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。

$$act(i,j,k,t+1) = !act(i,j,k,t) \;\&\&\; hum(i,j,k,t) \;\&\&\; (act(i+1,j,k,t) \,\|\, act(i-1,j,k,t) \,\|\, act(i,j+1,k,t) \,\|\, act(i,j-1,k,t) \,\|\, act(i,j,k+1,t) \,\|\, act(i,j,k-1,t)) \,\|\, act(i+2,j,k,t) \,\|\, act(i-2,j,k,t) \,\|\, act(i,j+2,k,t) \,\|\, act(i,j-2,k,t) \,\|\, act(i,j,k-2,t)) \quad (1)$$

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

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

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。

$$ext(i,j,k,t+1) = !ext(i,j,k,t) \;\&\&\; cld(i,j,k,t) \;\&\&\; (ext(i+1,j,k,t) \,\|\, ext(i-1,j,k,t) \,\|\, ext(i,j+1,k,t) \,\|\, ext(i,j-1,k,t) \,\|\, ext(i,j,k+1,t) \,\|\, ext(i,j,k-1,t)) \,\|\, ext(i+2,j,k,t) \,\|\, ext(i-2,j,k,t) \,\|\, ext(i,j+2,k,t) \,\|\, ext(i,j-2,k,t) \,\|\, ext(i,j,k-2,t)) \quad (4)$$

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。

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

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 $0 < q_{i,j,k} \leq 1$ 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

3. ビルボード処理を用いた高速画像生成

ﾋﾞﾙﾎﾞｰﾄﾞ処理は、樹木の表示によく用いられ、樹木の変わりに、樹木のﾃｸｽﾁｬをﾏｯﾋﾟﾝｸﾞした多角形を表示することで計算時間を削減する手法である[3]。ここでは、雲の表示にこれを応用する。そのため、単一のﾒﾀﾎﾞｰﾙの密度を積分した値とその透明度をもつﾃｸｽﾁｬを計算する。これをﾒﾀﾎﾞｰﾙの直径を一辺とする正方形にﾏｯﾋﾟﾝｸﾞしたﾋﾞﾙﾎﾞｰﾄﾞを用意する。このﾋﾞﾙﾎﾞｰﾄﾞをOpenGLのαﾌﾞﾚﾝﾃﾞｨﾝｸﾞ機能により重ね書きすることで光の一次散乱を考慮した雲の色を計算する。これは、次の2段階の処理により計算する。すなわち、1)各ﾒﾀﾎﾞｰﾙ中心に届く太陽光の強さを計算し、2)視点から見た雲の色を計算する。

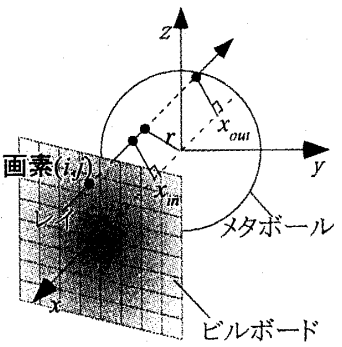

図1 ビルボード用テクスチャの生成

3.1 ビルボード用テクスチャの計算

図1に示すように、ﾒﾀﾎﾞｰﾙを原点に配置し、x軸に垂直でﾃｸｽﾁｬの各画素を通るﾚｲを考える。このﾒﾀﾎﾞｰﾙに単位強さの白色光が照射した場合、ﾚｲ方向へ散乱する光の強さはﾚｲに沿ってﾒﾀﾎﾞｰﾙの密度を積分した値に比例する。そこで、ﾃｸｽﾁｬのRGB成分はﾚｲに沿ってﾒﾀﾎﾞｰﾙの密度関数を積分した値とする。すなわち、

$$R(i,j)=G(i,j)=B(i,j)=q\int_{x_{in}}^{x_{out}} f(r,D)dx \qquad (8)$$

ここで、x_{in}およびx_{out}は、それぞれ、ﾚｲがﾒﾀﾎﾞｰﾙに交差する点のx座標であり、rはﾚｲ上の点と原点との距離である(図1参照)。また、単位強さの光がこのﾒﾀﾎﾞｰﾙをﾚｲに沿って通過するときに、減衰する割合をﾃｸｽﾁｬのA成分とする。これは、次式により計算できる。

$$A(i,j)=\exp(-\kappa R(i,j)) \qquad (9)$$

ここで、κは雲粒子の減衰係数である。式(8)、(9)から、ﾋﾞﾙﾎﾞｰﾄﾞ用のﾃｸｽﾁｬはﾒﾀﾎﾞｰﾙの中心密度ごとに用意する必要があるが、本稿では中心密度を32段階に量子化し、32個のﾃｸｽﾁｬを用意した。

3.2 メタボールに届く太陽光の計算

太陽光は各ﾒﾀﾎﾞｰﾙに届くまでに雲粒子により減衰する。その減衰率はそのﾒﾀﾎﾞｰﾙと太陽の間にあるﾒﾀﾎﾞｰﾙのA値を掛け合わせたものに等しい。よって、各ﾒﾀﾎﾞｰﾙ中心に届く太陽光の強さは次の処理により求まる。

1)視点を太陽位置に、視線を太陽方向と逆向きにし、平行投影に設定する。またｶﾗｰﾊﾞｯﾌｧ(以下、CB)のR値を1.0に初期化しておく。
2)太陽に近い順にﾒﾀﾎﾞｰﾙを並び替える。
3)各ﾒﾀﾎﾞｰﾙについて以下を繰り返す。
 a)ﾋﾞﾙﾎﾞｰﾄﾞが太陽方向と垂直になるよう回転する。
 b)ﾋﾞﾙﾎﾞｰﾄﾞを描画する。ただし、CB内のR値にﾋﾞﾙﾎﾞｰﾄﾞﾃｸｽﾁｬのA値を乗じながら描画する。
 c) ﾒﾀﾎﾞｰﾙ中心位置に対応するCB内の画素のR値を取り出し、太陽光のRGB値を乗じる。この値がﾒﾀﾎﾞｰﾙに届く光のRGB成分E_R,E_G,E_Bとなる。

3.3 視点から見た雲の色の計算

視点から見た雲の色は各ﾒﾀﾎﾞｰﾙに到達した光E_R, E_G, E_Bのうち、視点方向に散乱する光のRGB成分I_R, I_G, I_Bを算出し、これを全て足し合わせることで求められる。このとき、I_R,I_G,I_Bは視点に到達するまでに雲粒子により減衰する。この減衰率は各ﾒﾀﾎﾞｰﾙと視点との間に存在するﾒﾀﾎﾞｰﾙのA値を掛け合わせたものとなる。よって、以下の手順により画像生成を行える。

1)背景を描画する。
2)視点から遠い順にﾒﾀﾎﾞｰﾙを並び替える。
3)各ﾒﾀﾎﾞｰﾙについて以下の処理を繰り返す。
 a) ﾒﾀﾎﾞｰﾙ中心と視点を結ぶﾍﾞｸﾄﾙ$\mathbf{v}$を求める。
 b)ﾋﾞﾙﾎﾞｰﾄﾞがﾍﾞｸﾄﾙ$\mathbf{v}$と垂直になるよう回転する。
 c)雲粒子の位相関数からE_R,E_G,E_Bのうち、視点方向に散乱する成分I_R,I_G,I_Bを求める。
 d) ﾋﾞﾙﾎﾞｰﾄﾞﾃｸｽﾁｬのRGB値それぞれにI_R,I_G,I_Bを乗じたﾃｸｽﾁｬを用いてﾋﾞﾙﾎﾞｰﾄﾞを描画する。このとき、CB内のRGB値にﾋﾞﾙﾎﾞｰﾄﾞﾃｸｽﾁｬのA値を乗じながら描画する。

4. 適用例とまとめ

提案手法を用いて15秒間の雲のｱﾆﾒｰｼｮﾝを作成した。格子点数は80×80×10である。図2にその一部を示す。時刻は(a)から(d)へと変化する。画像サイズは400x300である。ﾘｱﾙな雲の動きを表現できていることがわかる。1ﾌﾚｰﾑの平均画像生成時間は、富士通FMV-6266 NS3/X (Pentium II 266MHz)を用いて3秒である。

ｾﾙｵｰﾄﾏﾄﾝを用いて雲の動きをｼﾐｭﾚｰｼｮﾝし、OpenGLを用いてその結果を高速表示する手法を提案した。

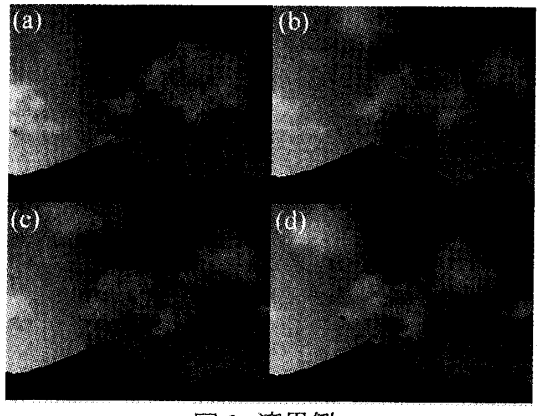

図2 適用例

参考文献

[1] JT Kajiya et. al: *Computer Graphics*, vol. 18, pp.165-174 (1984).
[2]菊地ら:画像電子学会誌,第27巻4号, 頁317-326(1998).
[3] T.McReynolds et. al:Siggraph'97 Course Note 22, pp. 28-53 (1997).
[4] N. Nagel, Physica A. 182, pp. 519-531 (1991).
[5] G. Wyvill et al.: *Proc. CG International*, pp.439-475 (1990).