

2003年度 卒業論文

リアルタイム3DCGにおける
風を考慮した降雪表現に関する研究

指導教員：渡辺 大地 講師

メディア学部 3DCG アプリケーション構築プロジェクト

学籍番号 99P027

安藤 大志

2003年度 卒業論文概要

論文題目

リアルタイム3DCGにおける
風を考慮した降雪表現に関する研究メディア学部
学籍番号：99P027

氏名

安藤 大志

指導
教員

渡辺 大地 講師

キーワード

バーチャルリアリティ, リアルタイムシミュレーション, 降雪, 風

近年、ハードウェアの処理能力の向上等によりさまざまな自然現象や物理現象をリアルタイムに処理することが可能になってきた。本研究ではその中でも季節感を感じさせる要素である降雪現象の表現向上手法を提案する。実際の降雪は風によって複雑に動きを変えるが、従来のリアルタイム3DCGにおける降雪表現では状況変化によって起こる突発的な風などを考慮しているものは少ない。風をシミュレートする方法としては数値流体力学的手法が考えられるが、一般的に膨大な計算時間が必要となる。そこで本手法では、数値流体力学の分野で使用されている手法の中でも状況変化に柔軟に対応しやすいと考えられる粒子法を参考に簡易的な空気のモデルを用意し、風の影響を考慮した降雪表現をリアルタイムに行うことを目的とした。この手法では、局所的な風を考慮することが可能である。また、建物や移動物においてはその部分の風を制御することで対応し、衝突判定も同時に行っている。本論文ではいくつかのシチュエーションを用意し、その効果を検証した。

目次

第1章	序論	1
1.1	はじめに	1
1.2	論文構成	4
第2章	風を考慮した降雪表現	5
2.1	風を考慮する	5
2.1.1	数値流体力学	5
2.1.2	オイレリアンとラグランジアン of 計算法	6
2.1.3	粒子法の考え方	6
2.2	空気の表現	8
2.3	雪の表現	10
2.4	建物等の表現	11
2.5	計算手順	11
第3章	動作検証	14
3.1	実装	14
3.2	実行結果	15
3.2.1	無風状態	16
3.2.2	全体的な風	16
3.2.3	局所的な風	17
3.2.4	建物の影響	18
3.2.5	移動物の影響	19
3.2.6	特殊な状況	20
3.3	空気モデルの検証	21
第4章	まとめ	23
	謝辞	24
	参考文献	25

第 1 章

序論

1.1 はじめに

3DCG (3次元コンピュータグラフィックス) によってリアルな世界を表現しようとする試みは数多く行われており、現在でも関心の高い分野の1つである。リアルな世界を表現する上で重要な要素は2つある。1つは「描写」であり、3DCGにおいてはワイヤフレーム表示やサーフェス表示に始まり様々なマッピング・シェーディング・レンダリング手法が考案されてきた。もう1つは物体の「挙動」であり、一般的なものとしてリジッドボディダイナミクス、ソフトボディダイナミクス、クロスシミュレーションなどがある。一昔前までこのようなCGを制作するためには高価なグラフィックス制作機器を使い長い時間をかけて計算する必要があったが、現在はパソコンの高速化・低価格化によって個人でも手が届くものとなった。またゲーム機に代表されるように、ハイクオリティなCGをリアルタイムで表現することも可能となってきている。

ゲームやバーチャルリアリティ等のリアルタイム3DCGにおいては物体のリアルな挙動をインタラクティブに表現することが臨場感の面で大きな役割を果たす。そのため、さまざまな物理現象・自然現象による表現が積極的に取り入れられている [1]。しかし、ハードウェアの処理能力の限界等により、実現されていない表現も数多い。本研究ではその中でも風の影響を受ける降雪表現に注目した。風は目に見えないため、一般的には風の影響を受けているものを通して間接的に表現す

る。映画等の特殊効果に見られるような降雪 CG は非常にリアルで実写と見分けがつかないほどだが、これらの多くはあらかじめ時間をかけて動きをシミュレートしレンダリングを行っている。この手の手法による降雪シーンの CG 生成に関する研究は村岡ら [2] によっていくつか報告されており、気流の乱れなども考慮に入れられている。一方リアルタイム CG による降雪表現においては図 1.1 のように雪に適当な乱れを与えるのみで一定方向に向けて降らせる手法が主流である [3]。しかし、降雪は草木や旗と違い空間全体を満たしており、なおかつ風の影響を如実に反映する性質がある。先の手法でもある程度リアリティを持たせることは可能だが、静止物や移動物によって生じる局所的な風の乱れなどは考慮していないため、動きがやや単調で全体的なものになってしまう。また雪が他の物体に衝突することも考慮していないため、建物などを突き抜けてしまうような現象が起こる。もしこれらの状況に応じた雪の挙動の変化をインタラクティブに表現できれば、臨場感や演出の面での向上が期待できる。

そこで本研究では降雪と降雪空間内の状況との関係に焦点を絞り、風によって挙動が変わる降雪、また静止物や移動物などの影響を受けて乱れたり消滅したりする降雪をリアルタイムで表現する手法を提案する。アプローチとしては、まず風を考慮するために空気の流れを考える。すなわち、空気の流れを計算し雪の挙動を計算するという 2 段階的な手順を踏む。この方法は直感的にも分かりやすく

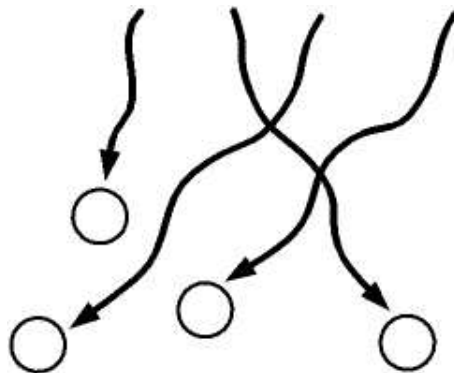


図 1.1: 適当な乱れによる降雪表現

現実にも即しているため、より自然な表現が可能であると考えられる。空気の流れは数値流体力学 [4] の手法を用いることで計算が可能である。最も一般的な手法は格子法 [5] と呼ばれるもので、空間を格子分割して計算を行う。この手法では処理の準備段階として流体が存在する空間を格子状に分割しておく必要があり、空間が複雑だと非常に煩雑な作業となる。また状況変化によって空間が変形した場合それに応じて格子を変化させる必要も生じる。こういった性質はさまざまな状況変化が想定されるリアルタイム 3DCG においてデメリットになる。そこで、本研究では粒子法 [6] という方法に注目した。これは粒子という概念を用い粒子の運動を計算することで流れをシミュレートするものである。前者に比べアルゴリズムが比較的単純で空間の格子分割も必要ないといった特徴を持つため、先に述べたデメリットは解消できる。しかし、理論としては発展途上にあり計算方法は確立されていない。また主な用途としては実験が困難な現象を分析することを想定しているため、計算コストより高精度を優先している。このような計算をそのままリアルタイムで、しかも 3 次元で行うことは現在のパーソナルコンピュータの性能では不可能である。そこで本研究では粒子法を参考としながら次の 3 点を満たす空気のモデルを想定した。

- 状況変化に柔軟に対応可能できること
- シーンへの適用が容易であること
- 現在一般的な性能のパソコンで実行可能であること

雪の挙動に関しては、個々の形状の違いや雪同士の衝突は考慮せず、単純に空気抵抗と重力を受ける粒子と仮定する。よって完全な無風状態の場合はまっすぐに落ちる。これはただ計算を簡略化するだけでなく、既存の手法との違いを検証しやすくする目的もある。

1.2 論文構成

本論文では第 2 章で具体的な表現手法を提案、降雪モデルを試作する。第 3 章では様々な状況における実行結果とその検証、第 4 章で今後の展望について述べる。

第 2 章

風を考慮した降雪表現

2.1 風を考慮する

一般的にリアルタイム CG で降雪を表現する場合、雪に適当な揺らぎを与える。これは、実際の雪が風の抵抗や自身の運動により複雑に挙動する様子をモデル化したものであり、計算は雪の数だけで済むため非常に手軽である。また、場所に応じた力を雪に与えることで、一定方向から吹く風や渦状に吹く風なども簡単に表現できる。しかし、ビルが建ち並び車が走っているなどの状況で吹く風を考慮しようとした場合、場所場所における時間的に変化する風を手動で与えることは難しい。そこで本研究では空気存在をモデル化し、場所に応じた風を自動的に計算する方法を試みた。

2.1.1 数値流体力学

空気などの流体は数値流体力学の分野で使用されている手法により計算できる。流体力学 [7] は古くからある物理学の 1 分野であるが、数値流体力学は、数値解析によって流れを研究、およびその数値解析自身を研究する学問のことであり、計算機の急速な発達とともに進歩してきた。現在では様々な状況下における流れをかなりの精度で解けるまでになっている [8]。

2.1.2 オイレリアンとラグランジアン of 計算法

数値流体力学において、流体を計算する手法は大きく分けて2つある。1つは計算点の位置が固定されているオイレリアンの計算法、もう1つは計算点が移動するラグランジアン of 計算法である。オイレリアン of 計算法に代表されるのは格子法であり、これは空間を格子状に分割し、その格子ごとに計算を行うことで流れを表現する。一方ラグランジアン of 計算法に代表されるのは粒子法であり、こちらは流体を粒子の集合として考え、その粒子の運動を計算することで流れを表現する。これらの考え方は数値流体力学の歴史上初期の頃から存在する [9][10]。

本研究ではリアルタイム 3DCG の世界を扱う。この中では常に様々な物体が動いていることが想定される。これらの影響を考慮した流れを計算する場合、ラグランジアン of 計算法の方がそれら物体と統一的に扱え、便利である。そこで、ここではこのラグランジアン of 計算法である粒子法に注目する。

2.1.3 粒子法 of 考え方

粒子法では流体を図 2.1 のように粒子の集合で表す。

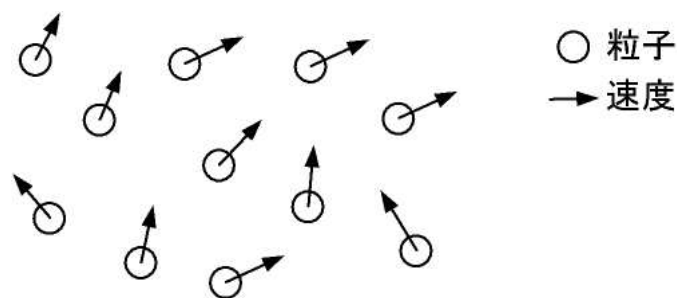


図 2.1: 粒子法

この粒子は計算のために導入された仮の存在で、たとえばある流体を 100 個の粒子で表したとしたら、その粒子 1 つ 1 つはそれぞれ 100 分の 1 の「流体の塊」であることを意味する。粒子法による流れのシミュレーションではこれら粒子の運動を計算することで流体の挙動を求める。

粒子法の1つであるMPS法 (Moving Particle Semi-implicit Method)[11][12]では、図2.2のような流れと相互作用しながら流体中を移動する剛体といった状況のシミュレーションを比較的簡単なアルゴリズムで計算することが可能である [13]。

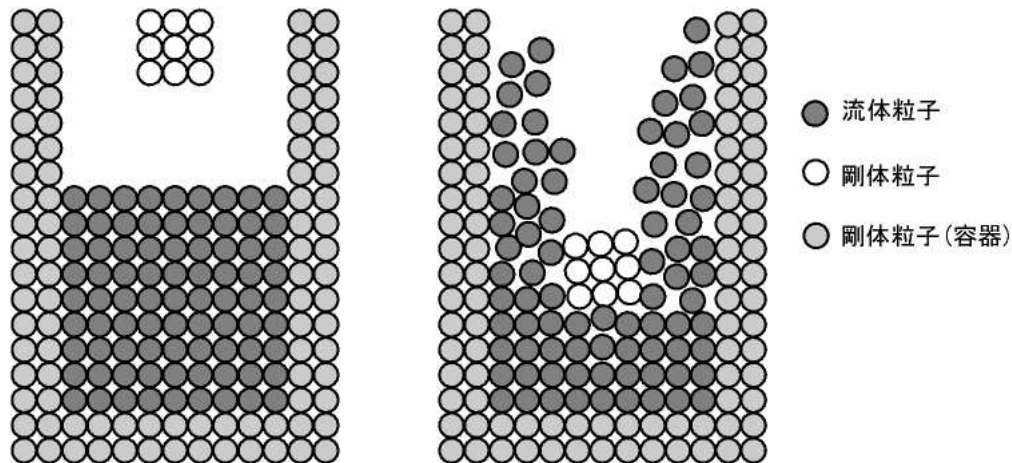


図 2.2: MPS 法における液体への固体の落下 (模式図)

この手法では、流体と剛体をそれぞれ粒子の集合で表し、

手順1 流体粒子と剛体粒子を区別せず、同じ流体粒子として扱い計算を行う

手順2 手順1で崩れてしまった剛体粒子の位置関係を修正する

という2段階の操作でシミュレーションを行う。これは、剛体も流体もほぼ同じものとして扱えることを意味する。つまり、流れを構成する粒子を制御するだけで、さまざまな状況が簡単に設定できる可能性を示している。

本研究ではこの粒子法の考え方を参考に、雪の挙動のベースとなる空気・構造物・移動物など3D空間にあるすべての物体を図2.3のように空気と同様に流体の一種として扱いリアルタイムシミュレーションする方法を提案する。また、雪は粒子として空気粒子との相互作用を考える。次節から具体的な手法について述べる。

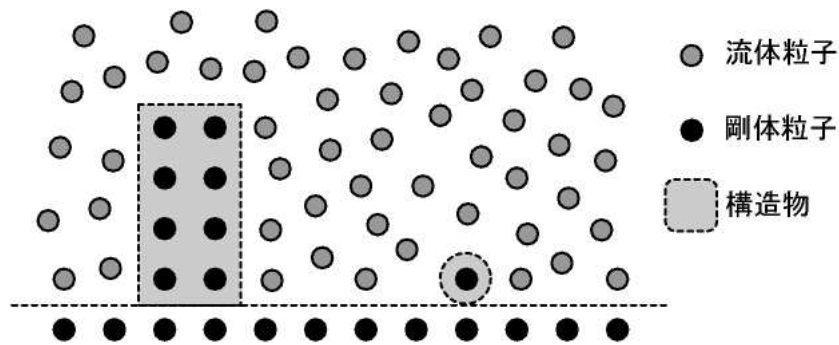


図 2.3: 流体粒子で構成された空間

2.2 空気の表現

空気等の流体は主な性質として粘性と圧縮性を持つ。粘性とは流体に働く摩擦力によって流体の運動をできるだけ均一にしようとする性質のことであり、圧縮性とは流体の密度が変化する性質のことである。ただし、本研究で扱う風程度の流れの場合、空気の圧縮性は無視してもよい。これは流速 u が音速 c に比べて十分遅い、すなわち u/c が 1 よりも十分小さい場合は、非圧縮性の流れとして扱ってよいからである。おおまかには音速の 30%以上の流速をもつ流れに対しては圧縮性の効果を考える必要があるといわれている。また空気の粘性も非常に小さく無視して構わない。よってここでは空気は非粘性・非圧縮性の流体であるものとする。

空気は粒子の集合とする。粘性は無視するため、粒子の運動は非圧縮性によるもののみを考える。一般的に非圧縮性の計算をする場合まず圧力を求め、そこから粒子間の斥力を求める。しかしこの計算は時間がかかるためリアルタイム処理には向かない。よってここでは図 2.4のように空間の広さをあらかじめ制限し粒子間に単純な斥力を働かせることで非圧縮性を表現し、高速化を図る。

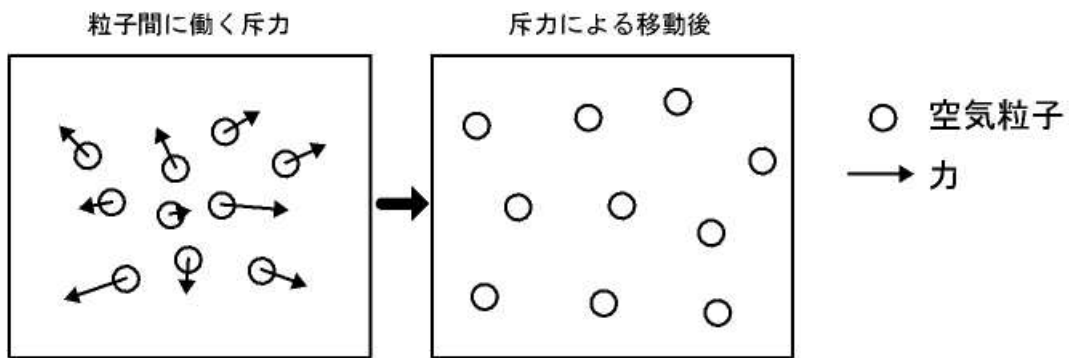


図 2.4: 斥力による非圧縮条件の表現

粒子間相互作用の計算には重み関数 $w(r)$ を使い、距離に応じた斥力を与える。また遠方まで影響が及ばないように工夫することで粒子間相互作用を計算する回数を減らし、高速化を図る。具体的な式としては次のようなものを用いる。

$$w_1(r) = \begin{cases} \frac{r_e}{r} - 1 & r \leq r_e \\ 0 & r > r_e \end{cases} \quad (2.1)$$

ここで r は 2 つの粒子間の距離であり、粒子間相互作用の及ぶ範囲を半径 r_e 以内に制限することを表している。本手法ではこの重み関数を使い次のような粒子間相互作用を考えた。

$$\overrightarrow{F_{j \rightarrow i}} = -\alpha(\overrightarrow{r_j} - \overrightarrow{r_i})w_1(|\overrightarrow{r_j} - \overrightarrow{r_i}|) \quad (\alpha > 0) \quad (2.2)$$

ここで $\overrightarrow{F_{j \rightarrow i}}$ は近傍の各粒子 j から粒子 i に作用する力、 \overrightarrow{r} は粒子の位置ベクトルであり、 α は力の大きさを調整する定数である。(2.2) 式では粒子 j から粒子 i にかかる力のベクトルの向きが必ず粒子 j から粒子 i の方向になるので、粒子間には斥力が働くことになる。また、(2.1) 式の重み関数では r が 0 に近づくとつれて重み関数が無限に大きくなるように工夫している。これにより空気粒子間に働く斥力が図 2.5 のように急激に増大するので、一定距離以下に近づきにくくなる。結果、粒子分布にむらを生じにくくする効果が期待できる。

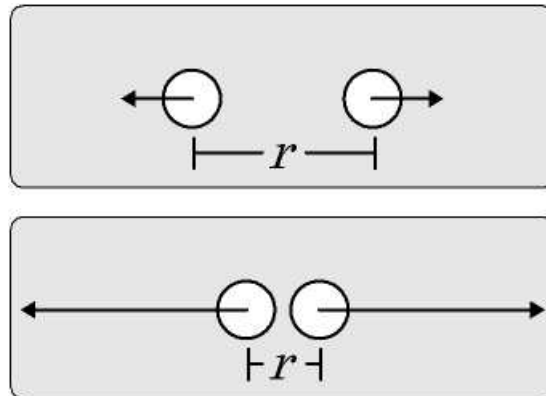


図 2.5: 粒子間距離に応じた斥力

2.3 雪の表現

雪はその挙動の大部分を空気の流れに依存するものとする。ここでは雪粒子は個体差のないものとし、空気抵抗と重力の影響のみを考える。空気抵抗を求めるためには、雪粒子の座標における空気の速度を求める必要があるが、ここでは空気粒子の分布が一様になっているものと仮定し、雪粒子と空気粒子の相互作用として計算するものとした。具体的には、雪粒子と近傍の空気粒子との距離と相対速度から空気抵抗を求める。

まず雪粒子と空気粒子の相互作用について、

$$w_2(r) = \begin{cases} \frac{r_e - r}{r_e} & r \leq r_e \\ 0 & r_e < r \end{cases} \quad (2.3)$$

のような重み関数を使用し粒子間距離に応じた影響力を考えた。雪粒子は空気粒子と交錯する場合もあるので、(2.3) 式は (2.1) 式と異なり r が 0 のとき最大値 1 となるものを用いている。空気抵抗は雪粒子 i と空気粒子 j との相対速度に比例するものとし、

$$\overrightarrow{F}_{j \rightarrow i} = -b(\overrightarrow{u}_i - \overrightarrow{u}_j)w_2(|\overrightarrow{r}_j - \overrightarrow{r}_i|) \quad (2.4)$$

とした。また、その反作用を

$$\overrightarrow{F_{i \rightarrow j}} = -\alpha b(\vec{u}_j - \vec{u}_i)w_2(|\vec{r}_i - \vec{r}_j|) \quad (\alpha > 0) \quad (2.5)$$

として与えた。ここで b は抵抗係数、 \vec{u} は粒子の速度ベクトルである。ここでも α によって調整を行う。また反作用を考えること (図 2.6) は空気に微妙な揺らぎを与え、より雪らしい効果が期待できる。

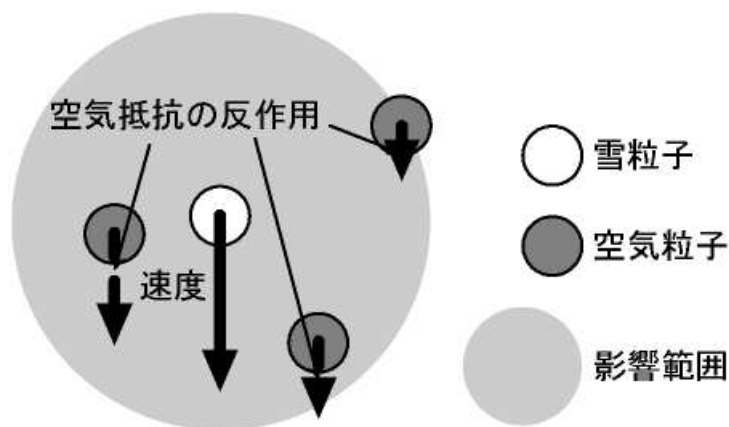


図 2.6: 空気抵抗の反作用

2.4 建物等の表現

建物などの構造物は位置を固定した空気粒子を並べることで表現する。これらの粒子では速度は0として計算する。また移動物の場合は速度・位置を制御できる空気粒子で表現する。雪粒子との当たり判定はこれら空気粒子との距離が一定以下になった場合に行うものとし、雪粒子と空気粒子の相互作用の計算に組み込む。

2.5 計算手順

計算の流れを図 2.7に示す。

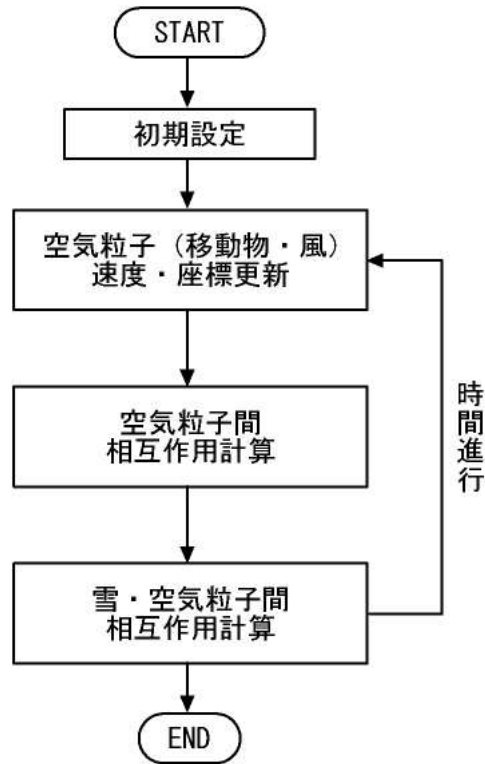


図 2.7: 計算のながれ

まず、時刻 n における粒子の座標, 速度をそれぞれ \vec{r}_i^n, \vec{u}_i^n とおく。第 1 段階では風や物体の移動を任意に行うために空気粒子に外力 f を与え、その速度, 座標を (2.6), (2.7) 式によって \vec{u}_i^*, \vec{r}_i^* に更新する。

$$\vec{u}_i^* = \vec{u}_i^n + f_i^n \Delta t \quad (2.6)$$

$$\vec{r}_i^* = \vec{r}_i^n + \vec{u}_i^* \Delta t \quad (2.7)$$

この計算によって非圧縮性が崩れ、空気粒子の分布にむらが生じると考えられるため、第 2 段階では空気粒子間隔がある程度一定になるように粒子間距離に応じた斥力 ((2.2) 式) を計算し、速度の修正値 \vec{u}_i^{\rightarrow} とする。これを (2.8) 式に示す。

$$\vec{u}_i^{\rightarrow} = \Delta t \sum_{j \neq i} \left[-\alpha (\vec{r}_j^* - \vec{r}_i^*) w_1 (|\vec{r}_j^* - \vec{r}_i^*|) \right] \quad (2.8)$$

この結果をもとに時刻 $n + 1$ における空気粒子の速度と座標を

$$\overrightarrow{u_i^{n+1}} = \overrightarrow{u_i^*} + \overrightarrow{u_i^j} \quad (2.9)$$

$$\overrightarrow{r_i^{n+1}} = \overrightarrow{r_i^*} + \overrightarrow{u_i^j} \Delta t \quad (2.10)$$

と修正する。第3段階では雪粒子 i にかかる重力と空気粒子 j からの空気抵抗の計算を行い時刻 $n + 1$ における雪粒子の速度を

$$\overrightarrow{u_i^{n+1}} = \overrightarrow{u_i^n} + \Delta t \left(\sum_{j \neq i} \left[-b(\overrightarrow{u_i^n} - \overrightarrow{u_j^{n+1}}) w_2 (|\overrightarrow{r_j^{n+1}} - \overrightarrow{r_i^n}|) \right] + g \right) \quad (2.11)$$

とし、最後に雪粒子の座標を

$$\overrightarrow{r_i^{n+1}} = \overrightarrow{r_i^n} + \overrightarrow{u_i^{n+1}} \Delta t \quad (2.12)$$

と更新する。このとき空気粒子 j が雪粒子 i から受ける反作用は

$$f_j^{n+1} = \sum_{i \neq j} \left[-\alpha b (\overrightarrow{u_j^{n+1}} - \overrightarrow{u_i^n}) w_2 (|\overrightarrow{r_i^n} - \overrightarrow{r_j^{n+1}}|) \right] \quad (2.13)$$

として、次のタイムステップに持ち越す。これで1タイムステップの計算が終了する。

第 3 章

動作検証

3.1 実装

本研究では、3次元グラフィックツールキットである「Fine Kernel Tool Kit」[14]を使用しプログラミングを行った。また粒子間相互作用の計算において影響範囲に入っているか否かの判定は、影響範囲の大きさを持った物体の衝突判定と考えることができるので、データのシリアルイズを利用した高速化 [15] を行った。

各要素については以下のような設定をしている。

- 空気粒子

空気粒子は縦 13, 横 13, 高さ 8 で等間隔に配置し、同時にそれを空間の広さとした。最下層, 最上層の空気粒子は固定、空間の四方は計算上それぞれ向かい合う面を結合し無限空間としている。

- 雪粒子

雪粒子は 1000 個用意し空間全体にランダムに降らせ、空間の最下部到達や当たり判定によって消滅した場合、最上部に再び生成するようにした。

- 各種パラメータについて

空気粒子間相互作用における影響範囲 r_e は空気粒子の間隔を 1 とした場合、空気粒子間については 2, 雪・空気粒子間については 1 で計算を行って

る。また雪粒子は抵抗係数 b は大きめに設定し、非常にゆっくり降るものとした。

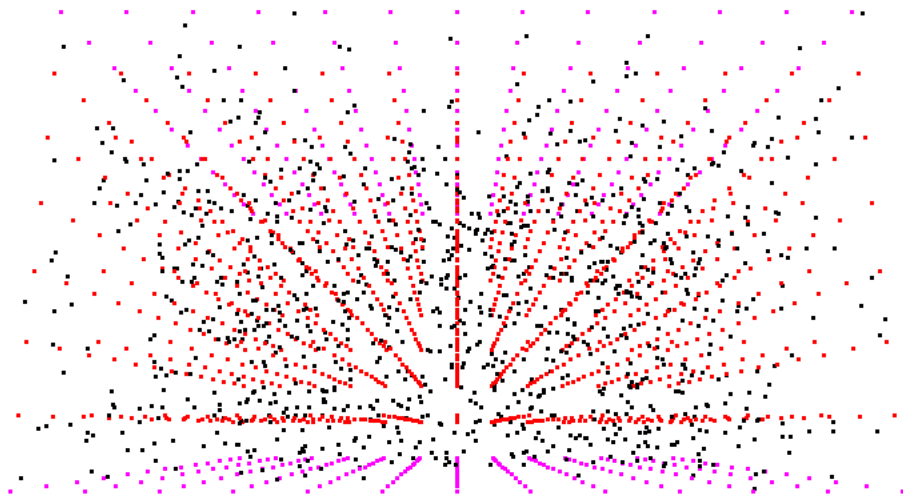


図 3.1: 初期位置 (黒:雪粒子 赤, 紫:空気粒子)

3.2 実行結果

実行は Pentium4 1.5GHz, 512MBRAM, Windows2000 上で行った。動作フレームレートは 24fps 前後であった。以下にいくつかのシチュエーションにおける実行結果とその考察を示す。なお画像には動きの可視化のため雪粒子の速度ベクトルを表示している。

3.2.1 無風状態

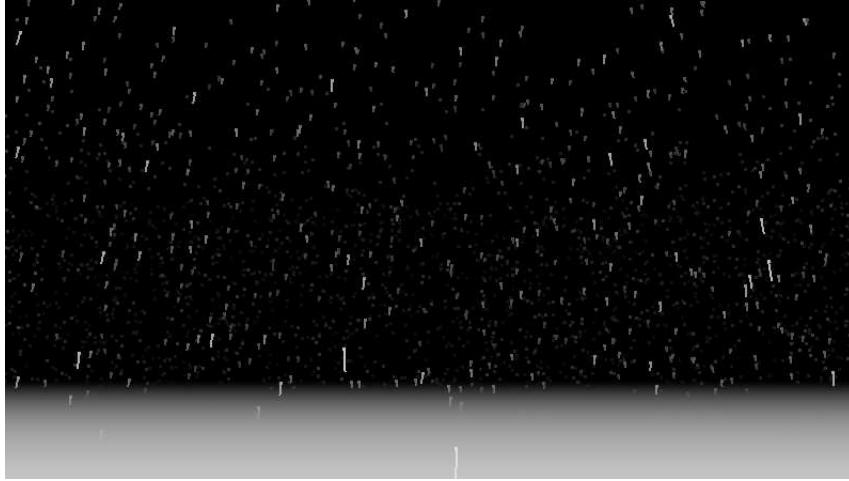


図 3.2: 無風状態における降雪

何もせず放置することでほぼ無風の状態を作った。空気粒子は完全には停止していないが、見た目には雪がほぼ垂直に落ちていく様子が確認できた。動きが単調であり、降雪表現としては味気ない。

3.2.2 全体的な風

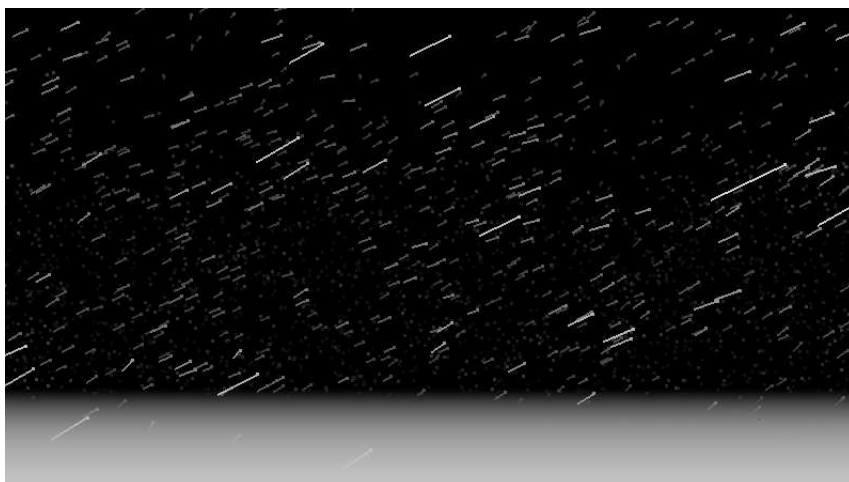


図 3.3: 全体的な風による影響を受ける降雪

全体的な風を起こす方法はいくつか考えられるが、ここでは空気粒子全体のうち20%の粒子に対して外力を与えることで風を起こしている。多少のばらつきがあるが、一定の方向に降る様子が確認できた。外力を与える粒子を一部としたのは、空気粒子の動きにある程度自由度を持たせる目的があったのだが、結局は時間とともに空気粒子全体が一方向に動く結果となった。こうなると無風状態の時と同じく動きが単調で降雪表現として物足りなさが残るが、外力の与え方に変化をつけることで改善できる問題だと考える。

3.2.3 局所的な風

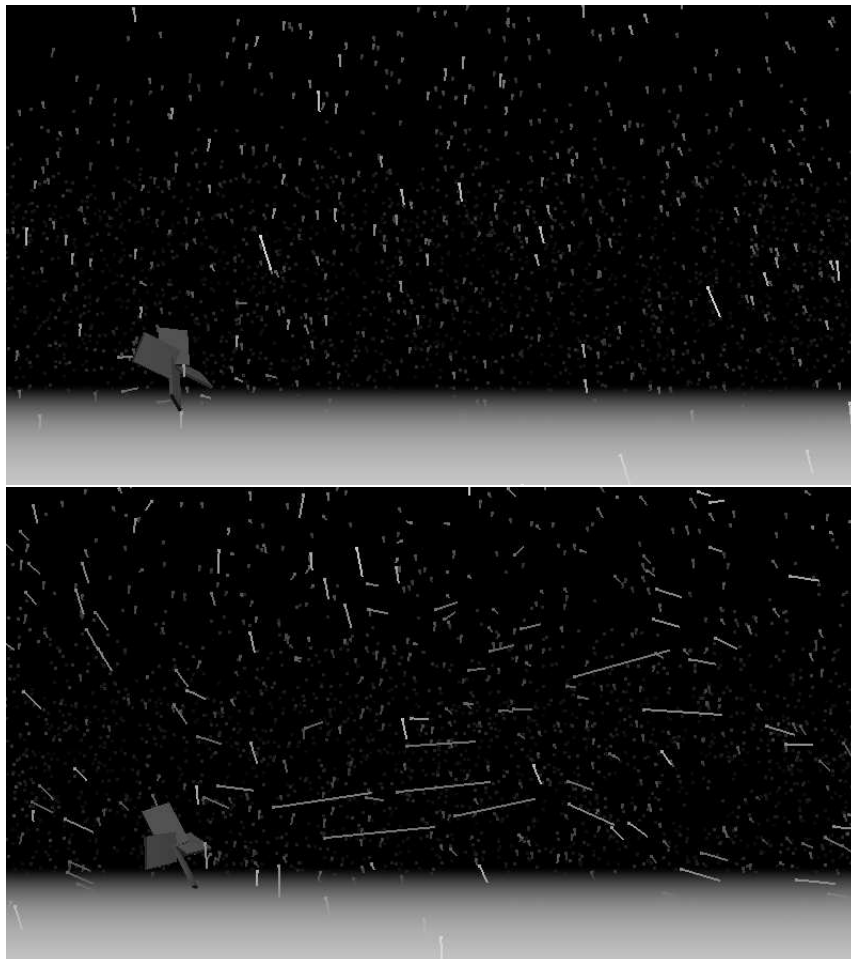


図 3.4: 局所的な風の影響を受ける降雪

局所的な風は、空間中の一定範囲内にある空気粒子に対してのみ外力を与えることで表現する。画面にはわかりやすいように対象となっている範囲の中央にオブジェクトをおいている。時間経過とともに風が次第に発達していく様子が確認できた。

3.2.4 建物の影響

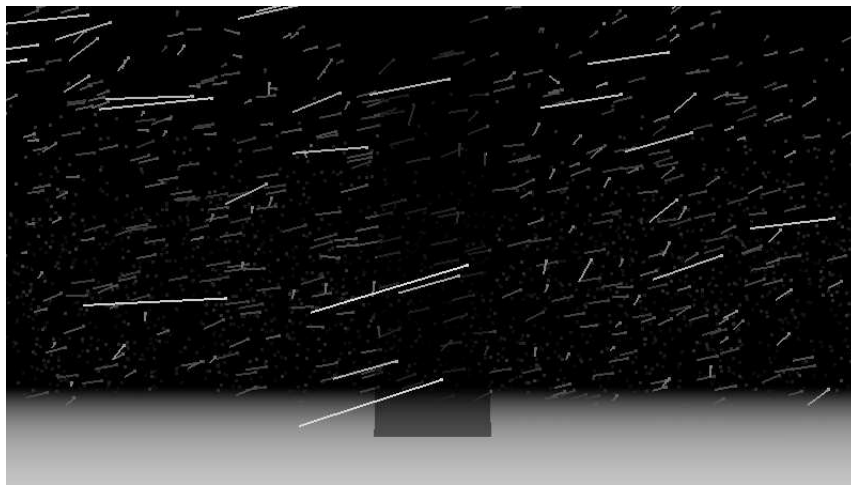


図 3.5: 建物による影響を受ける降雪

ここでは、建物を構成する空気粒子として縦 5, 横 1, 高さ 5 で初期配置時と同間隔で並べたものを空間中央に固定した。画面中では暫定的に同じ大きさの直方体を配置している。このまま放置すると無風状態の時とほとんど変わらないので、画面左側に向けて全体的な風を起こしている。結果、空気粒子は建物を回り込むような挙動をし、建物付近で雪粒子の動きが鈍る様子が確認できた。ただ、計算上この建物は球が重なり合ってるようなものであるため、現粒子配置のままでは表面にムラが生じており計算上あまり好ましくない。粒子分布との兼ね合いも考慮する必要があり、今後の課題である。

3.2.5 移動物の影響

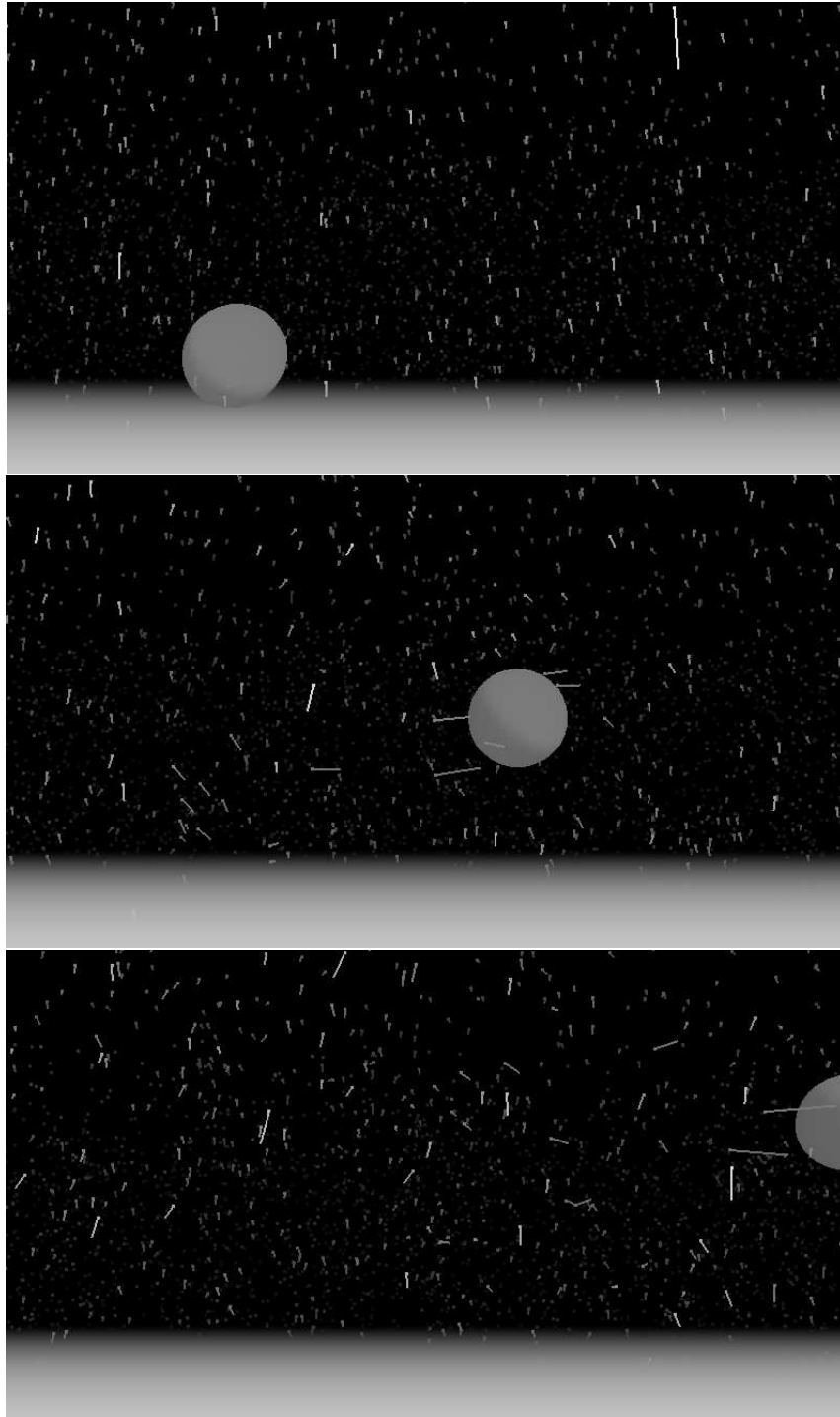


図 3.6: 移動物による影響を受ける降雪

移動物としては、最も単純に空気粒子1つを制御できるものを用意した。画面中では影響範囲を考慮し、球として表示している。空気粒子そのものを動かしているため近辺の雪はもちろん影響を受けるが、移動物を通ったあと空気が乱れて雪の挙動が変化する様子も確認できた。現時点では移動物は空気粒子間の斥力を受けないようにしているが、質量などを考慮することによって風の抵抗を受けて転がるといった表現も可能になると考える。

3.2.6 特殊な状況

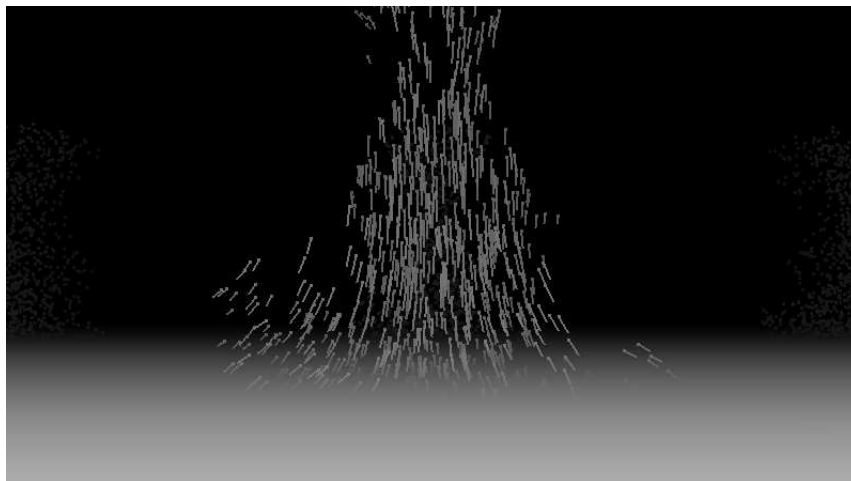


図 3.7: 特殊な状況下における降雪

これまでの例では雪粒子が空気粒子に与える影響があまり見られず、その効果がよく分からなかったので、ここではある特殊な状況を意図的に作った。表示はないが、四方の空気粒子を固定して空間を密閉し、雪は中心付近のみ降っている。この状況下で計算を行うと中心付近の空気が下に流され地表付近で広がり、雪が裾を広げるように降る様子が確認できた。現実には想定しにくい状況だが、このような演出効果も可能であった。

3.3 空気モデルの検証

斥力のみによって空気の挙動を表現することは、擬似的にはあるが空気の乱れや回り込む風なども再現でき、ある程度の実用できる可能性があることが確認できた。

本プログラムでは

- 空間の広さ
- 雪・空気の挙動
- 移動物・建物等のスケール
- 実行速度

を見ながら最もバランスがとれていると思われるものを探し、各パラメータを設定している。特に空気粒子の数、斥力の大きさは雪の挙動のベースとなるもので設定が難しかった。空気モデルの特性としては、1タイムステップの計算で影響範囲の半径 r_e より遠い範囲には影響が及ばないという性質があり、空気粒子の影響が伝わる速さに影響する。また、1タイムステップで空気粒子がほかの空気粒子とあまりにも接近すると計算が破綻してしまう問題がある。よって、計算時間間隔 Δt はある程度短くとる必要があるが、コンピュータの性能上難しい場合は粒子間隔を広げるしかない。本論文ではこれまで空間の具体的なスケールについて明言してこなかった。本プログラムにおける無風状態での雪の落ちる速さは、空気粒子の初期配置における間隔を1とすると、1秒間に1/4程度である。これは、仮に雪の落ちる速さを1m/sという速さだと仮定すると、空気粒子間隔は約4mとなり、空間の広さは縦50m、横50m、高さ30mほどということとなる。よって移動物や構造物を構成する単位は約4mということになり、これは現実世界に適用するには少々大きい。

またそれ以外の問題点として計算の不安定さがある。斥力によって粒子間隔を保つという計算はそれなりの計算速度を確保できたが、空気粒子の動きが安定し

にくく状況によっては粒子が近づきすぎ計算が破綻することがあった。この不安定さは表現対象が降雪のため逆にリアル感におけるメリットになっている。しかし現実的にはもっと安定した計算方法が望ましいと考える。

以上のように、実用的な処理をするにはまだまだ工夫が必要である。

第 4 章

まとめ

本手法を用いることで、様々なシチュエーションにおいて雪が風の影響を受ける様子を表現できた。使用している空気モデルは数値流体力学からは少し離れており、計算精度もあまり高くないが、演出・臨場感の面では大きな効果が期待できる結果となった。実行速度など現時点ではまだ課題が残るが、コンピュータ処理性能の向上にはめざましいものがあり、風を考慮することのメリットはさらに大きくなるだろうと考える。

謝辞

本論文を締めくくるにあたり、終始温かいご指導、適切な助言をくださいました渡辺大地講師および和田篤氏に心から感謝の意を表します。また気軽に相談に応じ励ましをくれた研究室メンバーに深く感謝いたします。

最後に、温かく見守ってくれた家族に感謝いたします。

参考文献

- [1] Havok.com, "Havok Extra",
<<http://www.havok.com/xtra/>>.
- [2] 村岡一信, 「降雪・積雪景観のビジュアルシミュレーションに関する研究」,
岩手大学博士論文, 1999.
- [3] BigBird, 「雪積もるスクリーンセーバー」,
<http://members.jcom.home.ne.jp/big.bird/soft_snow.html>.
- [4] 矢川元基編著, 『パソコンで見る流れの科学』, 講談社, 2001.
- [5] 蔦原道久、片岡武、高田尚樹, 「格子気体法・格子ボルツマン法 - 新しい数値
流体力学の手法」, コロナ社, 1999.
- [6] 越塚誠一著, 『数値流体力学』, 培風館, 1997.
- [7] 石綿良三著, 『流体力学入門』, 森北出版, 2000.
- [8] 日本流体力学会 -Japan Society of Fluid Mechanics-,
<<http://www.nagare.or.jp/>>.
- [9] Welch, J. E., Harlow, F. H., Hannon, J. P., and Daly, B. J., "The MAC
Method - A Computing Technique for Solving Viscous, Incompressible, Tran-
sient Fluid-Flow Problems Involving Free Surfaces" LA-3425, 1966.

- [10] Daly, B. J., Harlow, F. H., Welch, J. E., Sanmann, E. E., and Wilson, E. N., "Numerical Fluid Dynamics Using the Particle-and-Force Method", LA-3144, 1965.
- [11] 越塚誠一, 「粒子法による流れの数値解析」, 日本流体力学会 *ながれ* 21 巻 3 号, 230-239, 2002.
- [12] 粒子法 Particle Method Top Page,
<<http://www.tokai.t.u-tokyo.ac.jp/~rohonbu/lab/mps/mps-j.html>>.
- [13] Koshizuka, S., Tamako, H., and Oka, Y., "A Particle Method for Incompressible Viscous Flow with Fluid Fragmentation", *Computational Fluid Dynamics J.* 4, 29-46, 1995.
- [14] 渡辺大地, "Fine Kernel Tool Kit System",
<<http://www.teu.ac.jp/media/~earth/FK/>>.
- [15] 大森祐, 「複数基準によるシリアライズを利用した衝突判定の高速化アルゴリズムに関する研究」, 東京工科大学卒業論文, 2003.