

氷・霜の広がりを表現するビジュアルシミュレーション

天野 幹子 (指導教員：伊藤 貴之)

1 はじめに

3DCG を用いた映像がメジャーになり、映像制作を支援する様々なツールが開発されている。特に映画などの映像において VFX は表現の幅を広げ、映像を盛り上げる上で欠かせない存在となっているが、制作者が一から VFX の映像を作り上げていくのは大変な作業である。本研究では、VFX を表現するビジュアルシミュレーションの中でも、氷・霜の広がる様子を 3DCG で表現するスケッチインタフェースを提案する。

2 結晶の樹枝状成長

2.1 結晶の形状

Kim[1] が考察している通り、氷の結晶の形は成長の際の大気の状態によっても異なり、正確な幾何学構造は非常に複雑である。しかし、ほとんどの氷の結晶が樹枝構造と扇形板の連続で構成されており、それらの連続によって多くの種類の結晶構造が生まれる。また、霜の幾何学的特徴として、境界付近に樹枝状の成長がみられる。境界から遠い場所では氷が連続するプレートを作っており、視覚的には多くの氷の結晶が成長して接合したものが霜であると考えることができる。

2.2 結晶シミュレーション

氷や霜は、潜熱などの熱伝導による自由エネルギーの変化で結晶が樹枝状に成長する。具体的には Gibbs-Thomson 効果や Mullins-Sekerka 不安定性、また異方性などによって樹枝状の形が形成される。この結晶が成長する過程の界面移動に関する自由境界問題は非線形で非常に難しい。凝固成長についての研究は活発に発表されてきたが、中でもフェーズフィールド法は、計算アルゴリズムが比較的単純で、画期的な方法として注目されている。結晶の成長シミュレーションについては Kobayashi[2] が初めてフェーズフィールドを用いた結晶のシミュレーションに成功した。また、Kim[1] はグラフィックスにおいてこれに DLA を加えた方法を提案している。シミュレーションを用いたスケッチインタフェースにおいては関連研究で挙げる Lipton ら [3] の非構造化セルオートマトンを用いたものがあるが、物理的なこれらのシミュレーションを取り入れたものは管見の限り見られない。現在我々は結晶の成長を表すにあたり、単純で計算の速い拡散方程式のみを用いた方法をとっている。今後、より物理的に正確で細かな結晶の樹枝状の広がりの表現を可能にするフェーズフィールド法の導入を検討している。

3 関連研究

氷・霜を CG で表現する手法は多数発表されているが、我々が目指す手法に近いものとして Lipton らの手法 [3] があげられる。この手法ではアーティストが色を用いて広がりの速さ、形をペイントする。その形に広がるポイントについて非構造化セルオートマトンのシミュレーションにより速度場を生成し、この速度

場を用いてポイントを動かした後にエフェクトの線として用いている。本手法とはユーザによる広がりの設定方法、シミュレーションの方法が異なる。

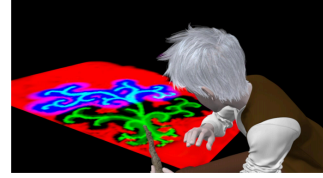


図 1: Lipton らの手法 [3] による氷・霜のデザイン

4 提案手法

本章では、本研究の提案手法を処理手順を追って説明する。

4.1 処理手順

処理手順は以下のとおりである。まず氷・霜が広がる領域の境界線と始点をスケッチ入力する。次に指定した領域内部に点群（以下ポイントと称する）を自動配置し各値を付与する。点群に付与した値を用いてシミュレーションを実施し、最後に結晶を生成する。なお本研究では CG 制作ソフト Houdini を用いている。

4.2 スケッチ入力

本手法のスケッチ入力には Houdini のノードを用いる。Houdini の Group Paint により、オブジェクトを見ながら色を塗るように広がりの領域を指定する。その指定した領域内に自動配置された点群から、広がりの始点となるポイントをユーザが選択する。これにより、ユーザによる単純なスケッチ入力で広がりの領域と始点を指定する。

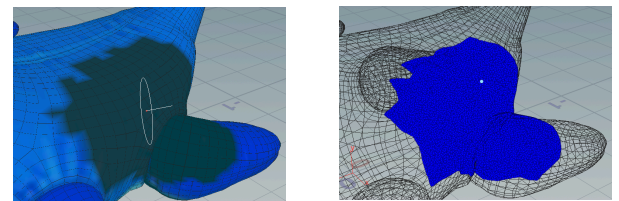


図 2: 左：広がりの指定/右：始点の指定

4.3 拡散シミュレーション

指定した領域に自動配置したポイントを用いてシミュレーションを行う。本研究では、Reiter らの手法 [4] の拡散方程式を用いている。これは 2D の結晶の生成に拡散方程式

$$\frac{\partial u}{\partial t} = a \nabla^2 u$$

を用いているものであり、ラプラシアンを六角格子について次の (1) 式のように近似する。

$$\nabla^2 u \approx \frac{2}{3} (-6u(t, P) + \sum_{N \in nn(P)} u(t, N)) \quad (1)$$

この(1)式において, t は time, P は position, $nn(P)$ は隣接ポイントセットを表し, 本研究では u はポイントが持つ値として初期値をユーザが任意に与える. 時間に伴う u の値の変化は α を用いて(2)式のように近似される.

$$u(t+1, P) \approx u(t, P) + \frac{\alpha}{12} (-6u(t, P) + \sum_{N \in nn(P)} u(t, N)) \quad (2)$$

(2)式が適用される処理の流れは次の通りである. まず, 広がり領域に配置したポイントを Receptive と Non-Receptive の二つの状態に分ける. ここで, Receptive は値が 1 以上のポイントとその隣接ポイント, Non-Receptive はポイントの値が 1 未満で, また 1 以上のポイントにも隣接していないポイントとする. 本研究では, 広がり領域の始点となるポイントの値を 1 としてシミュレーションを開始する. 二つの状態に分けたそれぞれの処理について図3に示す.

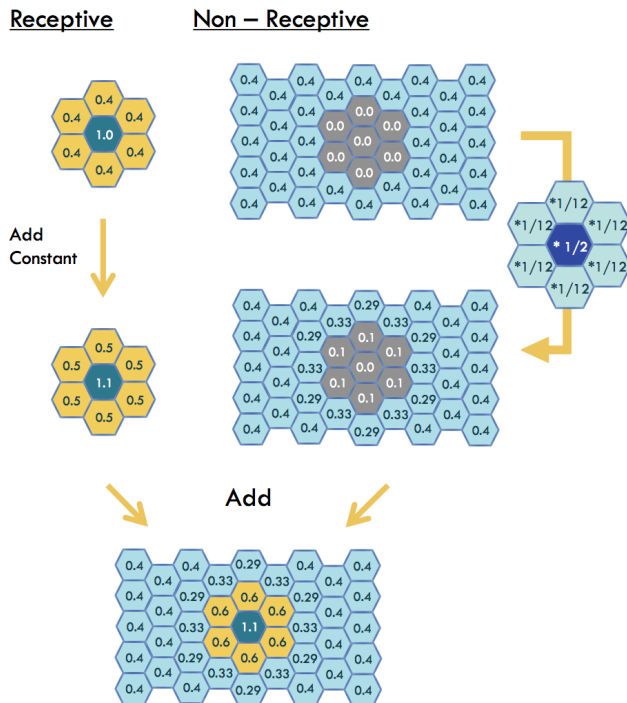


図 3: 小領域でのポイント処理のアルゴリズム

Receptive についてはユーザが指定する定数を加える. Non-Receptive については(2)式を適用する. これにより, 任意のポイントとその周りのポイントとの値の平均を, 中心の $1/2$ 倍, その周りの $1/12$ 倍をとって処理する. その後, それらの二つの計算結果を加算する. これにより値が 1 以上となる点を氷になるポイントとし, 一連の処理を繰り返す.

4.4 結晶生成

シミュレーション結果をもとに結晶を生成する. シミュレーション結果のポイントに六角柱を配置し, それをボリューム化した後, サーフェスを法線方向の外側に膨張させ, 最後に再びポリゴンに変換する. これにより, 六角柱を配置したのみの場合と比べ, 連続性のある結晶の表面を表現する.

5 実行結果

以上の処理によって氷・霜の広がりを表現した結果の途中のフレームを図4に示す. 広がり領域の始点を2点指定してシミュレーションを行っている.

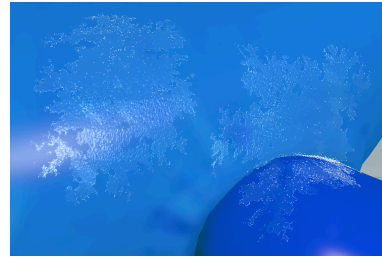


図 4: 実行結果

樹枝状成長の特徴は捉えているものの, 細かな広がりなど現実の氷・霜の形状の再現には及ばない結果となっている. ここでは一通りの結果のみ示すが, パラメータの変更により, ユーザによる形状の変化が可能であることは確認できた.

6 まとめと今後の課題

本研究では, 氷・霜の広がりを表現する映像の制作にあたって, スケッチ入力とシミュレーションを用いてインタラクティブに映像を作成するツールを提案した. シミュレーションについて, 今後 Kobayashi[2] や Kim[1] が取り入れているフェーズフィールド法を導入し, 物理現象を考慮したシミュレーションによってより繊細な結晶の樹枝状成長の表現を可能にする. また結晶の生成について, 現在は六角柱の配置という方法を採用しているが, より質感を意識したモデリングを適用したいと考えている. 半透明で複雑な氷・霜の光反射を考慮したレンダリングを導入するなど, より現実の氷・霜に近い表現を考慮した処理を適用する. それらの表現に加え, 速さなどの広がり制御性についても今後の課題である.

参考文献

- [1] Theodore Won-Hyung Kim, Physically-based simulation of ice formation, University of North Carolina at Chapel Hill, 2006, Ph.D. thesis.
- [2] R. Kobayashi, Modeling and numerical simulations of dendritic crystal growth, Physica D, 63, 1993, pp. 410-423.
- [3] D. Lipton, K Museth, B Sutherland, DW Animation, Jack's Frost: Controllable Magic Frost Simulations for 'Rise of the Guardians', ACM SIGGRAPH 2013 Talks.
- [4] Clifford A. Reiter, A local cellular model for snow crystal growth, Chaos, Solitons & Fractals, vol.23, 2005, pp. 1111-1119.
- [5] Creating Procedural Snowflake in Houdini by Entagma, <https://www.3dtut.com/portfolio-items/creatingproceduralsnowflakeinhoudiniby-entagma/> (最終閲覧日 2018 年 12 月 16 日).