が働く電場分布を生成できていることが観測された．

2.3 荷電粒子の運動計算処理

2.3.1 荷電粒子に働くローレンツ力の計算

アーティストの指定した形状に近づくような電場分布を生成した後は，それを使って荷電粒子の位置を更新する処理を行う．まず，荷電粒子に働くローレンツ力を計算する．ローレンツ力とは，電磁場の中を運動する荷電粒子にかかる力のことであり，粒子の速度と磁場のクロス積に電場を足したものとなることが分かっている．荷電粒子にかかる電場 \mathbf{E} と， y 軸方向に働く地球の磁場 \mathbf{B} から，ローレンツ力 \mathbf{F}_L は次の式 (2.10) で求めることができる．

$$\mathbf{F}_L = q_0(\mathbf{E} + \mathbf{v} \times \mathbf{B}) \quad (2.10)$$

ここで， \mathbf{v} は荷電粒子の xz 平面上での速度を表す．各荷電粒子にかかる電場 \mathbf{E} は，荷電粒子の属している近傍セルに格納された値を線形補間することで算出する．本手法では荷電粒子の電気素量に正の値を用いているため，電場の方向がローレンツ力の方向と一致する．そ

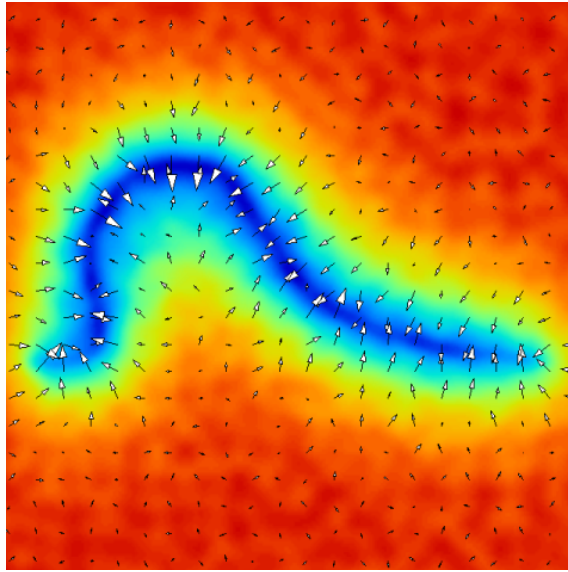


図 2.12: 生成した電位・電場分布

のため、ローレンツ力を荷電粒子を中間目標形状に近づけようとする力として扱うことができる。

2.3.2 荷電粒子間に働くクーロン力の計算

提案手法では、オーロラの高周波な揺らめくような動きを再現するために、中間目標形状近傍に荷電粒子を毎フレームランダムにサンプリングする手法ではなく、中間目標形状の電場分布から求めたローレンツ力を用いて荷電粒子を移流させる。しかし、ローレンツ力のみを考慮して荷電粒子の運動計算を行った場合、荷電粒子は中間目標形状に近づくように動きはするが、オーロラ形状が大きく折れ曲がる部分や、元形状と目標形状の位置の差が小さい制御点付近では、その地点に向かってローレンツ力がかかり続けることで荷電粒子が過度に密集してしまう。この問題に対処するため、提案手法では荷電粒子間の疑似的なクーロン力を導入する。クーロン力とは二つの荷電粒子の間に働く力のことを指し、同符号の電荷を持つ荷電粒子間では斥力として扱うことができる。

クーロン力の計算には、Müllerらの手法[22]に代表されるSPH法の圧力項の計算に基づいた方法を利用する。圧力項は粒子の分布を一様にするために、粒子が過度に近付くと斥力を発生させるため、疑似的なクーロン力として扱うことができる。SPH法は、流体をパーティクルで離散化し、各パーティクルの物理量を近傍パーティクルの重み付き和で近似する手法である。重み付き和を計算する際には、有効半径 h 内だけ値が非ゼロになるカーネル関数と呼ばれる関数を用いることで、自分を中心とした半径 h の範囲内に存在する近傍パーティクルからの影響だけを考慮に入れた物理量を計算する。また、カーネル関数にはパーティクル間の距離が近くなるほど物理量の影響を大きく、遠くなるほど小さくする性質を持った関数がいられる。

まず、各荷電粒子周辺の密度 δ_i を、近傍荷電粒子の重み付き和から求める。荷電粒子 i の位置を \mathbf{x}_i 、質量を m_i 、近傍に存在する荷電粒子の集合を N とすると、密度 δ_i を表す式は

次の式 (2.11) で表される.

$$\delta_i = \sum_{j \in N} m_j W_{poly6}(\mathbf{x}_j - \mathbf{x}_i, h) \quad (2.11)$$

ここで, $W_{poly6}(\mathbf{r}, h)$ は有効半径を h とした Poly6 カーネル関数であり, 式 (2.12) で表される.

$$W_{poly6}(\mathbf{r}, h) = \frac{4}{\pi h^8} \begin{cases} (h^2 - |\mathbf{r}|^2)^3 & 0 \leq |\mathbf{r}| \leq h \\ 0 & otherwise \end{cases} \quad (2.12)$$

その後, Desbrun ら [23] の方法と同様にして, 圧力と密度の状態方程式から導かれた以下の式 (2.13) で圧力値 p_i を求める.

$$p_i = k(\delta_i - \delta_0) \quad (2.13)$$

ここで, 流体のシミュレーションであれば δ_0 は対象の流体の密度を表すが, 今回の手法では中間目標形状の 1 セル当たりの平均密度を用いた. k はガス定数を表す.

最後に各荷電粒子に働くクーロン力 \mathbf{F}_P を次の式 (2.14) で求める.

$$\mathbf{F}_P = \sum_{j \in N} m_j \frac{p_i + p_j}{2\delta_j} \nabla W_{spiky}(\mathbf{x}_j - \mathbf{x}_i, h) \quad (2.14)$$

式 (2.11) で利用した Poly6 カーネル関数は, $\mathbf{r} = 0$ の時に勾配が 0 となってしまう, 完全に同じ座標の荷電粒子が存在した場合, これらの間に働く斥力が 0 になってしまうといった問題が生じる. そのため, 式 (2.14) ではカーネル関数として有効半径を h とした Spiky カーネル関数を用いている. Spiky カーネル関数 $W_{spiky}(\mathbf{r}, h)$ とその勾配 $\nabla W_{spiky}(\mathbf{r}, h)$ は次の式 (2.15), (2.16) で表され, \mathbf{r} が極端に小さくなった場合急激に大きくなるといった特徴により, 粒子同士が過度に集まることを防ぐことができる.

$$W_{spiky}(\mathbf{r}, h) = \frac{10}{\pi h^5} \begin{cases} (h - |\mathbf{r}|)^3 & 0 \leq |\mathbf{r}| \leq h \\ 0 & otherwise \end{cases} \quad (2.15)$$

$$\nabla W_{spiky}(\mathbf{r}, h) = -\frac{30}{\pi h^5} \begin{cases} (h - |\mathbf{r}|)^2 \frac{\mathbf{r}}{|\mathbf{r}|} & 0 \leq |\mathbf{r}| \leq h \\ 0 & otherwise \end{cases} \quad (2.16)$$

また, 式 (2.12) と式 (2.15) で示した 2 つの関数は, Müller らの手法 [22] における密度計算・圧力計算で用いられていたカーネル関数であるが, 他に SPH 法のカーネル関数として Becker ら [24] などが利用している, 式 (2.17) と (2.18) で表される Spline カーネル関数がある.

$$W_{spline}(\mathbf{r}, h) = \frac{10}{7\pi h^2} \begin{cases} 1 - \frac{3}{2}(\frac{|\mathbf{r}|}{h})^2 + \frac{3}{4}(\frac{|\mathbf{r}|}{h})^3 & 0 \leq |\mathbf{r}| \leq h \\ \frac{1}{4}(2 - \frac{|\mathbf{r}|}{h})^3 & h \leq |\mathbf{r}| \leq 2h \\ 0 & otherwise \end{cases} \quad (2.17)$$

$$\nabla W_{spline}(\mathbf{r}, h) = \frac{45}{14\pi h^4} \begin{cases} (\frac{|\mathbf{r}|}{h} - \frac{4}{3})\mathbf{r} & 0 \leq |\mathbf{r}| \leq h \\ -\frac{1}{3}(2 - \frac{|\mathbf{r}|}{h})^2 \frac{h}{|\mathbf{r}|}\mathbf{r} & h \leq |\mathbf{r}| \leq 2h \\ 0 & otherwise \end{cases} \quad (2.18)$$

本手法では、Poly6 カーネル関数と Spiky カーネル関数、もしくは Spline カーネル関数のどちらを使用するかをアーティストに設定してもらう。実際に2つの手法でシミュレーションを行ってみた結果、Poly6 カーネル関数と Spiky カーネル関数を用いた手法の方が、Spline カーネル関数を用いた手法に比べて距離の近い荷電粒子間で働く斥力の影響が強くなり、より分散したようなレンダリング結果が得られた。図 2.13 で表されるように、オーロラではレイと呼ばれる磁気に沿った縞模様が観測される [19]。従って、レイが鮮明に観測できるようなオーロラを表現したい場合には Spline カーネル関数を、レイの印象をよりぼやけさせたオーロラを表現したい場合には Poly6 カーネル関数と Spiky カーネル関数を用いるとよい。



図 2.13: 縞模様のレイを含んだオーロラ [6]

2.3.3 荷電粒子の位置の更新

荷電粒子に働くローレンツ力とクーロン力を用いて、位置の更新と再配置を行う。荷電粒子 i の位置 \mathbf{x}_i の更新は、ニュートンの運動方程式を用いて次の式 (2.19) のように求める。

$$\mathbf{x}_i^{t+1} = \mathbf{x}_i^t + \mathbf{v}_i^t \Delta t + \frac{1}{2} \frac{w_L \mathbf{F}_L + w_P \mathbf{F}_P}{m_i} \Delta t^2 \quad (2.19)$$

ここで、 w_L 、 w_P はそれぞれローレンツ力とクーロン力にかかる重み係数であり、本論文では実験により求めた $w_L = 0.6$ 、 $w_P = 0.4$ を用いた。同様に荷電粒子 i の速度 \mathbf{v}_i を次の式 (2.20) を用いて更新する。

$$\mathbf{v}_i^{t+1} = \mathbf{v}_i^t + \frac{w_L \mathbf{F}_L + w_P \mathbf{F}_P}{m_i} \Delta t \quad (2.20)$$

その後、荷電粒子が加速を続けて発散してしまうのを防ぐために、エネルギー保存則を考慮して速度の正規化を行う。

2.3.4 荷電粒子の再配置

位置の更新を行った後は、荷電粒子の過度な密集や目標形状からの離反を防ぐために、荷電粒子を追加・削除することで再配置処理を行う。再配置処理は中間目標形状を新しいものに更新する際に行われる。荷電粒子が更新前の中間目標形状の近傍に属しているかと、荷電粒子が属しているセルに含まれる荷電粒子数が過度に多いまたは少ないかを判定し、荷電粒子の追加・削除処理を行う。

まず、各セルが中間目標形状の近傍であるかどうかを判定する。近傍判定では、2.2.3 節で電荷密度を取得する際に配置した荷電粒子数 $N(i, j)$ が閾値以上であることと、そこから生成した電位 $\phi(i, j)$ が閾値以下であることを条件とする。近傍でない判定されたセル内にある荷電粒子は、中間目標形状から大きく離反していると判断し削除する。次に、近傍であると判定された各セルに所属している荷電粒子数が多過ぎるまたは少なすぎるかの判定を行い、多すぎる場合は荷電粒子を削除、少なすぎる場合は荷電粒子を追加する。この時、判定に用いる閾値は過多・過少それぞれに対してアーティストが設定する。荷電粒子を削除する際には、なるべく中間目標形状に近い荷電粒子を残すために、セル内の荷電粒子を中間目標形状との距離順にソートし、遠いものから順番に削除していく。荷電粒子を追加する時の位置は、配置するセル内部でランダムに決定し、周囲の荷電粒子の速度の平均値を初速度として与える。追加・削除する荷電粒子の数は、アーティストが設定した閾値との差を取って決定するが、一度の再配置処理で荷電粒子数を極端に変化させてしまった場合、オーロラ形状の不連続な変化が観測されてしまう可能性があるため、1セル当たり一度に追加・削除することのできる荷電粒子数の上限を予め設定しておく。

荷電粒子の再配置処理の有無による荷電粒子数分布の比較を図 2.14 に示す。青色の領域は荷電粒子が存在しないセルを表し、赤色に近づくほど荷電粒子が多く分布しているセルを表す。また、図 2.15 は実際にマウスドラッグ操作によってアーティストが設定した目標形状を表し、黒色の点が制御点、制御点を結ぶ緑色の線分がオーロラの概形を表す。再配置処理有りの場合は目標形状近傍に荷電粒子が分布し、大きく偏った部分は観測されなかった。一方で再配置処理無しの場合では、荷電粒子が過度に集中してしまう部分が発生したことに加え、目標形状近傍でも荷電粒子がほとんど存在しないセルが存在してしまい、レンダリングした結果が実際のオーロラとかけ離れたものとなってしまった。また、目標形状から離反してしまった荷電粒子が存在し続けていることも観測された。

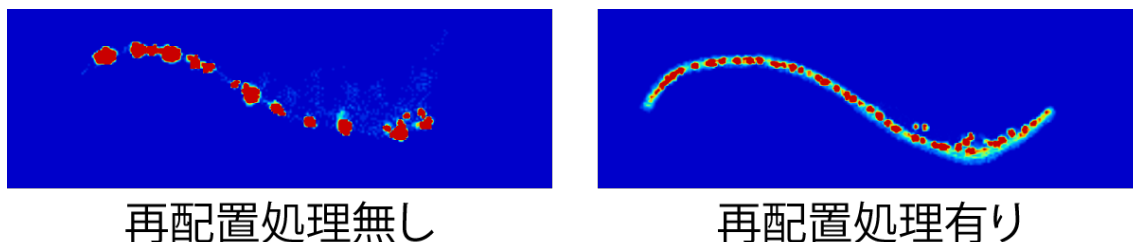


図 2.14: 再配置処理の有無による荷電粒子数分布の比較



図 2.15: 再配置処理比較の際に設定した目標形状

2.4 レンダリング

最後に、更新した荷電粒子の位置から y 軸方向への落下シミュレーションを行い、最終的なレンダリング結果を得る。レンダリングにはスクリーン空間を用いて落下処理を行う手法 [25] を用いる。

この方法ではオーロラのレンダリング処理で最も計算コストが大きい発光点の生成と投影を並列化するために、まず前計算として3次元空間上での落下処理を行い、荷電粒子の落下が終了する位置を相対置として算出しておく。その後、3次元空間上での落下開始位置と終了位置を発光点とみなし、光の届く範囲を半径とした発光粒子を生成する。この発光粒子をスクリーン空間上に投影することで、スクリーン空間上での落下開始位置と終了位置が求められる。これを基に、スクリーン空間上での荷電粒子の落下方向と、微小タイムステップ幅 $\Delta\tilde{t}$ ごとに落下する距離を求める。求めた落下方向と落下距離から、毎フレームスクリーン空間での落下シミュレーションを行い、発光地点を算出する。算出した発光地点から、米山らの手法 [8] を用いて大気粒子の発光シミュレーションを行い、最終的なカラーマップを生成する。

2.4.1 荷電粒子の落下シミュレーション

前計算として3次元のシミュレーション空間上での疑似的な落下処理を行い、荷電粒子が最終的に到達する落下終了位置を相対値として算出する。荷電粒子は、大気粒子との衝突を繰り返しがて停止するが、最終的に落下する距離は落下開始時に持つ初期エネルギーに依存する。荷電粒子が1度の衝突で失うエネルギーは平均約 100eV であるため [19]、大気粒子と衝突するたびに荷電粒子が持つエネルギーを減算していき、エネルギーが尽きた地点を落下終了位置とする。

荷電粒子の落下シミュレーションでは微小タイムステップ幅 $\Delta\tilde{t}$ を単位時間として用いる。荷電粒子の落下計算は、荷電粒子のエネルギーが尽きるまで1フレームあたりに何度も行われるため、微小タイムステップ幅 $\Delta\tilde{t}$ と、2.2節の荷電粒子の運動計算で用いたタイムステップ幅 Δt の関係は $\Delta\tilde{t} \ll \Delta t$ とする。まず、荷電粒子の $\Delta\tilde{t}$ あたりの落下ベクトル \mathbf{L}_{sim} を次の式 (2.21) で求める。

$$\mathbf{L}_{sim} = v_f \frac{\mathbf{B} + \boldsymbol{\omega}_{sim}}{|\mathbf{B} + \boldsymbol{\omega}_{sim}|} \Delta\tilde{t} \quad (2.21)$$

ここで、 v_f は荷電粒子が地球の磁場方向に落下する速さ、ベクトル $\boldsymbol{\omega}_{sim}$ は乱数によって生成した荷電粒子と大気粒子の衝突時に生じる微小なブレを表す。

その後、 \mathbf{L}_{sim} を用いて荷電粒子の位置を更新していき、更新した位置を衝突判定位置として大気粒子と衝突するかの判定を行う。衝突確率 P_c は、高層大気の密度分布 [26] を用いて次の式 (2.22) から求める。

$$P_c = n\pi r^2 |\mathbf{L}_{sim}| \quad (2.22)$$

n は高層大気の密度分布から求めた単位体積あたりの大気粒子数密度、 r は大気粒子の半径を表す。生成した乱数 R_c と衝突確率 P_c を比較することで、大気粒子との衝突判定を行う。 $R_c > P_c$ ならば衝突しないと判定し、再び \mathbf{L}_{sim} を用いて荷電粒子の衝突判定位置を更新する。 $R_c \leq P_c$ ならば大気粒子と衝突したと判定し、荷電粒子が持つエネルギーを減算する。

衝突によって荷電粒子の落下方向が変わるため、 ω_{sim} を再び乱数生成し、式 (2.21) を用いて L_{sim} を更新する。上記の処理を、荷電粒子の持つエネルギーが 0 になるまで行い、エネルギーが尽きた地点を荷電粒子の落下終了位置とし、各荷電粒子の落下開始位置から相対位置として格納しておく。

2.3.3 節で計算した荷電粒子の落下開始位置と、前計算で算出しておいた落下終了位置から、スクリーン空間上での落下計算を毎フレーム行う。処理の模式図を図 2.16 に示す。ま

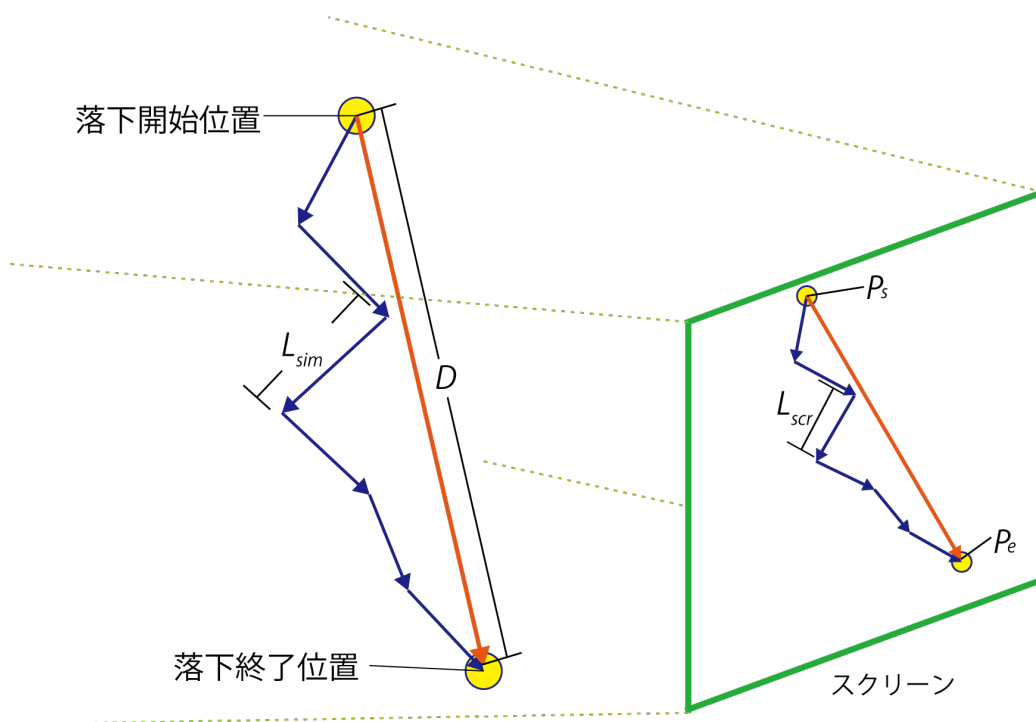


図 2.16: スクリーン空間上での落下シミュレーションの模式図

ず、シミュレーション空間上での落下開始位置と落下終了位置を発光点とみなし、発光の届く範囲を半径とした発光粒子を生成する。この粒子をスクリーン空間上に投影し、スクリーン空間上における落下開始位置 P_s と落下終了位置 P_e を得る (図 2.16 黄色の円)。その後、スクリーン空間上での微小タイムステップ幅 Δt あたりの落下ベクトル L_{scr} を、シミュレーション空間とスクリーン空間の落下距離の比から式 (2.23) を用いて求める。

$$L_{scr} = \frac{|P_e - P_s|}{D} L_{sim} \quad (2.23)$$

ここで D はシミュレーション空間上での荷電粒子の最終的な落下距離を表す。

次に、シミュレーション空間上での落下処理と同様に荷電粒子の衝突判定位置を更新し、荷電粒子との衝突判定を行う。現在の衝突判定位置を P_0 とすると、次の衝突判定位置 P は次の式 (2.24) で表すことができる。

$$P = P_0 + \frac{L_{scr} + \omega_{scr}}{L_{scr} + \omega_{scr}} \quad (2.24)$$

ここで、 ω_{scr} は大気粒子と衝突したことによるスクリーン空間上でのぶれを表すベクトルであり、シミュレーション空間上での衝突判定と同様、衝突が発生するたびに乱数を用いて生成する。式 (2.23) で求めた衝突判定位置と、式 (2.22) で求めた衝突確率 P_c を用いて、シミュレーション空間上での落下処理と同様に大気粒子との落下判定を行う。荷電粒子の持つエネルギーが 0 になるまで繰り返し、全ての衝突点を発光点として取得する。

2.4.2 大気粒子の発光シミュレーション

荷電粒子と大気粒子との衝突地点における、大気粒子の発光シミュレーションを行う。高層大気中に含まれる主要な大気粒子は、酸素原子 O 及び窒素分子 N_2 であり、これらの粒子が荷電粒子との衝突によってエネルギーを持った励起状態となり、オーロラの原因となる光を発する。本手法では酸素原子 O か窒素分子 N_2 との衝突によって生じる光のみを利用し、他の大気粒子との衝突によって生じる光は考慮しないこととする。また、衝突によって生じる光の波長は原子ごとに複数種類存在し、酸素原子と衝突した場合は 557.7nm と 630.0nm、窒素分子と衝突した場合は 391.4nm, 427.8nm, 670.5nm の波長が主に放出される。このため発光シミュレーションでは、まず酸素原子、窒素分子、それ以外の分子のうちどの大気粒子と衝突するかを、高層大気密度分布 [26] から決定する。次に、衝突した大気粒子が酸素原子なら 2 種類、窒素分子なら 3 種類の中から、どの波長の光を発するかを波長ごとの生成率 [27] を用いて決定する。酸素原子と窒素分子以外の大気粒子と衝突したと判定された場合には、その地点では発光が起こらないものとして扱った。

荷電粒子との衝突によって励起状態となった大気粒子は、そのエネルギーを放出して発光するまでに一定の時間を要する。その合間に他の大気粒子と衝突することで、再びエネルギーの遷移が起こってしまい、光を放射することができなくなるといった現象が観測される。このような現象は消光現象と呼ばれ、オーロラの色を決定するために考慮に入れる必要がある。荷電粒子が衝突してから大気粒子が発光するまでにかかる時間を τ とすると、消光確率 P_v は次の式 (2.25) で表される。

$$P_v = 1 - \exp(-\sqrt{2}n\pi r^2 v_a \tau) \quad (2.25)$$

v_a は大気粒子の速さを表し、マクスウェル分布に従い次の式 (2.26) で求める。

$$v_a = \sqrt{\frac{8k_b T}{\pi m_a}} \quad (2.26)$$

k_b はボルツマン定数、 T は高層大気温度、 m_a は大気粒子の質量を表す。

また、大気粒子が発する光は視点に届くまでの間に減衰するため、衝突点と視点が遠いほど暗く、近いほど明るく観測される。光の減衰率 c は様々な形式で表されるが、本手法では式 (2.27) に示される平方指数式を用いた。

$$c = \exp(-(dz_a)^2) \quad (2.27)$$

d は減衰係数を表す。 z_a はスクリーン空間上での発光点の z 座標を表し、落下開始位置と落下終了位置の z 座標を線形補間することで求めた。

2.4.3 カラーマップ生成

各発光地点における波長の種類と輝度値を用いて、最終的なカラーマップ（出力画像）を生成する。発光の届く範囲を半径、発光点を中心とした発光粒子を生成し、その粒子の半径内に存在する画素に輝度値を積算していく。このとき、発光の届く範囲は各波長ごとに予め設定しておく。輝度値の積算を行う際には、ガウシアンフィルタを用いることで発光点に近い画素ほど積算される輝度値が大きく、遠い画素ほど小さくなるようにする。標準偏差を σ 、発光点と画素中心との距離を b としたガウシアンフィルタ関数は次の式 (2.28) で表される。

$$G(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{b^2}{2\sigma^2}\right) \quad (2.28)$$

ガウシアンフィルタ関数の計算を各粒子及び各画素ごとに行うのは計算コストが高いため、発光粒子の中心と画素中心との距離をキーとしたテーブル化を行った。

全ての発光点における輝度値を格納し終わったら、各画素の波長と輝度値を CIE-XYZ 表色系の三刺激値 X, Y, Z に変換する。光の波長を λ 、波長 λ の輝度値を $L(\lambda)$ とすると、三刺激値 X, Y, Z は以下の式 (2.29) で求めることができる [28]。

$$\begin{aligned} X &= K \int_{380}^{780} \bar{x}(\lambda)L(\lambda)d\lambda \\ Y &= K \int_{380}^{780} \bar{y}(\lambda)L(\lambda)d\lambda \\ Z &= K \int_{380}^{780} \bar{z}(\lambda)L(\lambda)d\lambda \end{aligned} \quad (2.29)$$

K は視感度の最大となる光の波長であり、683lm/W であるとされている。 \bar{x} 、 \bar{y} 、 \bar{z} は XYZ 表色系における等色関数の値であり、本論文では表 2.1 を用いた。最後に、式 (2.29) で求め

表 2.1: 等色関数 [29]

波長 [nm]	\bar{x}	\bar{y}	\bar{z}
557.7	0.5739	0.9993	0.0018
630.0	0.6924	0.2981	0.0000
391.4	0.0049	0.0005	0.0239
427.8	0.2736	0.0299	1.4463
670.5	0.0893	0.0345	0.0000

た X, Y, Z の値を、式 (2.30) を用いて RGB 値に変換する [30]。

$$\begin{pmatrix} R \\ G \\ B \end{pmatrix} = \begin{pmatrix} 3.24097 & -1.53738 & -0.498612 \\ -0.969241 & 1.87596 & 0.0415554 \\ 0.0556299 & -0.203976 & 1.05697 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (2.30)$$

2.5 GPU による並列化

本手法では、荷電粒子にかかるクーロン力の計算と再配置処理、レンダリング処理において、NVIDIA CUDA(Compute Unified Device Architecture)[31] を用い GPU による並列化

を行った。GPUではマルチコアCPUと異なり、数千ものシンプルなコアを用いてマルチスレッドで処理を行うため、スレッド分割数によって計算速度が大きく変化する。メモリレイテンシをカバーするためにも、一般的にスレッド数は多い方がよい。また、GPUによる並列化を行うには、必要なデータをCPU側から転送する必要があり、提案手法の場合は、荷電粒子の位置や大気粒子との衝突位置などがそれに当たる。逆に、GPUによって計算された値をCPUで扱うために、GPUからCPUへのデータ転送も必要となる。これらの転送は非常に時間のかかる処理であり、可能な限り転送回数を少なくすることが望ましい。提案手法では、クーロン力の計算と再配置処理、レンダリング処理のそれぞれに対して適した並列化手法を採用することで、効果的な高速化を行った。

2.5.1 荷電粒子間に働くクーロン力の計算処理の並列化

荷電粒子にかかるクーロン力の計算では、SPH法 [22] における圧力項の計算手法を用いて、有効半径 h 以内にある近傍荷電粒子の密度や圧力といった物理量の重み付き和を計算することで、自身の物理量を求めている。この近傍粒子の探索を単純に全ての荷電粒子に対して行った場合、粒子数 N のシミュレーションでは計算量が $O(N^2)$ と膨大になってしまう。そこで、近傍探索の高速化のために空間分割法 [32] を利用する。空間分割法は、探索空間をグリッドセルで分割し、各粒子が属しているセル番号を格納しておき、近傍探索の際に近傍セルに所属している粒子との探索のみを行うことで、近傍探索を高速化する手法である。例として等間隔グリッド分割した平面と、そこに配置した荷電粒子を図 2.17 に示す。黄色の円が荷電粒子を表し、内部に表示された数字が各荷電粒子の番号を表す。また、各セル左下に表示されている灰色の数字は各セルの番号を表す。

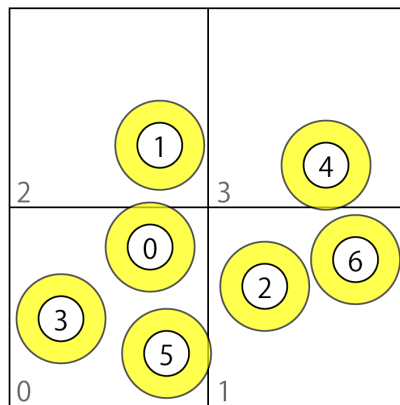


図 2.17: 空間分割法におけるグリッドと配置された荷電粒子の例

空間分割法ではまず、各荷電粒子の番号と、その荷電粒子が所属しているセル番号を格納した配列を作成する。実際に図 2.17 から作成した配列は次の表 2.2 のようになる。ここで、荷電粒子番号を格納した配列を ParticleIndex, セル番号を格納した配列を CellIndex と定義する。

その後、GPUによる並列化のため、CellIndexの値をキーとした基数ソートを行う。基数

表 2.2: 配列に格納した荷電粒子番号とセル番号

ParticleIndex	0	1	2	3	4	5	6
CellIndex	0	2	1	0	3	0	1

ソート後の配列を表 2.3 に示す。また、各セルに所属している荷電粒子番号の開始位置と終了位置を格納した配列 CellStart と CellEnd を、表 2.4 のように作成する。

表 2.3: ソート後の荷電粒子番号とセル番号

ParticleIndex	0	3	5	2	6	1	4
CellIndex	0	0	0	1	1	2	3

表 2.4: 荷電粒子番号の開始位置と終了位置

CellStart	0	3	5	6
CellEnd	2	4	5	6

クーロン力の計算では、有効半径 h 以内にある荷電粒子の影響のみを考慮に入れている。本手法では分割セルの幅が有効半径 h と等しくなるようにしているため、対象としている荷電粒子が所属しているセルの 8 近傍セルに所属している荷電粒子とのみ近傍判定を行えば、全ての荷電粒子と近傍探索を行った場合と同じ結果を得られる。近傍であると判定された荷電粒子の重み付き和を計算することで、自身の物理量を得ることができる。

上記の空間分割法によるクーロン力計算の流れを図 2.18 に示す。まず、CPU で各荷電粒子が所属しているセル番号を取得し、CellIndex と ParticleIndex を作成する。この処理は、2.2.3 節で各セルの荷電粒子数 $N(i, j)$ を求める際に同時に行っておく。その後、2次元平面を分割したグリッド数で GPU をスレッド分割し、CellIndex と ParticleIndex を転送して基数ソートを行う。GPU での基数ソートのアルゴリズムは Satish ら [33] の手法を用いた。次に、ソートされた配列を基に CellStart と CellEnd を CPU で作成する。生成した CellStart と CellEnd, ParticleIndex を、荷電粒子数でスレッド分割した GPU に転送する。GPU では、CellStart と CellEnd を基に近傍セルに所属している荷電粒子のみと近傍探索を行い、近傍であった場合は重み付き和から密度を計算する。最後に、求めた密度から最終的に荷電粒子にかかるクーロン力を算出する。

2.5.2 荷電粒子の再配置処理の並列化

再配置処理では、荷電粒子の削除を行う際に、中間目標形状から遠い荷電粒子から順に削除するため、荷電粒子を中間目標形状との距離順にソートする必要がある。しかし、再配置処理は中間目標形状の近傍セルのみを対象とした処理であるため、全てのセルの情報を GPU に転送し再配置処理を行うと、データの転送コストや近傍以外のセルでの再配置処理時間が余分にかかってしまう。そのため、近傍セルの情報のみを詰めた配列を作成し、これを処理に用いる。このとき問題となるのは GPU の各スレッドがこの配列のどこにアクセスすべきかがわからないことである。これを解決するために、近傍セルで 1、それ以外で 0 を

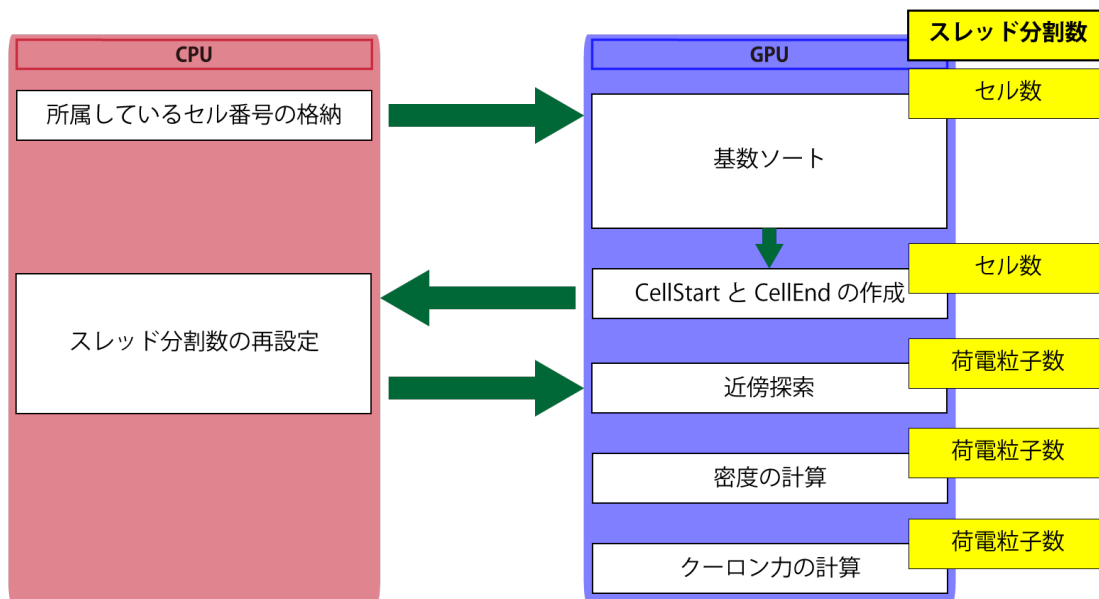


図 2.18: クーロン力計算の並列化の流れ

格納した配列を作成し，その Prefix Sum(Scan) を計算することで，全てのセルから近傍セルのみを抽出した配列内の対応する位置を格納した配列を得る．この配列を CellScanned と定義する．作成した CellScanned を用いて，近傍セルのみで荷電粒子の再配置処理を行う．

荷電粒子の再配置処理を実際に GPU で並列化した処理の流れを図 2.19 に示す．まず，GPU を全セル数でスレッド分割し，各セルの荷電粒子数と電位値を転送する．その後，荷電粒子と電位値を用いて各セルごとに GPU での近傍判定を行う．ここで，近傍のセルであれば 1，近傍でなければ 0 を格納した Scan 配列を作成する．生成した Scan 配列に対して Scan 処理を行う．Scan 処理は，元の配列を $\boldsymbol{x} = \{x_0, x_1, \dots, x_n\}$ とすると， $s_i = \sum_{j=0}^i x_j$ で表される配列 $\boldsymbol{s} = \{s_0, s_1, \dots, s_n\}$ を新しく生成する処理である．Scan 処理によって生成された配列 \boldsymbol{s} は，その要素が何番目の近傍セルであるかを表すため，これを用いることで近傍セル番号だけを格納した配列を使って処理できる．Scan 処理の全体図を図 2.20 に示す．

次に，GPU を近傍セル数でスレッド分割し (近傍セル数は CellScanned の最後の要素参照することで得られる)，近傍セルの配列上の位置だけを格納した配列 CellScanned と荷電粒子の位置を格納した配列を転送する．GPU では，近傍セルで削除・追加が必要かの判定を行う．削除が必要であると判定されたセルでは，セル内にある荷電粒子を中間目標形状との距離順にソートする．ここで，ソートアルゴリズムには挿入ソートを用いた．挿入ソートは，予めある程度ソートされている配列を対象とした場合は高速にソートを行える．荷電粒子を格納した配列は，1 ステップ前の再配置処理におけるソート結果から大きく順番が変更されていないと考え，この手法を採用した．最後に，近傍セルにおいて荷電粒子の削除・再配置処理を行う．

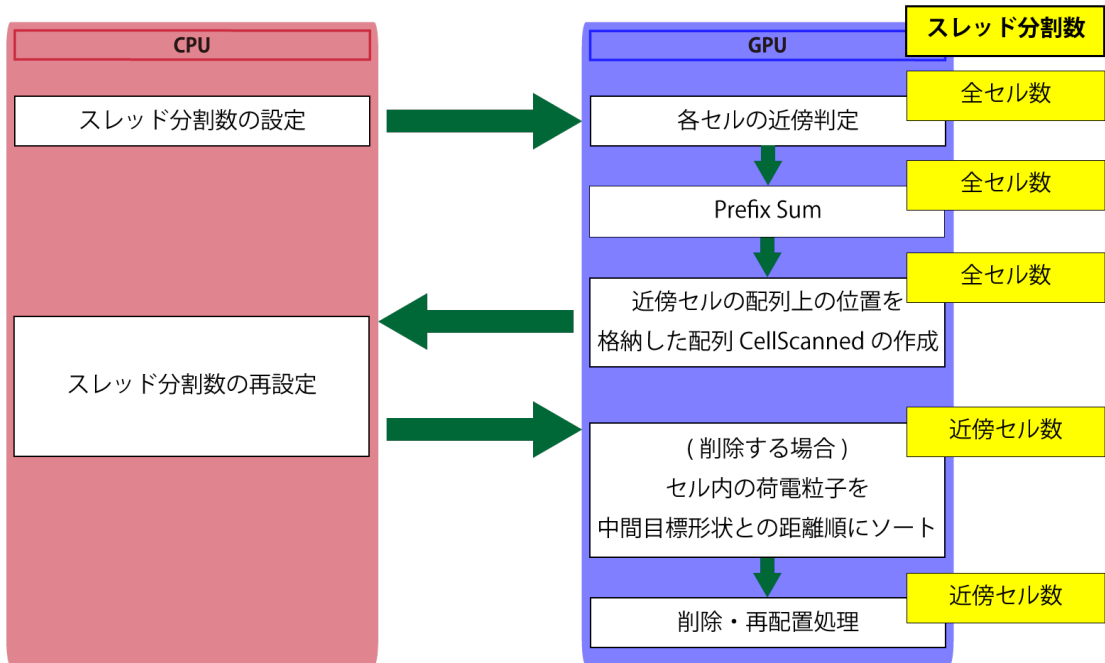


図 2.19: 荷電粒子の再配置処理の並列化の流れ

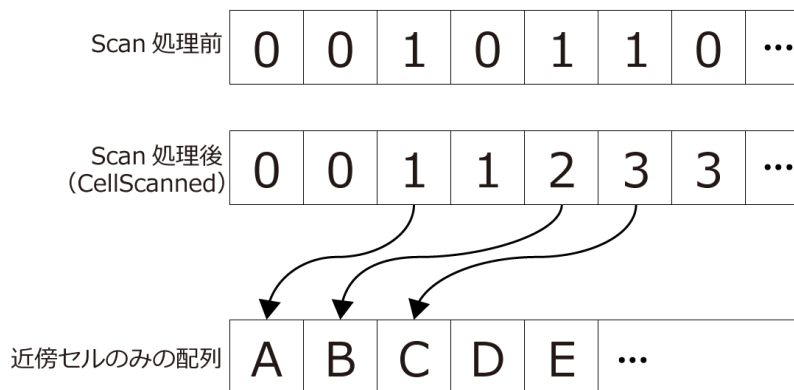


図 2.20: スキャン処理

2.5.3 レンダリング処理の並列化

レンダリング処理では、スクリーン空間上での荷電粒子の落下シミュレーション、大気粒子の発光シミュレーションによる輝度値の積算、最終的なカラーマップ生成の3つの処理を異なるスレッド分割数で並列化する。レンダリング処理の流れを図 2.21 に示す。まず、GPU を荷電粒子数でスレッド分割し、荷電粒子の落下開始位置を GPU に転送する。落下シミュレーションをエネルギーが尽きるまで行い、算出された大気粒子との衝突位置を CPU へ転送する。次に、GPU を衝突位置の数でスレッド分割し、算出した大気粒子との衝突位置を GPU へ転送する。GPU では算出された衝突位置における大気粒子の発光シミュレーションを行い、発光粒子の半径内にある画素に輝度値の積算を行う。輝度値の積算を行う際には、あるスレッドが処理を行っている最中に、別のスレッドが同じ画素にアクセスしてしまう可能性がある。この問題は、アトミック関数を用いることで解決した。アトミック関数とは、あるスレッドがグローバルメモリやシェアードメモリ上のデータの読み込みや書き込みを行う際に、他のスレッドがそのメモリ領域への書き込みを行わないことを保証する関数である。最後に、スレッドをカラーマップの画素数で再分割し、GPU で輝度値から RGB 値への変換を行い、最終的なカラーマップを生成する。

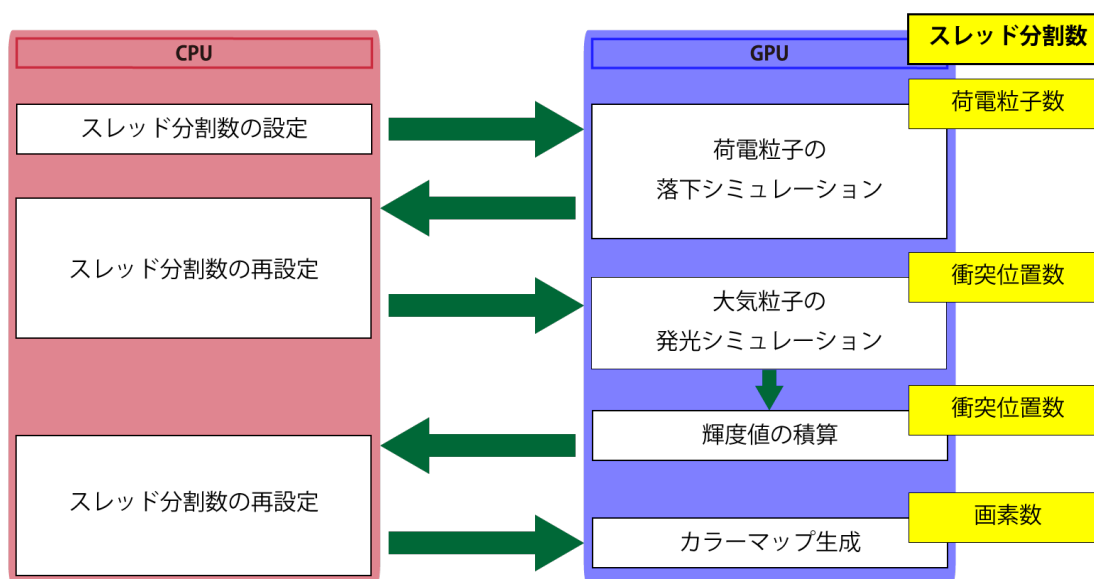


図 2.21: レンダリング処理の並列化の流れ

第 3 章

結果と考察

本論文の手法を実装し，実験を行った．実装・実験環境を以下の表 3.1 に示す．

表 3.1: 実装・実験環境

CPU	Intel Core i7-6700K
Memory	16GB
GPU	GeForce GTX 1080
OS	Microsoft Windows 10 Pro
開発言語	C++ , C for CUDA
グラフィックス API	OpenGL

3.1 編集画面

提案手法で実装したオーロラ形状の編集画面を図 3.1 に示す．画面左側が元形状の編集領域，右側が目標形状の編集領域を表す．編集画面では，アーティストがマウスをクリックしてから離すまでの軌跡を取得し，制御点とそれを結ぶ線分で離散化することでオーロラ概形を取得する．元形状編集領域における黒色の点が制御点，それを結ぶ緑色の線分がオーロラ概形を表す．アーティストがマウスドラッグ操作を複数行う事で，複数本のシート状構造を指定することができる．図 3.1 の例では，一つのシート状構造当たり 32 個と設定している．目標形状編集領域には，変形前の形状との相対位置がわかりやすくなるよう，半透明で元形状のオーロラ概形を描画している．半透明の黄色の線分が次に編集するシート状構造と対応した部分を表し，緑色の線分がそれ以外の部分を表している．また，分断が発生する場合には分断させたい線分を，結合が発生する場合には結合させたい 2 つの制御点を選択する．分断の様子を図 3.2，結合の様子を図 3.3 に示す．

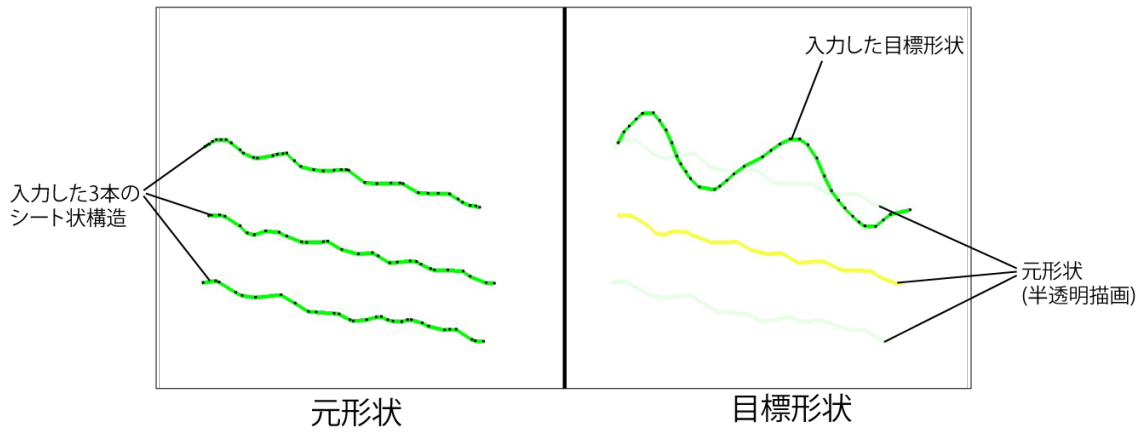


図 3.1: オーロラ形状の編集画面

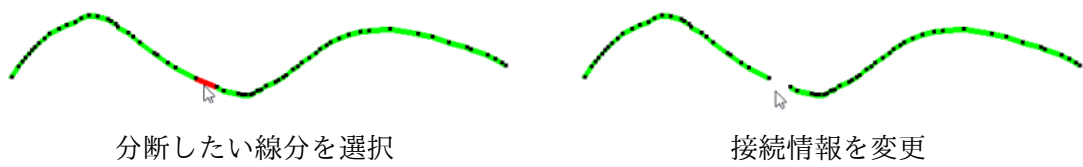


図 3.2: 分断の設定

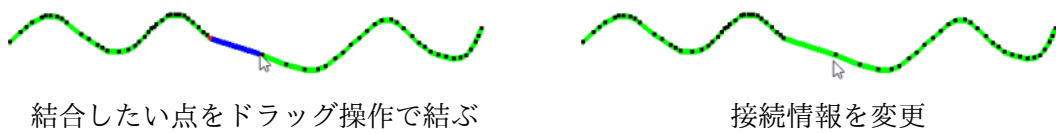


図 3.3: 結合の設定

3.2 形状制御検証

提案手法を用いてアーティストが設定した形状からオーロラのアニメーションを生成し、形状制御の検証を行った。全てのシミュレーションで共通して用いたパラメータの値を表 3.2 に示す。カーネル関数はレイ構造を表現するために Spline カーネル関数を用いた。

表 3.2: 実験に用いた各パラメータ

生成する画像サイズ	1280 × 720 [px]
グリッド分割数	256 × 256
初期配置する荷電粒子数	12 [万個]
各荷電粒子の大気粒子との衝突回数	100 [回]
各形状の遷移に要するフレーム数	140 [frame]
タイムステップ Δt	0.0333 [s]
微小タイムステップ $\Delta \tilde{t}$	0.0033 [s]

また、再配置処理に関する閾値は、実験で求めた表 3.3 の値を用いた。ここで、全てのセルの電位の平均値を ϕ_α 、荷電粒子数の平均値を N_α とした。

表 3.3: 再配置処理に関する各パラメータ

近傍であると判定する電位の閾値	$0.25 \times \phi_\alpha$ [V]
近傍であると判定する荷電粒子数の閾値	$0.10 \times N_\alpha$ [個]
追加判定に用いる閾値	50 [個]
1 グリッド追加する荷電粒子数の最大値	10 [個]
削除判定に用いる閾値	500 [個]
1 グリッドで削除する荷電粒子数の最大値	100 [個]

表 3.2, 3.3 のパラメータは最終的なアニメーションの質に影響するもので、形状制御には影響しない。

シーン 1

形状遷移と再配置処理が正確に行えているかの検証として、1 本のシート状構造を持つオーロラの右端が折り返す、ピーリングと呼ばれる現象のアニメーションを生成した。指定したオーロラ形状を図 3.4, アニメーションの生成結果を図 3.9 に示す。各図のキャプションに示した数値 f はシミュレーションのフレーム数を示す。1 フレーム当たりの生成時間は約 2.5~3.3 秒であった。

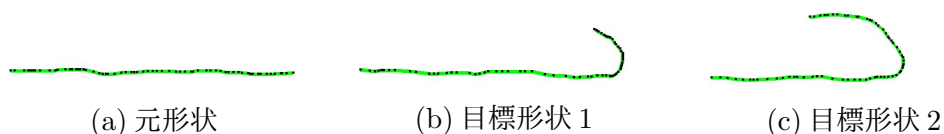


図 3.4: シーン 1 で指定したオーロラ形状

シーン 2

分断現象の検証として、1本のシート状オーロラがS字状に変形し、中心部分から分断することで2本のオーロラを形成するアニメーションを生成した。指定したオーロラ形状を図3.5、アニメーションの生成結果を図3.10に示す。1フレーム当たりの生成時間は約2.5~3.7秒であった。

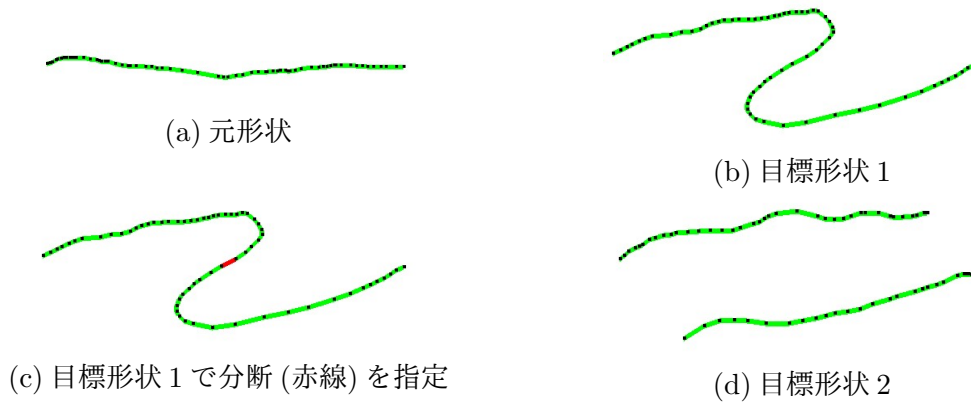


図 3.5: シーン 2 で指定したオーロラ形状

シーン 3

結合現象の検討として、3本のシート状オーロラのうち、2本のオーロラが接近し結合することで長いオーロラと短いオーロラを形成するアニメーションを生成した。指定したオーロラ形状を図3.6、アニメーションの生成結果を図3.11に示す。1フレーム当たりの生成時間は約2.9~3.9秒であった。

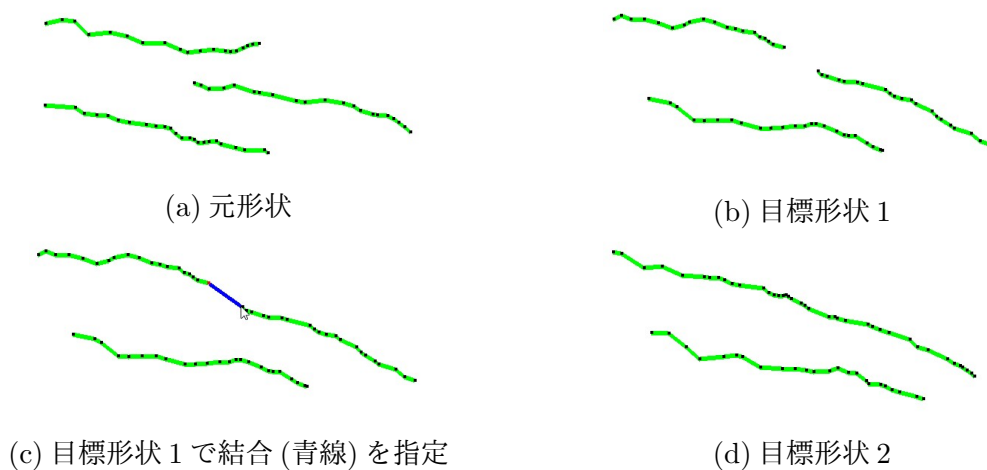


図 3.6: シーン 3 で指定したオーロラ形状

シーン 4

ひだの発生・回転運動，分断現象，結合現象を複合した形状遷移の検証としてシーン 4 を設定した．図 1.3 で示すように 1 本のオーロラから 2 つのひだが発生し，そのひだが拡大・回転することで分断し 3 本のオーロラとなる．その後，分断したオーロラが再び結合することで，最終的に二層のオーロラを形成するアニメーションである．指定したオーロラ形状を図 3.7，アニメーションの生成結果を図 3.12 に示す．1 フレーム当たりの生成時間は約 3.0～5.0 秒であった．



図 3.7: シーン 4 で指定したオーロラ形状

シーン 5

大規模なシミュレーションの検証としてシーン 5 を設定した。指定したオーロラ形状を図 3.8, アニメーションの生成結果を図 3.13 に示す。1 フレーム当たりの生成時間は約 7.8~10.1 秒であった。

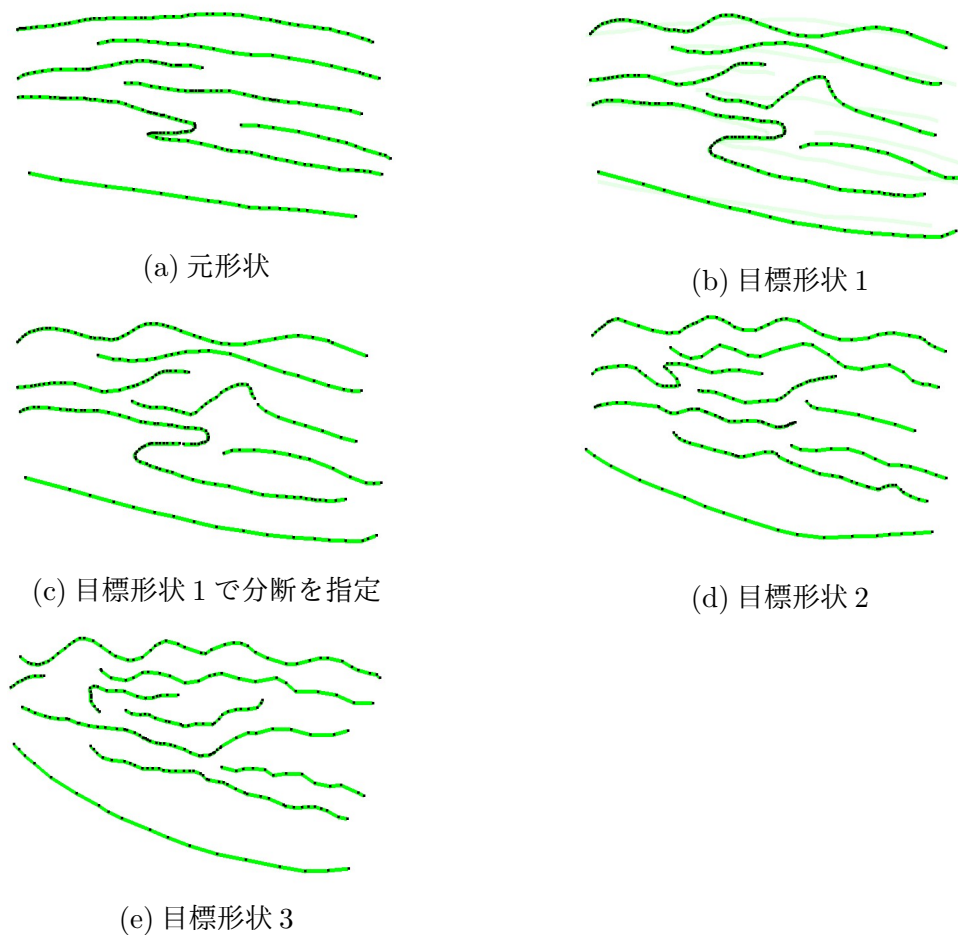


図 3.8: シーン 5 で指定したオーロラ形状

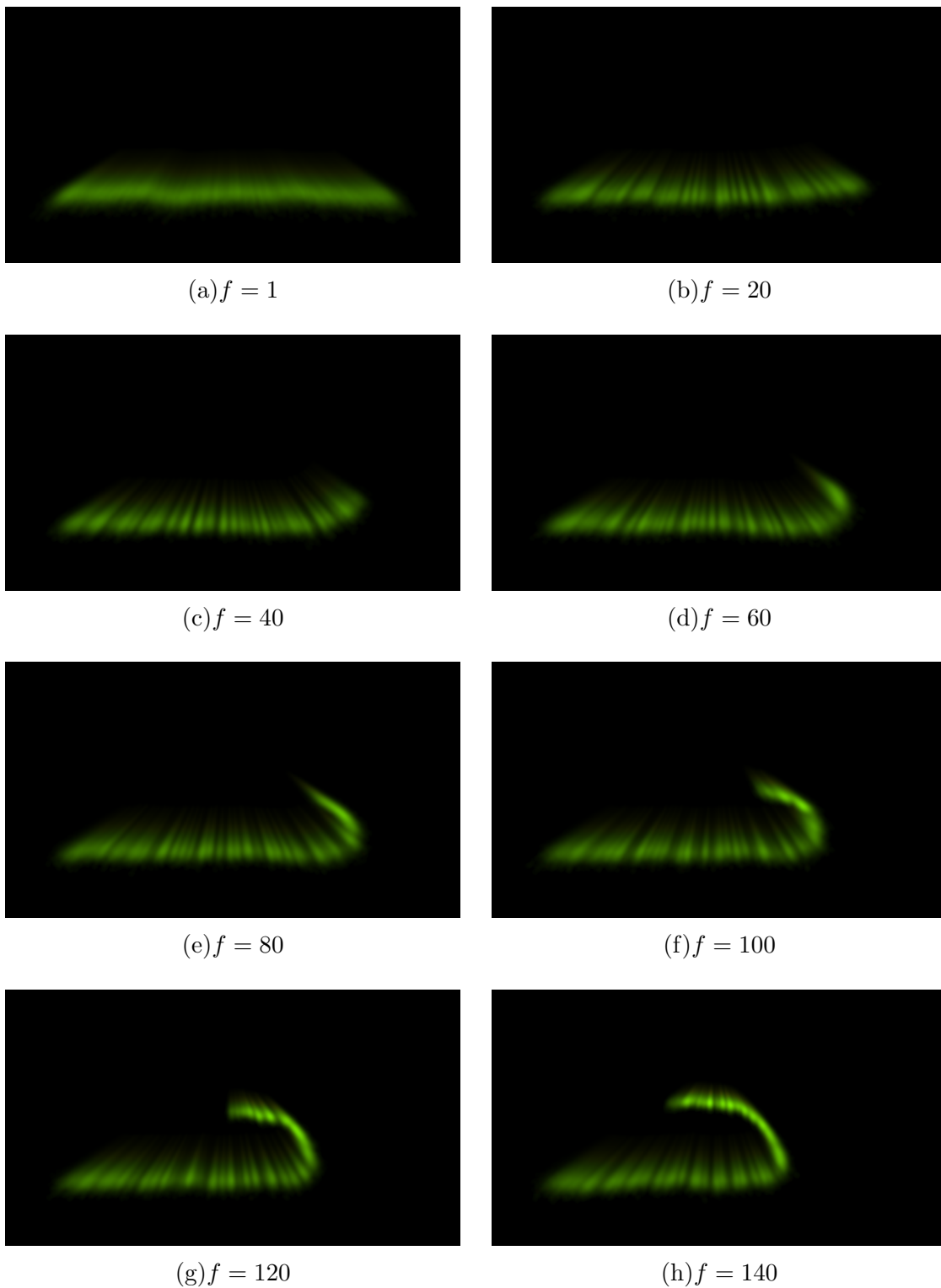


図 3.9: シーン 1 のレンダリング結果

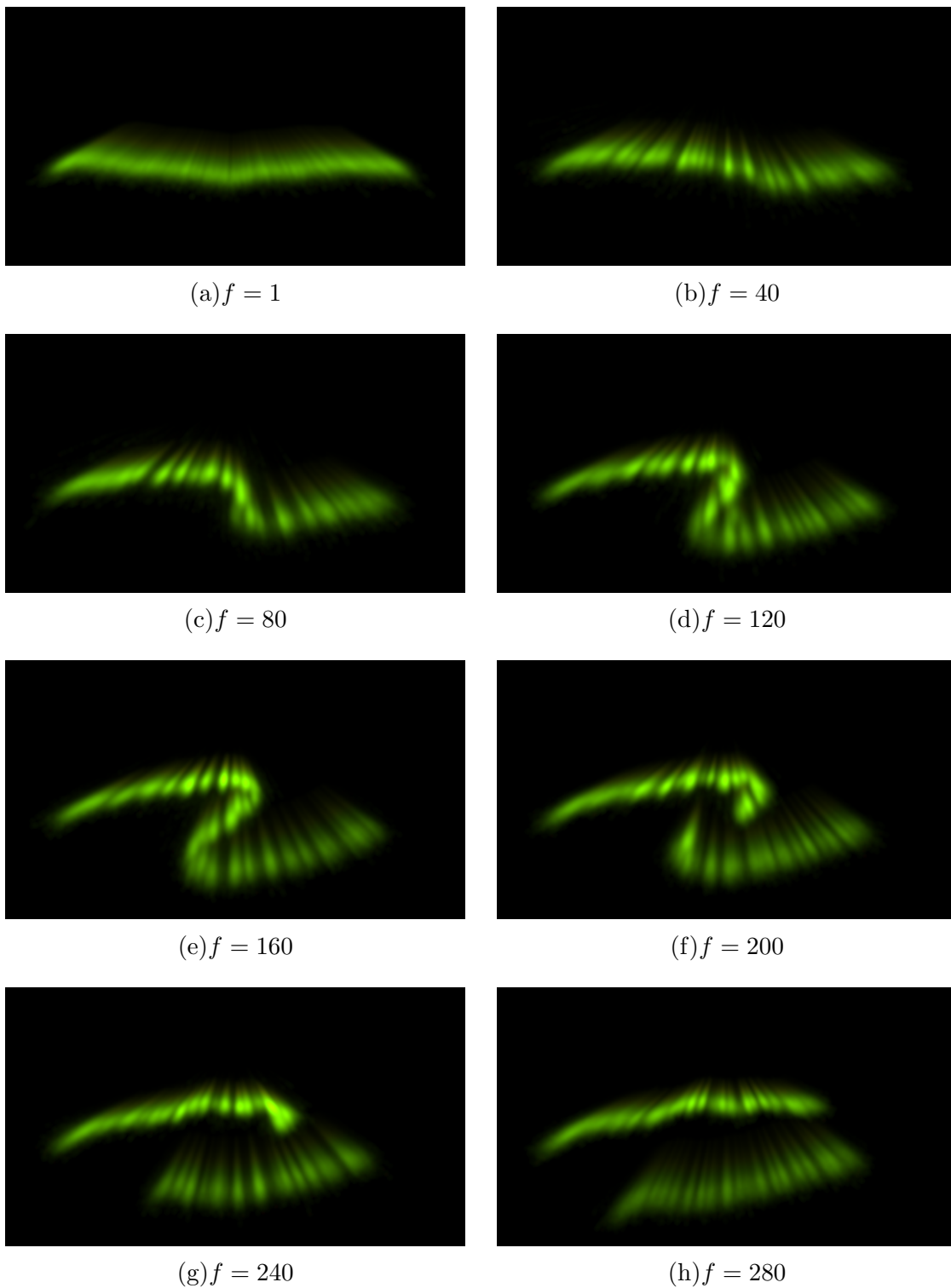
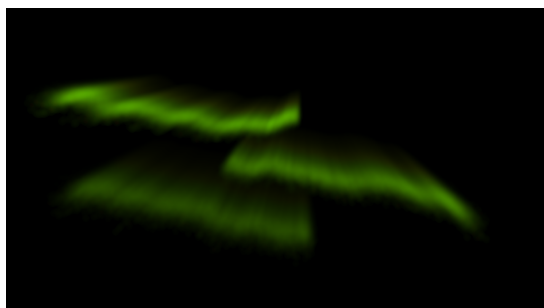
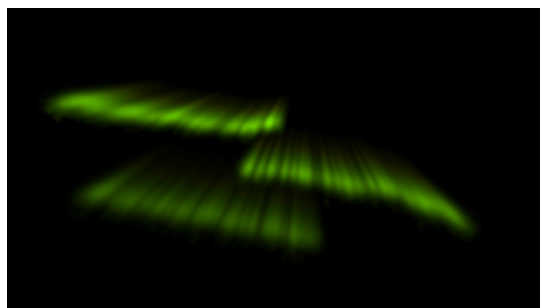


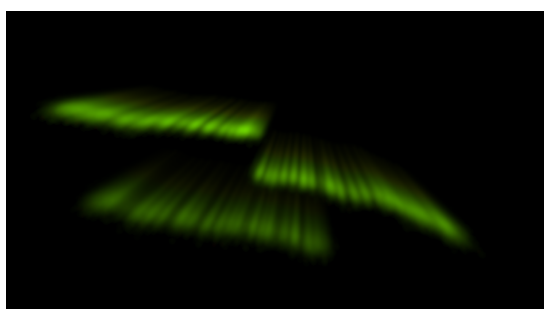
図 3.10: シーン 2 のレンダリング結果



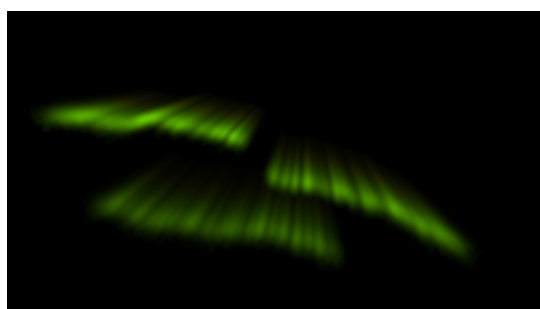
(a) $f = 1$



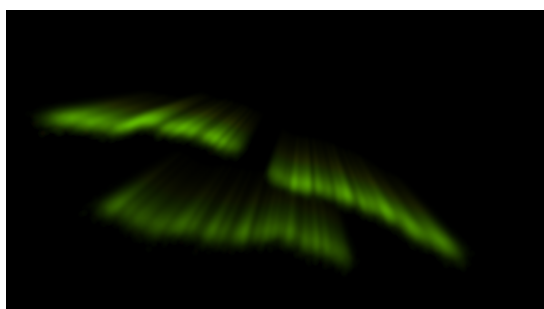
(b) $f = 40$



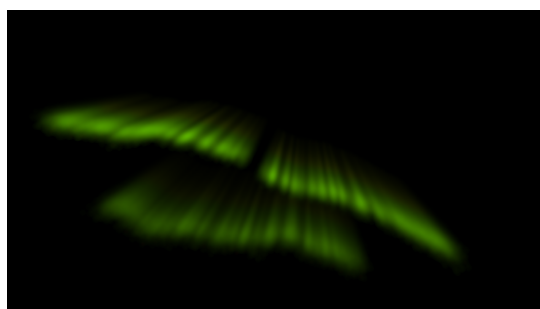
(c) $f = 80$



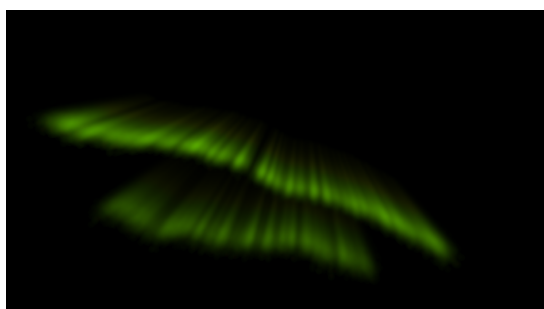
(d) $f = 120$



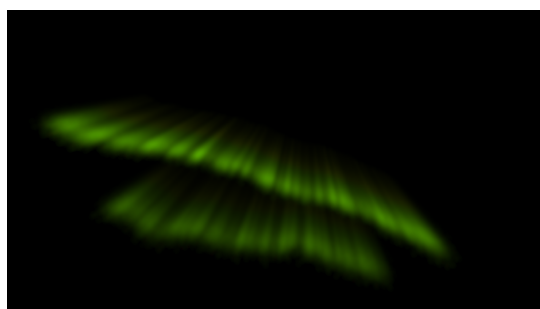
(e) $f = 160$



(f) $f = 200$



(g) $f = 240$



(h) $f = 280$

図 3.11: シーン 3 のレンダリング結果

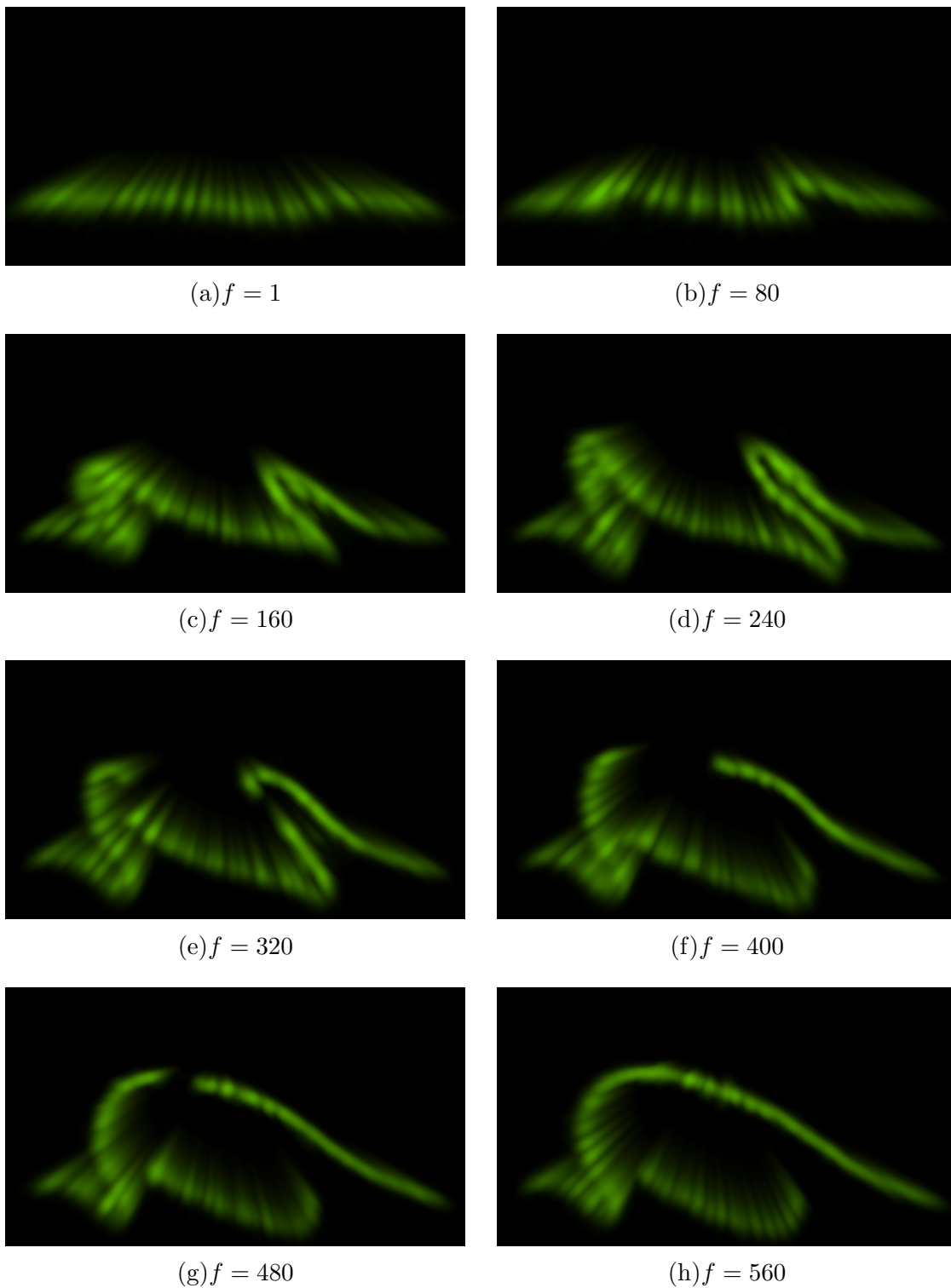
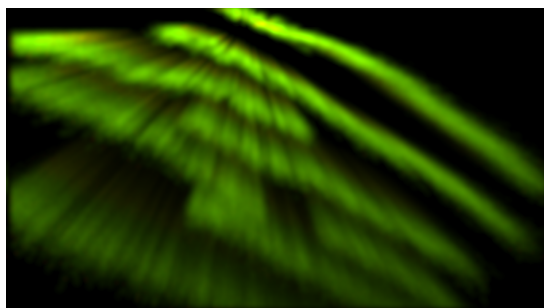
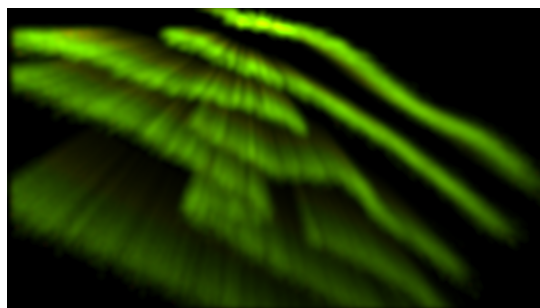


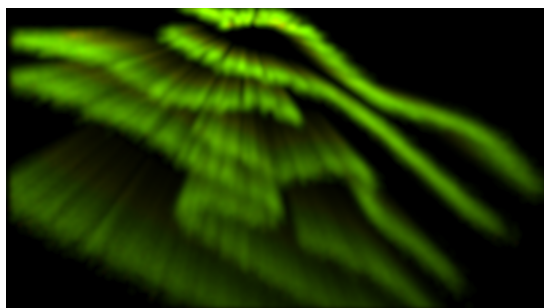
図 3.12: シーン 4 のレンダリング結果



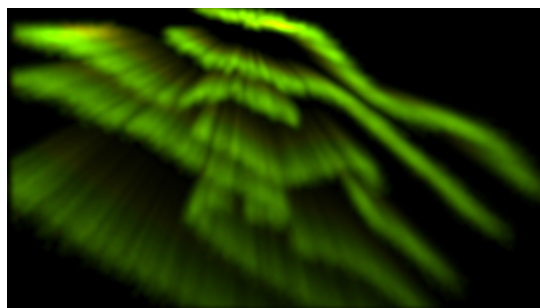
(a) $f = 1$



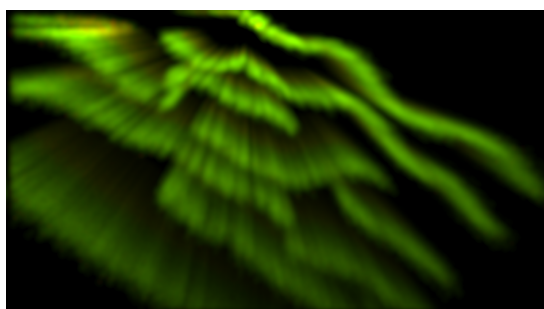
(b) $f = 60$



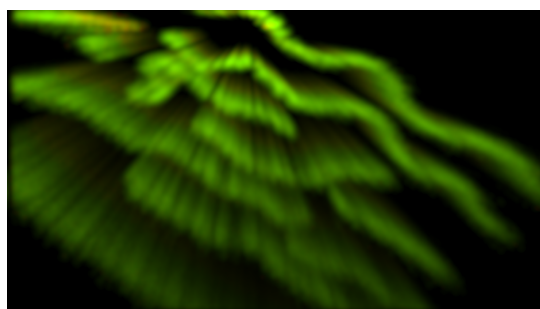
(c) $f = 120$



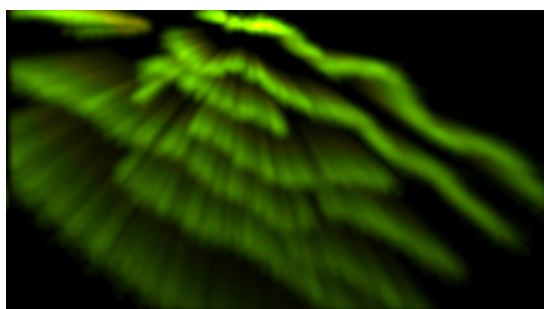
(d) $f = 180$



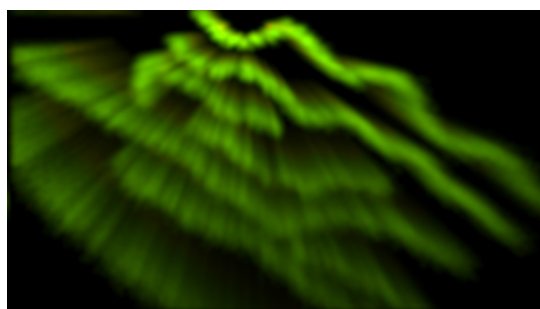
(e) $f = 240$



(f) $f = 300$



(g) $f = 360$



(h) $f = 420$

図 3.13: シーン 5 のレンダリング結果

3.3 考察

シーン 1(図 3.9)

シーン 1 では、1 本のシート状構造を持つオーロラの右端が折り返す、ピーリングと呼ばれる現象を表現した。図 3.9 から、目標形状として指定した図 3.4 に従ってオーロラの形状遷移が行えていることが確認できた。更に、元形状では元々荷電粒子が分布していた部分に、折り返しが発生している状態(図 3.9(e)-(h))では荷電粒子が分布していないことから、電場分布による荷電粒子の移流と、再配置処理による荷電粒子の追加・削除が的確に行えていることが観測できた。ここで、折り返した後の形状(図 3.9(h))では、ガウス・ザイデル法による平滑化の影響で折り返しの内側の電位(図 3.14 黒丸)が近傍判定の閾値を超えてしまっている。しかし、近傍判定に電位による判定だけではなく荷電粒子数による判定を行っているため、折り返しの内側が近傍でないとして判定され、適切に荷電粒子の削除処理が行えていることがわかる。また、Spline カーネル関数を用いたクーロン力を導入していることで、荷電粒子の過度な集中を防ぎつつ、実際のオーロラで観測されるような縞状のレイ構造を表現することができている。

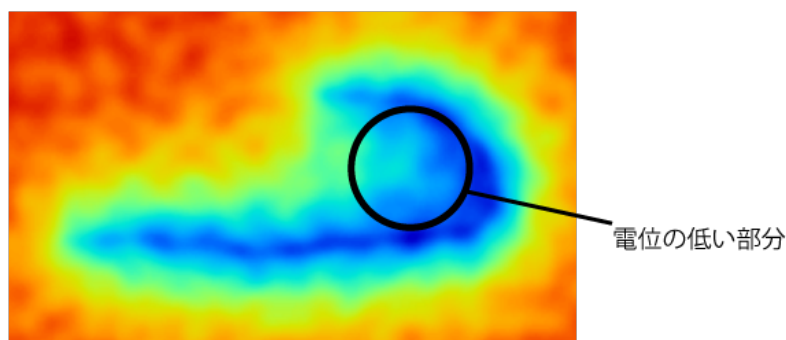


図 3.14: 折り返し後の形状(図 3.9(h))における電位分布

1 フレーム生成にかかる計算時間は、元形状(図 3.4(a))に近い状態ほど短く、目標形状 2(図 3.4(c))に近づくほど長くなった。これは、制御点の xz 座標の最小値と最大値を用いて生成する AABB の面積が、元形状から目標形状に比べて大きくなったことが原因であると思われる。AABB の面積が大きくなった結果、中間目標形状近傍に荷電粒子のランダムサンプリングを行う際に、近傍と判定されずに破棄される荷電粒子が増え、サンプリングにかかる計算時間が長くなったと考えられる。この現象はシーン 2~5 でも同様に観測された。

シーン 2(図 3.10)

シーン 2 では 1 本のシート状オーロラが S 字状に変形し、中心部分から分断することで 2 本のオーロラを形成するアニメーションを再現した。図 3.10(a)-(d) で S 字状のオーロラを形成し、その後図 3.10(e)-(h) において、中心部分が徐々に分断していき、最終的に 2 本のオーロラを形成するアニメーションを生成できたことが確認できる。図 3.10(d) の S 字状構造では、シーン 1 と同様折り返しの内側の電位は近傍判定の閾値を超えているが、荷電粒子

数による近傍判定が適切に行われているため、荷電粒子が分布せず適切なS字形状を形成できている。また、分断が発生し始めている図 3.10(d)-(f) から、分断した線部分の荷電粒子が不連続に消失しているのではなく、徐々に消失していることが観測できる。これは、分断を設定した際の制御点の接続情報の更新により、中間目標形状の線分が徐々に分断されているためである。

1 フレーム生成にかかる計算時間はシーン 1 に比べて約 0.2~0.4 秒ほど遅くなった。これは、シーン 1 に比べて AABB の面積が大きくなった影響でサンプリングにかかる時間が長くなった点と、S 字構造の折り返し付近における近傍荷電粒子の影響を考慮したクーロン力計算の計算量が大きくなった点が原因であると考えられる。

シーン 3(図 3.11)

シーン 3 では、3 本のシート状オーロラのうち、2 本のオーロラが接近し結合することで長いオーロラと短いオーロラを形成するアニメーションを再現した。図 3.11(e)-(h) において、手前の 2 本のオーロラが徐々に近づいていき、1 本のオーロラを形成する結合現象を表現できていることが確認できた。シーン 2 で検証した分断現象と同様、制御点の接続情報を適切に更新していることで、選択した 2 つの制御点の間に不連続に荷電粒子が現れるのではなく、2 本のオーロラから徐々に線が伸びていくようなアニメーションを生成できている。ただし、3 本のシート状構造を表現するために、シーン 1 とシーン 2 に比べて制御点の個数が多くなっており、1 フレーム生成にかかる計算時間は長くなった。

シーン 4(図 3.12)

シーン 4 では、ひだの発生 (図 3.12(a)-(c))、ひだの回転と拡大 (図 3.12(c)-(d))、ひだの分断による 3 本のオーロラの形成 (図 3.12(d)-(f))、分断した部分が再結合し 2 層のオーロラを形成する動き (図 3.12(f)-(h))、といった複数の動きを組み合わせたアニメーションを生成できていることが確認できた。また、制御点の接続情報の更新によって分断では徐々に線分が分かれていき、その部分が再結合によって徐々に近付いていく動きを観測することができる。ただし、このような複雑な現象を再現しようとする、図 3.7 に示したように多くの目標形状と接続関係を入力する必要がある点は問題と考えられる。目標形状の数が多くなった影響によって、形状指定にかかる時間も長くなり、元形状と全ての目標形状を設定する際に約 4 分の時間を要してしまった。

シーン 5(図 3.13)

シーン 5 では、複数本のシート状オーロラが揺らめき、分断・結合を行う大規模なアニメーションを生成した。複数本のオーロラが接近した場合でも、各オーロラの間荷電粒子を再配置処理によって削除しているため、1 本 1 本が独立したオーロラとして観測された。しかし、シート状構造の本数が増えた影響で、荷電粒子をランダムサンプリングする際に近傍判定を行う線分の数が多くなり、中間目標形状を 1 つ生成するのに 10 秒以上の時間がかかるといった問題が生じた。複数本のシート状構造をレンダリングする際に衝突点における

発光範囲が重なり、積算時のアトミック演算による遅延が大きくなった点も計算時間が長くなった要因として挙げられる。また、複数本のシート状構造を含む形状指定では、単純な形状変化に比べて形状指定に要する時間が多くなってしまい、操作に慣れた人でも元形状と全ての目標形状を設定するのに7分ほどかかってしまった。

第4章

結論

4.1 まとめ

本論文では、アーティストの指定した形状から、その形状に近付けるような力を働かせる電場分布を生成することで、オーロラの高周波な揺らめくような動きを保持したまま、分断や結合を含めた形状制御を行う手法を提案した。アーティストの指定した目標形状の近傍に荷電粒子を配置し、配置した荷電粒子の分布から反復法を用いて電位・電場分布を生成する。それを基に疑似的なローレンツ力とクーロン力を考慮した運動計算を行う事で、オーロラの形状制御を実現した。また、オーロラ形状を制御点とそれを結ぶ線分で近似し、制御点間の接続情報を段階的に変更することで、オーロラ特有の現象である分断・結合現象の表現を可能とした。加えて、荷電粒子にかかるクーロン力の計算、荷電粒子の再配置処理、レンダリング処理において、それぞれの処理に適したGPUでの並列化手法を提案し、CUDAによる実装を行った。実験では、実際のオーロラで観測されるピーリング現象や分断・結合現象を含めた形状制御を行い、提案手法の有用性を確認した。

4.2 今後の課題

今後の課題として、分断や結合が発生する場合の目標形状生成の自動化が挙げられる。提案手法では、シーン2やシーン3の目標形状設定(図3.5, 3.6)で示したように、分断や結合が発生する場合は、分断・結合する直前の形状を目標形状として設定し、その目標形状にたいして分断・結合が発生する線分を選択する必要がある。そのため、シーン4やシーン5のように複雑なオーロラアニメーションを生成する場合には、多くの目標形状の設定が必要となり、アニメーション生成までに時間がかかるといった問題点がある。この問題を解決するため、分断や結合が発生する際のオーロラの動きを観測的な特徴からパターン化し、元形状と目標形状を設定するだけで分断・結合直線の形状を自動生成できるようにする必要がある。

また、現在は制御点の対応関係がアーティストのマウスドラッグ操作に依存してしまっている。このため、より実際のオーロラ現象に近いアニメーションを生成するには、オーロラ構造の特徴から、元形状と目標形状間での制御点の対応関係の自動設定を実現する必要がある。

今後はこれらの問題を解決していくとともに、使用感に関するアンケート調査などを基にしたインターフェースの改良や、シミュレーションの高速化を行っていき、より使いやすい制御システムの実現を目指す。

謝辞

本論文を締めくくるにあたり，多岐にわたって研究をご指導して頂いた藤澤誠助教に心から感謝の意を表します。また，合同ゼミにおいて多くのご助言・ご意見を頂いた三河正彦准教授に深く感謝いたします。そして，日々の研究活動において多くのご協力を頂きました物理ベースコンピュータグラフィックス研究室ならびにソーシャルロボット研究室の皆様，諸先輩方に深く御礼申し上げます。

参考文献

- [1] 渋川雄平, 土橋宜典 山本強, “ガス状物体のボリュームレンダリングのための特徴量に基づく伝達関数設計手法”, 映像情報メディア学会誌, Vol. 68, No. 2, pp. J66-J71, 2014.
- [2] G. Kawada and T. Kanai, “Procedural Fluid Modeling of Explosion Phenomena Based on Physical Properties”, In *Proc. Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, pp. 167-176, 2011.
- [3] G. Baranoski, J. Rokne, P. Shirley, T. Trondsen and R. Bastos, “Simulating the Aurora”, *J. Visual. Comput. Animat.*, Vol. 14, No. 1, pp. 43-59, 2003.
- [4] 片岡龍峰, “岩波化学ライブラリー 243 オーロラ!”, 岩波書店, 2015.
- [5] ナヌック・オーロラツアーズ, “車でオーロラを追いかける! 世界で一番オーロラに近い場所・イエローナイフ”, <http://www.aurora-guide.com/auroradiary.html>, (参照 2017-12-06) .
- [6] 国立極地研究所, “南極サイエンス基地 いろいろなオーロラ”, <http://polaris.nipr.ac.jp/academy/science/aurora/index.html>, (参照 2017-12-06) .
- [7] 小島啓史, 竹内亮太, 渡辺大地, 三上浩司, “特徴的な動き方を考慮したオーロラのビジュアルシミュレーション”, 芸術科学会論文誌, Vol. 12, No. 1, pp. 24-35, 2012.
- [8] 米山考史, 近藤邦雄, “発光原理を考慮したオーロラのビジュアルシミュレーション”, 日本図学会 2005 年度大会学術講演論文集, pp. 69-74, 2005.
- [9] 井上太郎, 牧野光則, “CG によるオーロラのモデリング”, 第 11 回 NICOGRAPH 論文コンテスト論文集, pp. 161-170, 1995.
- [10] G. Baranoski and J. Wan, “Simulating the Dynamics of Auroral Phenomena”, *ACM Transactions on Graphics*, Vol. 24, pp. 37-59, 2005.
- [11] 津郷晶也, 玉木徹, 金田和文, “荷電粒子が電磁場から受ける力を考慮したオーロラのアニメーション”, 情報処理学会研究報告 (グラフィクスとCAD研究会報告), No. 12, pp. 31-36, 2009.
- [12] 小島啓史, 竹内亮太, 石川知一, 三上浩司, 渡辺大地, 柿本正憲, 近藤邦雄, “分断と再結合現象を考慮したオーロラのビジュアルシミュレーション”, 情報処理学会論文誌, Vol. 5, No. 8, pp. 1886-1898, 2014.
- [13] L. Shi and Y. Yu, “Taming Liquids for Rapidly Changing Targets”, In *Proc. Eurographics/ACM SIGGRAPH Symposium on Computer Animation*, pp.229-236, 2005.
- [14] R. Fattal and D. Lischinski, “Target-Driven Smoke Animation”, *ACM Transactions on Graphics - Proc. ACM SIGGRAPH 2004*, Vol. 23, No. 3, pp.441-448, 2004.

- [15] 水谷圭佑, 佐藤周平, 土橋宜典, 山本剛, “流体シミュレーションによる炎の形状制御”, 情報処理学会研究報告, Vol.2014-CG-155, No.8, 2014.
- [16] Y. Dobashi, K. Kusumoto, T. Nishita and T. Yamamoto, “Feedback Control of Cumuliform Cloud Formation based on Computational Fluid Dynamics”, *ACM Transactions on Graphics*, Vol. 27, No. 3, pp. 94:1-8, 2008.
- [17] 荒川雄太郎, 伊藤弘樹, 菊池司, “形状を制御可能とした花火のビジュアルシミュレーション”, 映像情報メディア学会技術報告, Vol. 38, No. 16, pp. 99-100, 2014.
- [18] S. Martin, B. Thomaszewski, E. Grinspum and M. Gross, “Example-Based Elastic Materials”, *ACM Transactions on Graphics*, Vol. 30, No. 4, pp. 72:1-8, 2011.
- [19] 小口高, “オーロラの物理学入門”, 名古屋大学太陽地球環境研究所, 2010.
- [20] 遠藤雅守, “電磁気学-初めて学ぶ電磁場理論-”, 森北出版, 2013.
- [21] K. Perlin, “An Image Synthesizer”, *ACM SIGGRAPH Computer Graphics*, Vol. 19, No. 3, pp.287-296, 1985.
- [22] M. Müller, D. Charypar and M. Gross, “Particle-based Fluid Simulation for Interactive Applications”, In *Proc. Symposium on Computer Animation 2003*, pp.154-159, 2003.
- [23] M. Desbrun and M. P. Cani, “Smoothed Particles: A New Paradigm for Animating Highly Deformable Bodies”, In *Proc. Eurographics Workshop on Computer Animation and Simulation*, pp.61-76,1996.
- [24] M. Becker and M. Teschner, “Weakly Compressible SPH for Free Surface Flows”, In *Proc. 2007 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, pp. 209-217, 2007.
- [25] 関口智大, 藤澤誠, 三河正彦, “スクリーン空間を用いたオーロラの高速なシミュレーション”, *Visual Computing /グラフィクスと CAD 合同シンポジウム 2016 予稿集*, pp. 23:1-4, 2016.
- [26] 丸山隆, “6. 電離圏プラズマ (< 小特集 > 宇宙天気予報)”, *プラズマ・核融合学会誌*, Vol.82, No.11, pp.762-766, 2006.
- [27] 国立極地研究所 (編), “南極の化学 2 オーロラと超高層大気”, 古今書院, 1983.
- [28] 森礼於, 富永守, “講座テレビジョン画像の評価技術 (第 2 回):テレビ画像の光学的測定”, *テレビジョン学会誌*, Vol.37, No.9, pp. 736-744, 1983.
- [29] Colour & Vision Research Laboratory, “Colour Matching Functions”, <http://www.cvrl.org/>, (参照 2017-12-13) .
- [30] 内村創, “ゲームのための色彩工学”, *Computer Entertainment Developers Conference 2016*, 2016.
- [31] NVIDIA Corporation, “CUDA Toolkit”, <https://developer.nvidia.com/cuda-toolkit>, (参照 2017-12-22) .
- [32] S. Green, “Cuda Particles”, *NVIDIA Whitepaper*, 2008.

- [33] N. Satish, M. Harris and M. Garland, “Designing Efficient Sorting Algorithms for Manycore GPUs”, In *Proc. 2009 IEEE International Symposium on Parallel & Distributed Processing*, pp. 1-10, 2009.