セルオートマトンを用いた雲のシミュレーションとその表示

2ZC－6

土橋宜典 西田友是[†] 沖田豪
広島市立大学情報科学部 [†]東京大学理学部

1. はじめに

映画やコマーシャルフィルムなどで魅力的な雲の動きを捉えた映像が用いられることは多い。近年、そのような映像をコンピュータグラフィクスにより再現する試みが多く行われており、大気流体の数値シミュレーションを行う手法やパーティクルシステムによる簡易シミュレーションによる手法が提案されている[1][2]。しかし、従来法では、処理が複雑であり、計算時間を多く必要とするという欠点を有する。

本稿では、セルオートマトンを用いた雲のアニメーションを作成する手法を提案する。雲の動きは簡単な規則により表現され、少ない計算量でシミュレーションを行える。さらに、ビルボード処理[3]によりその結果の高速表示を行う。

2. 雲の動きのシミュレーション

2.1 セルオートマトンによる雲の成長モデル[4]

提案手法では、Nagel らによって提案されている雲の成長モデル[4]を利用する。このモデルでは、シミュレーション領域を格子状に分割する。各格子点には、水蒸気の有無(*hum*)、水蒸気から雲への相変化の判定(*act*)、雲の有無(*cld*)を表す 3 つの変数が割り当てられ、1 または 0 の値を取る。これらの変数の時刻 t+1 での値は時刻 t での値を用い、式(1)から(3)に示す遷移規則により計算される。

$$\begin{aligned} act(i,j,k,t+1) = {} & !act(i,j,k,t)\ \&\&\ hum(i,j,k,t) \\ & \&\&\ (act(i+1,j,k,t) \parallel act(i-1,j,k,t) \parallel act(i,j+1,k,t) \\ & \parallel act(i,j-1,k,t) \parallel act(i,j,k+1,t) \parallel act(i,j,k-1,t)) \\ & \parallel act(i+2,j,k,t) \parallel act(i-2,j,k,t) \parallel act(i,j+2,k,t) \\ & \parallel act(i,j-2,k,t) \parallel act(i,j,k-2,t)\ ) \end{aligned} \quad (1)$$

$$cld(i,j,k,t+1) = cld(i,j,k,t) \parallel act(i,j,k,t) \quad (2)$$

$$hum(i,j,k,t+1) = hum(i,j,k,t)\ \&\&\ !act(i,j,k,t) \quad (3)$$

適切な初期状態からこれらの規則を逐次適用することで雲の成長を表現できる。しかし、この方法では、一旦、*cld* が 1 となった格子点はその後、変化が起こらない。すなわち、雲の消滅が起こらない。また、得られる結果は 0/1 であるため、リアルな雲を表示することができない。そこで、これらの点を解決する手法を以下に述べる。

2.2 雲の消滅の表現

雲の消滅を表現するため、新たに変数 *ext* を導入する。*ext* の遷移規則は *act* に準じて式(4)で定義する。

$$\begin{aligned} ext(i,j,k,t+1) = {} & !ext(i,j,k,t)\ \&\&\ cld(i,j,k,t) \\ & \&\&\ (ext(i+1,j,k,t) \parallel ext(i-1,j,k,t) \parallel ext(i,j+1,k,t) \\ & \parallel ext(i,j-1,k,t) \parallel ext(i,j,k+1,t) \parallel ext(i,j,k-1,t)) \\ & \parallel ext(i+2,j,k,t) \parallel ext(i-2,j,k,t) \parallel ext(i,j+2,k,t) \\ & \parallel ext(i,j-2,k,t) \parallel ext(i,j,k-2,t)\ ) \end{aligned} \quad (4)$$

次に、*ext* を用いて、雲を消滅させるため、式(2)で表される *cld* の規則を式(5)に示すよう変更する。

$$cld(i,j,k,t+1) = !ext(i,j,k,t)\ \&\&\ (cld(i,j,k,t) \parallel act(i,j,k,t)) \quad (5)$$

式(4)および(5)から、*ext* は *cld*=1 である格子を cld=0 に変更しながら伝播する。よって、*act* と *ext* により雲の生成・消滅を表現できる。しかし、*cld* の値が頻繁に変更され、雲の生成・消滅が短い周期で繰り返されると不自然な雲の動きとなってしまう。これを避けるため、雲の寿命 T_{ext} を導入する。すなわち、一旦、*cld*=1 となった格子点は、それ以後 T_{ext} ステップの間、状態変化は起こらないとする。これにより、自然な動きを表現できる。

2.3 シミュレーションの開始と雲の動きの制御

シミュレーションを開始するために、各格子点の *hum*, *act*, *ext* の値を指定された確率に従ってランダムに 0 から 1 に変更する。また、シミュレーション途中においてもこれらの変数の値を0から 1 へ変更することで雲の動きを制御できる。例えば、雲を多く発生させたい領域には、*hum* が 1 となる確率を高く設定すればよい。このとき、全ての格子点についてその確率を指定することは煩雑な作業を要するため、指定されたいくつかの代表点についてのみ確率を指定する。各格子点の確率は代表点の確率から補間により求める。代表点の位置や確率を時間的に変化させることで雲の大まかな動きを制御できる。

2.4 連続な分布の算出

連続な雲の密度分布は各格子点にメタボールを配置することで算出する。メタボールは中心ほど高くなる密度関数を割り付けた球である[5]。任意の点 $\mathbf{x}$ の密度 $\rho(\mathbf{x})$ はメタボールの密度関数を用いて次式により表現する。

$$\rho(\mathbf{x}) = \sum_{i,j,k \in \Omega(\mathbf{x})}^{n(\mathbf{x})} q_{i,j,k} f(|\mathbf{x} - \mathbf{x}_{i,j,k}|, D) \quad (6)$$

ここで、$\mathbf{x}_{i,j,k}$ および $q_{i,j,k}$ は、それぞれ、格子点(i, j, k)のメタボールの中心座標および中心密度、また、f は密度関数、D は有効半径である。$\Omega(\mathbf{x})$ は $|\mathbf{x} - \mathbf{x}_{i,j,k}| < D$ を満たす格子点の集合を表し、$n(\mathbf{x})$ はその数である。有効半径 D はユーザにより指定する。中心密度 $q_{i,j,k}$ は式(7)により、0/1 の分布を平滑化することで算出する。

$$q_{i,j,k} = \frac{1}{n} \times \sum_{l,m,n \in \Omega(\mathbf{x}_{i,j,k})}^{n(\mathbf{x}_{i,j,k})} cld(l,m,n) \quad (7)$$

なお、中心密度の値が 0 のメタボールは削除する。また、式(7)から $0 < q_{i,j,k} \le 1$ となる。

Animation and Fast Rendering of Clouds Using Cellular Automaton

Y. Dobashi, †T. Nishita, T. Okita
Hiroshima City University, †The University of Tokyo