

氷・霜アニメーション生成のための外形入力と 初期値最適化による成長速度制御

天野幹子¹⁾(非会員) 佐藤周平²⁾(非会員) 伊藤貴之¹⁾(正会員)

1) お茶の水女子大学 2) 富山大学

A Growth Control by External Shape Input and Optimization of Initial Parameters for Animation of Ice and Frost

Mikiko Amano¹⁾ Syuhei Sato²⁾ Takayuki Itoh¹⁾

1) Ochanomizu University

2) University of Toyama

{amano.mikiko, itot}@is.ocha.ac.jp, ssato@eng.u-toyama.ac.jp

概要

本論文では、映像制作に欠かせないビジュアルエフェクツの表現の中でも、氷や霜がオブジェクトに張り付いて広がる様子を表現する一手法を提案する。本手法の主要な要素技術は外形入力と初期値最適化である。本手法では平面のオブジェクト上においてユーザが入力した「広がり範囲」と「始点」に沿ってシミュレーションを実行する。ここで最急降下法を導入してシミュレーションの初期値を最適化することで、ユーザの所望の速度での広がりを実現する。これにより、始点から指定した範囲内にユーザの所望の速度で氷・霜が樹枝状に広がるアニメーションを制作できる。

Abstract

This paper presents a technique for the creation of visual effects that ice and frost spread over arbitrary objects. External shape input and optimization of initial parameters are key technical components of the presented technique. The technique performs a physical simulation based on the spread domain and starting points specified by the user. The technique realizes the simulation of the spread of ice and frost in the user-specified time periods by optimizing the initial parameters adopting the steepest descent method. Users can create animations of the spread of ice and frost with user-specified starting points, spread domain, and spread speed.

1 はじめに

3次元コンピュータグラフィックス（以下3DCGと称する）を用いた映像がメジャーになり、ビジュアルエフェクト（以下VFXと称する）は映画やコンピュータゲームなどに広く適用されている。自然現象を現実であるかのように再現するアニメーションによって、多様なシーンにおいてリアリティの高い映像が制作されている。近年では、VFXの制作をサポートするために、多くのソフトウェアやツールが開発されている。

氷と霜の結晶化は、VFXの中でも重要な自然現象の1つである。いくつかのシミュレーション方法 [1–3] が3DCGにおける氷や霜の形成手法として提案されている。これらの方法は氷や霜の形成のリアルなアニメーション制作を可能にする。しかしこれらの方法では、ユーザが所望の映像を容易に制作できることに焦点を当てておらず、それを実現するためにはシミュレーションのパラメータを試行錯誤的に調整する必要がある。これにはシミュレーションの実行とパラメータの調整作業を何度も繰り返す必要があり、多くの時間と労力を要する。この問題に対し我々は、ユーザが望む形状の氷の形成アニメーションのデザインを、外形を入力することで手軽に実現する研究に取り組んでいる。本論文では外形の指定方法にスケッチ入力を用いることで、ユーザは簡単に外形の指定・編集ができる。また、氷や霜の成長のランダム性を保ちつつ、アニメーション全体の成長速度を制御する手法も提案する。本手法により、ユーザはいくつかのパラメータを調整するだけで、多様なシミュレーション結果を得ることができる。また本手法では、アニメーション全体としての氷・霜の成長速度を指定可能とするために、最急降下法によってシミュレーションの初期値を最適化する。これによりユーザは、細かいパラメータ調整をせずに、想定するフレーム数でアニメーションが終了するような初期値を設定することができる。本稿の内容は、過去の我々の発表 [10] と同一内容である。

2 関連研究

2.1 結晶の樹枝状成長

氷や霜の結晶の視覚的な形状の特徴として樹枝状構造があげられる。氷や霜の広がり表現するにあたってこの特徴は非常に重要であり、この構造を再現すること

によって本研究で表現するVFXが特徴づけられると考える。

Reiter [2] は、3つのパラメータの調整のみで雪の結晶の多様な形状を表現可能とした。この研究のシミュレーションは物理的な熱伝導などを考慮するものではないが、実際の雪の結晶の写真と比較しても遜色ない形状表現から、実際の雪よりも複雑な、芸術的形狀まで表現可能にしている。本手法ではこのReiterの手法を結晶モデルとして適用する。なお、この手法は六方向に伸びる氷の結晶成長を表現するものであり、それらが結合した霜のようなランダムな成長を表現するものではない。我々はここに乱数を導入することにより、ランダム性の表現を可能にする。

Reiterの手法が雪の結晶を表現するものであったのに対し、オブジェクトに霜が張る様子を表現する手法がいくつか提案されている。Kim [1] はフェーズフィールド法と拡散律速凝集 (DLA) を組み合わせ、オブジェクトに霜が張るアニメーションを生成する手法を提案している。フェーズフィールド法とは、物質の凝固過程の自由境界問題であるStefan問題を扱いやすくモデル化したものである。この手法では、凍結温度のマップを作成し、それに応じて結晶成長のしやすさが変化する。一方でRen [3] は、フェーズフィールド法に分子の配向の自由度を追加したモデルによって広がりの方角をコントロールする手法を提案している。この手法は配向のマップを作成することで、オブジェクトに張る霜の成長の方角を決定する。

これらの手法は、始点となる点から樹枝状に伸びる結晶成長について、成長のしやすさや方向を指定してシミュレーションを実行するものであり、最終的な広がり形状を明示的にデザインするものではない。その代わりに、シミュレーションの実行に先立って結晶の広がりを表すマップを生成しておく必要がある。この方法では、ユーザが最終的な広がり形状を直感的に入力・調整しにくい点が問題である。

2.2 結晶成長表現のためのスケッチインタフェース

氷や霜の結晶を3DCGで表現する研究は既にいくつか発表されている。その中でもスケッチ入力を用いるものとしてLiptonらの手法 [4] があげられる。この手法ではアーティストが広がり速度と形状を色ごとに分けてペイントする。続いて、その形状に沿って拡散する点

群(以下ポイントと称する)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する。この速度場を用いてポイントを動かした軌跡をエフェクトの線として用いている。しかし、この手法では樹枝状結晶として氷・霜をシミュレーションしていないため、生成される結果の写実性は前述の3つの手法に比べて劣る。

本研究では、写実的な氷・霜をシミュレーションでき、生成される形状の変化に対して直感的なパラメータを持つ Reiter の手法を採用し、そのシミュレーションを制御する手法を提案する。

2.3 CG と最適化法

水や煙、炎などの複雑な現象は手作業で作成するのは非常に困難であり、シミュレーションが使われてきた。特に Fedkiw らは流体シミュレーションの従来手法を踏まえた効率的なシミュレーション手法を提案し、CG の分野における自然現象シミュレーションを発展させた [5] [6] [7]。しかし、そのようなシミュレーションにおいてユーザの思い通りの結果になるようパラメータを制御することは難しい。物理シミュレーションの抽象化に加え、最適化を導入することなどによりユーザ所望のアニメーションをより簡単に制作する方法が検討されてきた [8] [9]。

本手法では、ユーザがパラメータ設定を試行錯誤することなく意図する速度での広がりを実現可能とするために、最適化を導入する。本手法で採用するのは制約なし非線形最適化法である。この手法は目的関数の勾配を利用してその値が小さくなる方向に計算を繰り返す反復手法である。代表的な手法として、最急降下法、ニュートン法、準ニュートン法があげられる。現時点での我々の実装では、安定性が高く実装しやすいという理由により、最急降下法を適用している。どの最適化法が本研究において最適であるか、今後も検討する必要がある。

3 氷・霜アニメーション生成のためのスケッチ入力と初期値最適化

本研究では、手続き的なシミュレーション方法を採用することにより、単純なスケッチ入力と直感的なパラメータの調整のみで氷・霜の広がりを表現する。まず、ユーザのスケッチ入力をもとにシミュレーション領域を設定する。そして、Reiter の手法 [2] を用いてシミュ

レーションを実行する。本稿では Reiter の手法に乱数を導入することで、氷の広がりにより多様性を持たせられるよう拡張する。また、ユーザは任意の最終フレームを指定し、そのフレームに氷がシミュレーション領域の端に到達するよう最急降下法によりパラメータを最適化することで、所望の速さでの広がりを実現する。現在は、3D のオブジェクト上にユーザがペイントしたものを展開したと仮定し、二次元の任意の領域に均等に配置した格子点に対し処理を適用しており、今後はその結果を3D のオブジェクトにマッピングするという形で応用することを検討している。なお本研究ではCG制作ソフト Houdini *1 を用いて提案手法を実装した。

3.1 スケッチ入力と格子点配置

ユーザはまず、2次元平面上にペイント操作を通して氷や霜の広がりの範囲を入力する。次にユーザによって指定された領域内部に均等にシミュレーション空間を離散化するためのポイントを配置する。本手法では Reiter の手法 [2] をベースにシミュレーションを実装することから、一様な六角形格子配列を構成するようにポイントを配置する(図1参照)。このとき、スケッチ入力の領域からひと回り大きくポイントを配置する。この外周のポイントは境界条件を計算する際に使用する。ユーザは次に、領域内に配置したポイントから始点となる点(seed point)を選択する。なお、始点を複数指定しても問題なくシミュレーションを実行できる。

3.2 拡散シミュレーション

上述のように設定した領域においてシミュレーションを実行する方法について述べる。蒸気の場合 $u_t(\mathbf{p})$ を氷の状態を表す値として定義する。ここで、 t はフレーム番号、 \mathbf{p} は格子上的位置を表す。まず初期フレームにおいて、蒸気の場合の各格子点にユーザ指定の初期値 $u_0(\mathbf{p}) = \beta$ を割り当てる。次に、各フレームにおける蒸気の場合の状態に応じて各格子点を「ice 格子」「ice 格子の隣接点」「それ以外の点」の3つの状態に分類する(図1参照)。 $1.0 \leq u_t(\mathbf{p})$ を満たす格子点が ice 格子として分類され、この ice 格子が氷・霜として可視化される。各フレームでの計算の過程において、各 ice 格子とその隣接点には定数値 γ を加算する。一方、ice でもその隣接点でもない格子点では、蒸気の場合に対して拡散方程式

*1 <https://www.sidefx.com/ja/products/houdini/>

を計算する。この拡散方程式の計算において、ice とその隣接点の格子点については値を 0.0 として計算を実行する (詳細については [2] を参照いただきたい)。最後にこれら 2 つの結果を組み合わせ、その結果を使用して次のフレームでの蒸気場を計算する (図 1 参照)。しか

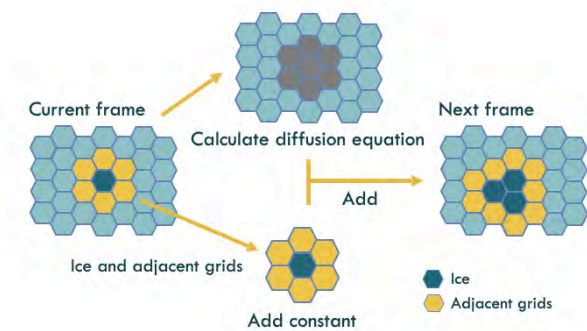


図 1 小領域での Reiter の手法による結晶成長アルゴリズム。

し、Reiter の手法では霜の成長のランダム性を表すことはできない。我々はこの問題を解決するために擬似乱数を導入することにより、Reiter の手法を拡張する。この手法では、ice の隣接点である各ポイント (図 1 黄色の格子) に擬似乱数 ($0.0 \leq r_t(\mathbf{p}) \leq 1.0$) を割り当てる。また、ユーザが閾値 r_{th} を設定する。そして、定数値 γ を r_{th} よりも大きい乱数値を持つ隣接点にのみ加算する (式 (1))。ここで式 (1) の $r_t(\mathbf{p}) \leq r_{th}$ の場合の式は、Reiter の手法による ice でもその隣接点でもない格子点に適用される拡散方程式である。 α はユーザ指定の重み付けの係数で、広がり形状に影響を与える。 $nn(\mathbf{p})$ は隣接ポイントセットを表す。

$$u_{t+1}(\mathbf{p}) = \begin{cases} u_t(\mathbf{p}) + D_t(\mathbf{p}) & (r_t(\mathbf{p}) \leq r_{th}) \\ u_t(\mathbf{p}) + \gamma & (r_t(\mathbf{p}) > r_{th}) \end{cases} \quad (1)$$

$$D_t(\mathbf{p}) = \frac{\alpha}{12} (-6u_t(\mathbf{p}) + \sum_{\mathbf{q} \in nn(\mathbf{p})} u_t(\mathbf{q}))$$

閾値の値を大きく設定するほど、複雑な成長の表現が可能になる。この拡張方法により Reiter の手法と比べ、氷のさまざまな成長パターンを表すことができる。

3.3 境界条件

ユーザによるスケッチ入力に対し、ひと回り大きく格子点を配置することによって、領域の外周のポイントを境界としてシミュレーションを実行する。境界のポイン

トは ice 格子として設定されることはなく、境界ポイントの $u_t(\mathbf{p})$ には隣接する 6 点のうち境界ポイントでないポイントの $u_t(\mathbf{p})$ の平均をとった値を与えて計算を実行する。

3.4 初期値の最適化

本手法では、シミュレーションの初期値 β を最適化することで、氷の成長速度を制御する。現時点の我々の実装では、最適化の方法として最急降下法を適用している。シミュレーションにおいて各ポイントに与える初期値 β が変わると、各ポイントの持つ蒸気場 $u_t(\mathbf{p})$ のシミュレーション過程の値、ひいては広がり速度が変化する。この初期値 β を最適化することで、ユーザの所望のフレームで氷がスケッチした領域の端に到達するように制御する。

これを実現するために本研究では、最適化の目的関数を以下のように定義する。

$$f(\beta) = (T - T_{sim}(\beta))^2 \quad (2)$$

T は目標終了フレームを表し、ユーザが直接指定する値である。 $T_{sim}(\beta)$ は初期値 β でシミュレーションを実行した際の終了フレームを表す。本手法では、それらの差の二乗が最小になるような初期値 β を求める。現時点での我々の実装では、ice となったポイント数が全体のポイント数の 9 割以上となるフレームを終了フレームとしている。ここで、最急降下法の各ステップにおける更新式は以下ようになる。

$$\beta^{i+1} = \beta^i - \eta \frac{\partial f(\beta^i)}{\partial \beta} \quad (3)$$

ここで、 i は最急降下法のステップ数であり、 η はユーザ指定の係数である。そして、目標終了フレームとシミュレーション結果の終了フレームの差が十分小さくなった場合に最適化計算を終了し、その時点の β^i を最終的な初期値として採用する。現時点での我々の実装では、誤差がアニメーション再生時の時間において 1/6 秒以下であれば、その誤差は十分に小さいと判断している。そのため、シミュレーションにおいて設定した 1 フレーム当たりの時間幅から 1/6 秒に相当するフレーム以内となった場合に、シミュレーションは収束したとみなしている。

3.5 結晶の生成

3.2 節で述べた通り、本手法では ice 格子として分類された格子点の集合を、氷・霜が生成された位置の集合と

して出力する．現段階の実装では，ice となった格子点 (2 の青いポイント) に六角柱を配置して立体化することで，結晶を生成する (図 3 参照)．ここで，配置する六角柱の高さは格子点ごとに変更可能である．我々の実装では原則として，全ての六角柱の高さを等しくすることで，平坦な霜の広がりを表現している．一方で，六角柱の高さを任意に調整することで，不規則な凸凹を有する広がりや，始点を中心とした山状の広がりなどの表現も可能である．

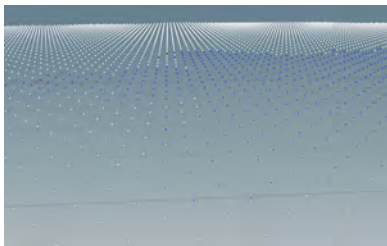


図 2 シミュレーション途中の格子点の拡大図．ice 格子である青いポイントに六角柱が配置される．

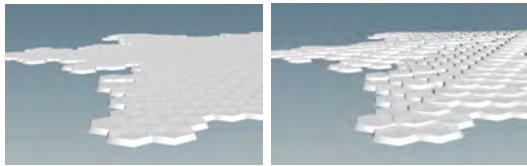


図 3 ice 格子に六角柱を配置した様子．(左) 等しい高さの六角柱を配置．(右) 高さに変化を持たせた六角柱の配置．

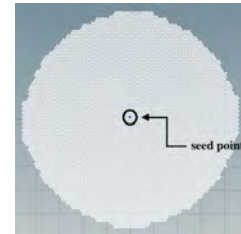
4 実験結果

提案手法による結果を示す．結果の可視化には，Houdini のレンダラである Mantara を使用した．各実験結果に対するシミュレーション条件やパラメータ設定については以下で個別に記載する．

4.1 擬似乱数を導入したシミュレーション

擬似乱数を導入した氷・霜の広がりのシミュレーション結果を図 4 に示す．図 4 では，始点を 1 点指定した場合の結果を示している． $\beta = 0.35$ ， $\gamma = 0.005$ のとき，ランダム性を表現する乱数の閾値 r_{th} が 0.0，0.8 のそれぞれの場合について，20 フレーム，40 フレーム，60 フレームの結果を示す．閾値が 0.0 の場合には従来手法で表現されている均等な氷の結晶成長の様子が表現できている．閾値を大きくすると，閾値 0.0 の成長と比較して

広がりの方向が一様でない，ランダム性のある成長を表現することが可能となった．



20 frame 40 frame 60 frame

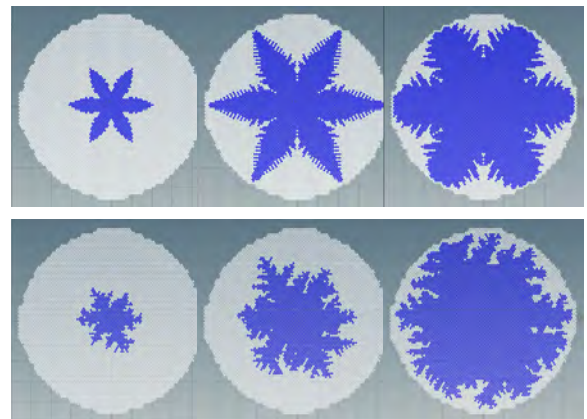


図 4 各フレームでの乱数の閾値ごとのシミュレーションの様子．(1 段目) シードの位置．(2 段目) $r_{th} = 0.0$ の結果．(3 段目) $r_{th} = 0.8$ の結果．

4.2 多様な形状に対するシミュレーション結果

本節ではユーザによる指定の自由度を示すために，単純な円ではなく複雑な形状を指定した場合や，広がりの始点を複数指定した場合に，我々の疑似乱数による拡張を行ったシミュレーションにおいて破綻が起こらないことを確認する．

まず，ドーナツのように中心に穴のあいた形状で，始点が 1 点でのシミュレーション例を図 5 に示す．境界条件を指定したことによって，広がりの形状が偏ることなく，樹枝状の広がりを表現することができている．また，始点が 2 点でのシミュレーション例を図 6 に示す．

4.3 最急降下法による最適化

次に最適化の実験結果を示す．乱数 r_{th} の閾値が 0.0 の場合に目標終了フレームを 130，ステップ幅 η を 1.0×10^{-6} として最急降下法を適用した結果が表 1 および図 7 である．また，乱数 r_{th} の閾値が 0.8 の場合に目標終了フレームを 130，ステップ幅 η を 1.0×10^{-6} として最急降下法を適用した結果を表 2 および図 8 に示す．

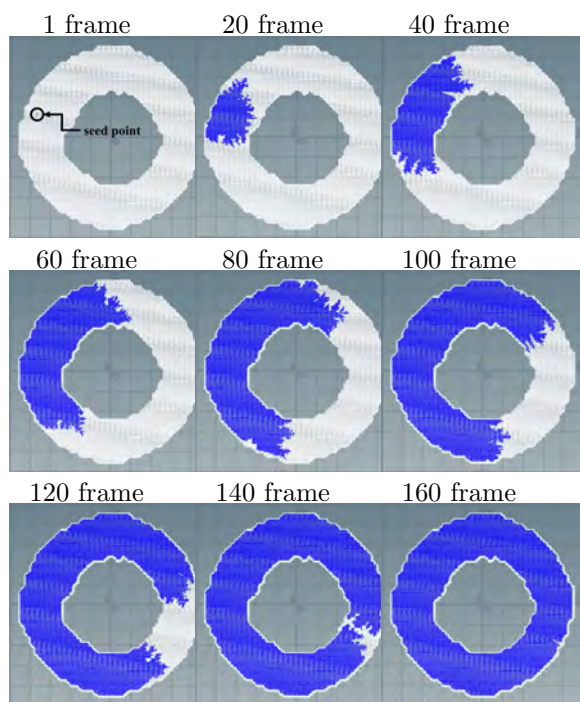


図5 ドーナツ形・始点1点のシミュレーション実行例.

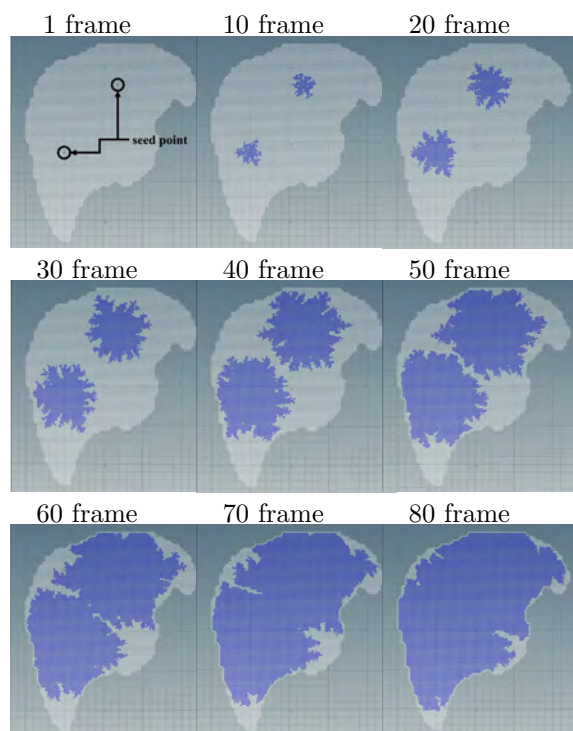


図6 歪な形状, 始点2点のシミュレーション実行例.

表1 閾値 0.0 の最急降下法による初期値

ステップ	初期値	終了フレーム
0	0.25	160
1	0.261856	152
2	0.295357	139
3	0.307597	126

表2 閾値 0.8 の最急降下法による初期値

ステップ	初期値	終了フレーム
0	0.25	162
1	0.294132	135
2	0.316651	124
3	0.315713	127

本実験では作成される映像が 24fps となるようにシミュレーションの時間幅を設定しており, この場合最適化の終了条件は目標終了フレーム ± 4 フレームの間となる. 閾値が 0.0 の場合, 0.8 の場合のいずれも, 最急降下法により終了フレームが目標終了フレーム ± 4 フレームの間となるような初期値を探索できていることを確認できた.

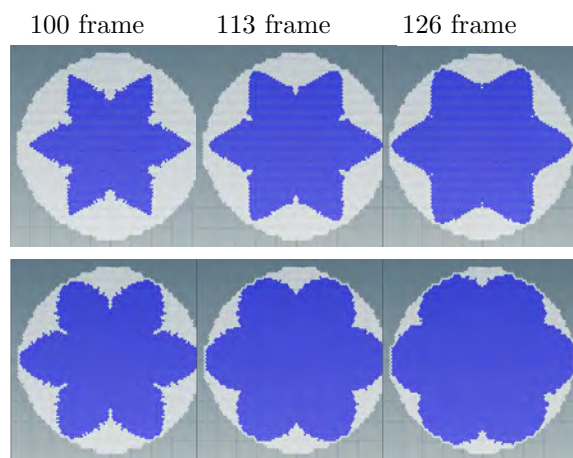


図7 表1におけるステップ0およびステップ3の各フレームでのシミュレーションの様子.

4.4 結晶生成方法の比較

3.5 節にて論じた通り, 現時点での実装では六角柱の高さを等しくしている. 図4をレンダリングした結果を図9に示す. 霜の広がりが見えが, 滑らかなアニメーションを実現できている.

一方で, 六角柱の高さをランダムに変化させた結果を図10に示す. この結果では, 結晶形状の視認性を高めるためにオブジェクトを赤色にしている. この結果のほ

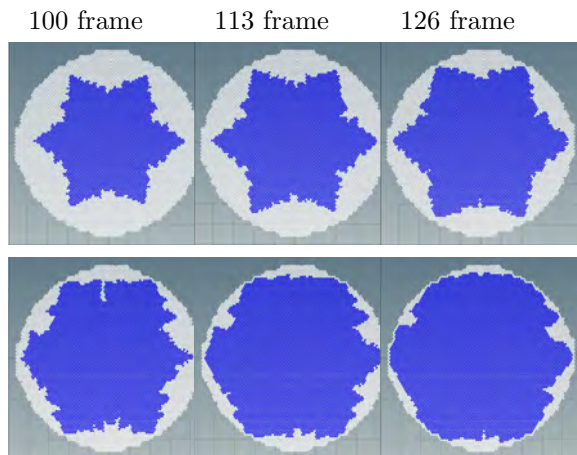


図8 表2におけるステップ0およびステップ3の各フレームでのシミュレーションの様子。

うが立体感をもたらすことができる。一方で、シームレスなアニメーションを実現できず、フレームごとにながつきが生じてしまうことがある。この問題の解決手段については次章にて議論する。

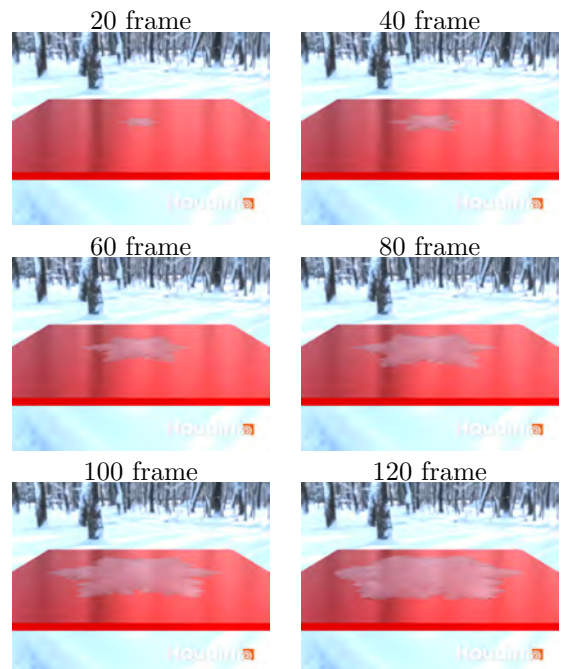


図10 六角柱の高さに変化を持たせた場合のレンダリング実行例。

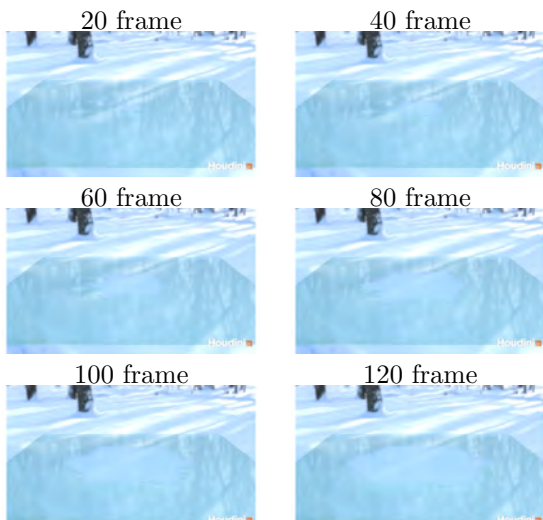


図9 六角柱の高さが均一の場合のレンダリング実行例。

5 考察と課題

本手法によって、従来手法 [2] と同様な六方向への広がりから、ランダムウォークのような樹枝状の広がりまで、ともに表現可能であることが確認できた。また、本手法によって円形状の入力に対しては樹枝状成長を表現できた。

一方で、最急降下法を用いた初期値の最適化について

は課題が残った。4.3節で紹介したような複雑な形状で初期値と終了フレームの関係にばらつきが生まれる原因として、初期値の最適化によって広がり形状が変化することが考えられる。これは3.5節で述べた結晶生成の問題にも関係する。初期値のみを変更すると、広がりにおいて全く同一の形状を維持することができない。そのため、iceになっていない格子点の残り方が初期値に依存することとなる。

上記に関連し、提案手法では終了フレームを大きく変更すると β の値が大幅に変化し、結晶の詳細な形状が大きく変わってしまう場合がある。また、 $0 \leq \beta < 1$ であるため、終了フレームとして極端な値 (1 や 10000 など) を指定してもその通りに制御ができない場合がある。これらの問題の解決には、他のパラメータ (α や γ , r_{th}) も同時に最適化することが有効な可能性があるため、今後検討していく予定である。

我々の実験では、大域的な β の変化と終了フレームの間にはおおそ比例するような関係があるが、 β の微細な変化によってその関係が逆転する場合があることを発見した。特に終了フレームを全ポイントが ice となった時と定義した場合このような現象が発生し、最急降下法がうまく収束しない場合があった。現時点での実装で

は、ice となったポイント数が全体のポイント数の 9 割以上となるフレームを終了フレームとすることで、この問題を回避している。しかしこの実装では、意図通りの形状の氷が得られる前に終了フレームとなってしまう場合がある。そのため、根本的にこの問題を解決するためには最適化の手法を検討する必要があると我々は考えており、今後は最急降下法以外の非線形最適化法を用いて実験を行う予定である。

また、現時点での実装では初期値 β をシミュレーション空間全体で単一としたが、これを始点からの各方向毎に変えられるよう定義し、どの方向の領域の境界へも氷・霜の広がりへの到達時間が等しくなるように、広がりへの始点から境界までの距離を考慮した最適化を実装したい。

4.4 節では、レンダリングについて 2 つの例を示した。六角柱の高さに変化をつけるとフレームごとにながつきが生じてしまう問題について、配置する格子点の解像度を上げることでシームレスな広がりへ近づくと考えられる。しかし、初期値のみを変更すると広がりにおいて全く同一の形状を維持することができないという問題においてこの対処法は適切でない。格子点が増えることで、この問題がより顕著になってしまう。また、ユーザの表現の自由度も下げる事となる。こちらも前節と同様に複数パラメータの最適化により対処する、あるいはフレームごとの結晶をシームレスに繋ぐ結晶生成方法を検討したい。一方で、現時点での実装では、広がり全体が一つの結晶であるかのように見えてしまうという問題もある。そのため、時間ごとに垂直方向に成長し、レイヤーのように重なって厚みを増す霜の結晶成長の様子をリアリスティックに表現できる結晶生成方法も検討したい。

6 まとめ

本論文では、氷・霜の広がりを表現する映像の制作を目的として、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氷・霜の広がりをデザインするツールを提案した。また、最急降下法を用いて初期値を最適化することにより、ユーザの所望のフレームで氷・霜の広がりを完了するようにシミュレーションの初期値を決定する手法を提案した。

今後の課題として、前章で議論した課題に沿って本研究を拡張したい。また、本稿ではアニメーション全体と

して、氷・霜の成長速度の制御を行ったが、成長速度を時間・空間的にローカルに指定できることもデザインツールとして重要であり [4]、これを実現できるよう今後拡張していく。平面的なオブジェクトにとどまらず、立体的なオブジェクトに対しても本手法を適用することも課題である。

参考文献

- [1] T. W.-H. Kim: *Physically-based simulation of ice formation*, University of North Carolina at Chapel Hill, 2006, Ph. D. thesis.
- [2] C. A. Reiter: *A local cellular model for snow crystal growth*, *Chaos, Solitons & Fractals*, Vol. 23, 2005, pp. 1111-1119.
- [3] B. Ren, J. Huang, M. C. Lin, S.-M. Hu: *Controllable Dendritic Crystal Simulation Using Orientation Field*, *Computer Graphics Forum*, Vol. 37, No. 2, 2018, pp. 485-495.
- [4] D. Lipton, K. Museth, B. Sutherland, *Jack's Frost: Controllable Magic Frost Simulations for 'Rise of the Guardians'*, ACM SIGGRAPH 2013 Talks.
- [5] R. Fedkiw, J. Stam, H. W. Jansen, *Visual Simulation of Smoke*, ACM SIGGRAPH 2001, pp. 15-22.
- [6] N. Foster and R. Fedkiw, *Practical Animation of Liquids*, ACM SIGGRAPH 2001, pp. 23-30.
- [7] D. Q. Nguyen, R. Fedkiw, H. W. Jensen, *Physically Based Modeling and Animation of Fire*, *ACM Transactions on Graphics*, Vol. 21, No. 3, 2002, pp. 721-728.
- [8] A. Treuille, A. McNamara, Z. Popovic, J. Stam, *Keyframe Control of Smoke Simulations*, *ACM Transactions on Graphics*, Vol. 22, No. 3, 2003, pp. 716-723.
- [9] A. McNamara, A. Treuille, Z. Popovic, J. Stam, *Fluid Control Using the Adjoint Method*, *ACM Transactions on Graphics*, Vol. 23, No. 3, 2004, pp. 449-456.
- [10] 天野 幹子, 佐藤 周平, 伊藤 貴之, スケッチ入力を用いた氷・霜アニメーション生成のための初期値最適

化による成長速度制御, 情報処理学会コンピュータグラフィックスとビジュアル情報学研究会 第 180 回研究発表会, No. 16, 2020.

任. 2019 年から同大学文理融合 AI・データサイエンスセンター長兼任. 2014 年より 2016 年まで芸術科学会会長. 情報可視化, マルチメディア, インタラクション, コンピュータグラフィックスなどの研究に従事.

天野幹子



2019 年お茶の水女子大学理学部情報科学科卒業. 2021 年同大学大学院人間文化創成科学研究科理学専攻博士前期課程修了. コンピュータグラフィックスに関する研究に従事.

佐藤周平



富山大学学術研究部工学系助教. 2014 年北海道大学大学院情報科学研究科博士後期課程修了. 博士 (情報科学). (株) ユビキタスエンターテインメント, (株) ドワンゴ, プロメテック・ソフトウェア (株) を経て, 2019 年より現職. コンピュータグラフィックス, 特に流体アニメーションの生成や編集に関する研究に従事.

伊藤貴之



1992 年早稲田大学大学院理工学研究科電気工学専攻修士課程修了, 日本アイ・ピー・エム (株) 東京基礎研究所研究員. 1997 年博士 (工学). 2005 年お茶の水女子大学理学部情報科学科助教授. 2011 年同大学教授. 2017 年まで同大学シミュレーション科学教育研究センター長兼