

Video Texture を用いた動的な雲の高速レンダリング Fast Rendering of Dynamic Clouds by Using Video Texture

江口 千央† 土橋 宜典† 山本 強†
Yukihisa Eguchi Yoshinori Dobashi Tuyoshi Yamamoto

1. あらまし

雲、煙、炎、水などの自然現象のシミュレーションはCGにおいて重要な研究分野の一つであり、特に雲の表示は景観画像の作成に重要な役割を果たす。屋外景観のアニメーション映像を作成する場合、時々刻々と変化する雲の映像を作成する必要がある。しかし、動的な雲に対して毎フレーム輝度計算を行って画像生成を行うことは計算コストが高い点が問題となる。

そこで、本研究では、Video Texture と呼ばれる手法を利用し、雲のアニメーション映像を高速に生成する手法を提案する。Video Texture を用いれば、与えられた短時間での動的な密度分布から任意の長さの雲の動的な密度分布を生成できる。そこで、与えられた密度分布に対し、前もって輝度計算を行っておくことによって高速に雲のアニメーション映像を生成する手法を提案する。

2. Video Texture の雲への適用

本研究では[1]の考え方をを用いて与えられた短時間での動的な密度分布から任意の長さの雲の動的な密度分布を生成する。Video Texture は入力映像を用いて任意の長さの映像を生成する技術である。与えられた短時間の入力映像を構成する全てのフレームから任意の二枚のフレームを選択し、フレーム間の各画素の色の差分を計算してフレーム間の類似度を定義する。本論文では、入力データとして流体シミュレーションによって得られた動的な雲のボリュームデータを用いている。そのため、ボリュームデータ間の差分は各ボクセルの密度の差になる。二つのボリュームデータ V_i および V_j 間の類似度 D_{ij} は式(1)のように定義できる。

$$D_{ij} = \|V_i - V_j\|^2 \quad (1)$$

ただし、 $\|V_i - V_j\|^2$ はボリュームデータ V_i および V_j の各ボクセルの密度差の2乗和を表す。いま、 n 個のボリュームデータが与えられたとする。まず、任意の二つのボリュームデータの組み合わせについて式(1)により類似度を計算する。そして、類似度の高いボリュームデータを確率的に選択しながら時間的に連続して表示することで任意の長さの映像を作り出すことができる。例えば、ボリュームデータ V_i を表示した場合、その次に表示するボリュームデータは以下のように選択する。ボリュームデータ V_i と他の全てのボリュームデータとの類似度 D_{ij} に比例した確率分布を考える。この確率分布に従う乱数をもとに次に表示するボリュームデータを選択する。これにより、繰り返し映像に陥ることなく任意の長さの映像を作り出すことがで

きる。なお、本稿では、上記の乱数によって次のボリュームデータを選択する処理をリープとよぶこととする。

本研究では上記の考え方をを用いて、以下のように滑らかに変化する雲の動的な密度分布を生成する。まず、上記のリープ処理は毎フレーム行うのではなく、指定された一定のフレーム間隔 m おきに行って、連続するボリュームデータを生成する。このデータに対し、さらにクロスディゾルブの考え方を適用することで滑らかな雲の時間変化を実現する。以下、詳しく説明する。

いま、 l 番目のフレームで乱数による選択処理を行い、ボリュームデータが V_i から V_j に変化した場合を考える。

ボリュームデータ V_j は V_i と類似した密度分布を持つが滑らかな変化とはならない場合も多い。そこで V_{i-m+1} ,

V_{i-m+2}, \dots, V_i なる連続するボリュームデータと V_{j-m+1} ,

V_{j-m+2}, \dots, V_j なる連続するボリュームデータをクロス

ディゾルブすることで解決する。すなわち、フレーム

$l-k$ ($k=1, \dots, m$) でのボリュームデータ V_l は次式により表現される。

$$V_{l-k} = w_k \times V_{i-k} + (1.0 - w_k) \times V_{j-k} \quad (2)$$

ただし、 $w_k = k/m$ である。

3. 前計算による雲の高速表示

太陽の光が雲を通過するとき、光は雲を通過する距離と雲の密度に応じて減衰する。ボリュームデータの各ボクセルに対して輝度計算をするには、太陽の光が各ボクセルに到達するまでの減衰率を計算しなくてはならない。ボクセル p に対する減衰率は次式で表せる。

$$L(p) = \exp(-\tau(p)) \quad (3)$$

式(2)において、光学的距離の計算には、ボクセル p と光源の間に存在する雲の密度を積分しなくてはならず、計算コストが高い。そこで、提案法では、 $L(p)$ を全てのボクセルについて前もって計算しておくことにより、高速に雲の表示を行う。全てのボリュームデータの全てのボクセルについて、式(3)により減衰率 $L_i(p)$ ($i=1, 2, \dots, n$) を計算しておけば、動的に変化する雲のアニメーションを高速に表示できる。しかし、前節で述べたクロスディゾルブを適用して生成したボリュームデータの場合には、上記の考え方をそのまま適用することはできない。その理由は、式(3)で表される減衰率は、密度に対して非線形であるためである。すなわち、クロスディゾルブによって生成されたボリュームデータについては、減衰率を新たに計算しなおさなくてはならず、レンダリング速度が低下してしまう。提案法では、この問題を以下のようにして解決する。

前節と同様に、フレーム l でリープする場合を考える。

† 北海道大学情報科学研究科

このとき、クロスディゾルブによりフレーム $l-m$ から l までのボリュームデータの密度は、式(2)によって与えられている。まず、光学的距離は密度に対して線形であるため次式が成り立つ。

$$\tau(p) = -w_k \tau_{i-k}(p) - (1.0 - w_k) \tau_{j-k}(p) \quad (4)$$

この式を式(3)に代入すると減衰率は次式によって計算できる。

$$\begin{aligned} L_p &= \exp(-w_k \tau_{i-k}(p) - (1.0 - w_k) \tau_{j-k}(p)) \\ &= \{\exp(-\tau_{i-k}(p))\}^{w_k} \times \{\exp(-\tau_{j-k}(p))\}^{(1.0-w_k)} \\ &= (L_{i-k}(p))^{w_k} \times (L_{j-k}(p))^{(1.0-w_k)} \end{aligned} \quad (5)$$

したがって、 $L(p)$ は前もって計算しておいた減衰率

$L_{i-k}(p)$ および $L_{j-k}(p)$ を参照するのみで計算できる。

しかし、我々の実験では、べき乗の計算のコストが高く、このままでは高速に $L(p)$ を算出することは難しい。そこで、提案法では、上式を次式のように近似した式を用いる。

$$L(p) = ((1.0 - w_k) L_{i-k}(p)) \times (w_k L_{j-k}(p)) \quad (6)$$

これにより低コストに $L(p)$ の減衰率を計算できる。

4. 実験

前計算による出力の比較実験と高速化の検証を行った。実験結果を図1と図2に示す。なお、実験を行う計算機環境はCPUがPentium(R) 43.80GHz(RAM 2.00GB)のPCである。図1は前計算を行った場合の出力結果、図2は前計算を行わない場合の出力結果である。二つの出力結果はほぼ同じである。図1の前計算を行ったとき、1フレーム表示するのにかかる時間は平均0.289secであり、図2の前計算を行わないとき、1フレーム表示するのにかかる時間は平均0.445secであった。前計算により高速に表示ができていくことがわかる。

5. 結び

与えられた密度分布に対し、前もって輝度計算を行っておくことによって高速に雲のアニメーション映像を生成する手法を提案した。なお、今回は太陽の直射光のみを考慮して前処理を行っているが、天空光などを考慮する事により更にレンダリング速度の向上を実現できると考えられる。

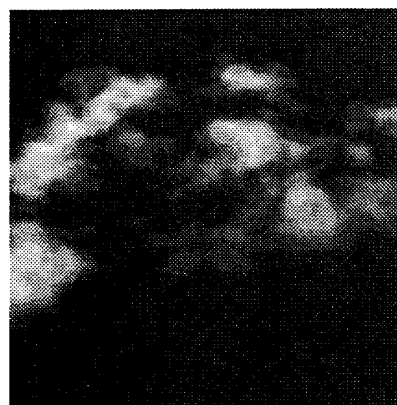


図1 前計算による輝度値の出力結果



図2 前計算をしない場合の輝度値の出力結果

文献

[1]A. Schödl, R. Szeliski and D. Salesin. "Video Texture," Proc. SIGGRAPH'00, pp.489-498 (2000)

以上