

論文内容の要旨

Modeling Granular Material and its Application to CG Animation (粒状物質のモデリングとそのCGアニメーションへの応用)

尾上 耕一

(複雑理工学専攻 複雑計算科学研究室)

1 概要

コンピュータグラフィックスの分野において、水、雲、煙、炎など自然現象のモデリングおよびレンダリングは重要な課題の一つである。しかし、様々な自然現象の中でも、砂などの粒状物質を扱った研究はほとんど行われていない。粒状物質のアニメーションはバーチャリアリティー、映画、ゲーム等の様々な分野への応用が期待でき有用な研究である。一方、物理シミュレーションの分野では粒状物質のシミュレーション手法が提案されているが、特にCGで有用となるような粒状物質で構成された地面に適用する場合には非常に計算コストが高いことが知られている。CG、特にインタラクティブなCGアプリケーションへの応用のためには、粒状物質の挙動を完全に再現する手法の開発よりも、単純なモデルを用いて崩落などの粒状物質に特徴的な挙動を表現する方法が有効である。そこで、本稿では単純なモデルを用いて粒状物質を対象とするアニメーションを効率的に生成する手法を提案する。

2 はじめに

本論文では、粒状物質を対象としたアニメーション生成のためのモデリングおよびレンダリング手法を提案する。粒状物質は主に風による作用と物体による作用を受ける。本論文ではそれぞれの場合についてのモデリング法について述べる。まず、風が粒状物質に及ぼす作用の結果として形成される風紋と砂丘について、モデリングおよびレンダリング法を3節で説明する [1, 3]。次に物体が粒状物質に及ぼす作用に関して、まず柔らかい地面上に残る物体の痕跡の形成について4節で述べる [2]。そして、物体による粒状物質のインタラクティブな変形法と粒状物質の崩落のレンダリングについて5節で述べる [4]。また、物体が粒状物質で構成された地面に高速に衝突した際の粒状物質の飛び散りを表現する方法を6節で述べる [5]。

3 風紋・砂丘を含む砂漠景観の表示法

本節ではCGを用いて砂漠の景観をリアルに表示するための方法を提案する。まず、砂漠地形のモデリングについて議論する。砂漠地形は主に砂丘とその表面にできる細かい縞模様である風紋からなっている。それぞれを形成するためのモデルについて述べる。このモデルでは、砂丘と風紋を単位面積当たりの砂の量、すなわち高さをもつセルの二次元配列(ハイトフィールド)で表し、風による砂粒子の動きをモデル化した式を用いて計算することにより、砂丘と風紋を形成する。次に砂丘と風紋を組み合わせることで砂漠の景観をレンダリングする方法について提案する。レンダリングでは砂丘の表面上に小さな凹凸である風紋を表現するためにバンプマッピングの手法を用いている。この計算の効率化のためにLOD(Level Of Detail)の手法を用いる。また砂漠の景観をよりリアルに表示するために、砂丘の影、風紋の影を考慮している。

適用例として、砂丘に風紋をバンプマッピングした例、砂丘の影を表示した例、風紋の影を表示した例を図1に示す。

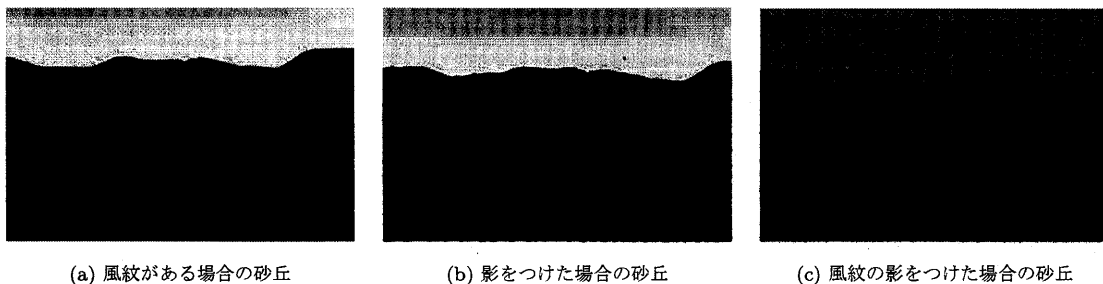


図 1: 砂漠景観の表示例

4 柔らかい地面上の物体による痕跡の高速表示法

CGによるアニメーションに関して、近年、特にキャラクターの動作に伴う二次的な現象も研究対象となっている。例えば、キャラクターが柔らかい地面（砂、泥、雪など変形しやすいもの）に残す足跡や自動車や自転車のタイヤ跡、ボールが落とされたり地面の上を引きずられたりした跡も、二次的な現象に含まれる。本節では物体の衝突により地面に残る痕跡を計算する方法を提案する。

提案法では、(1) 地面と物体との衝突判定、(2) 衝突部分の地面の物質の圧縮と周囲への移動、(3) 急斜面での崩落、をそれぞれ計算することによって物体の接触による地面の変形を計算する。地面と物体との衝突判定はハイトフィールドが一価関数であるという性質を利用してZバッファを用いて行う。Zバッファ法を用いることによりグラフィックスハードウェアを利用して高速に衝突判定を行うことができる。地面の上を物体が引きずられている場合には、物体がアニメーションのフレーム間で動いた軌跡を考慮しないと地面の正しい変形結果が得られない場合がある。そこで、物体が引きずられている場合には前フレームでの物体と現在のフレームの物体の間を掃引してできた物体との衝突判定を行う。また、地面の変形を効率的に行うために、多重解像度のハイトフィールドによる最適化法も提案する。基準となる解像度のハイトフィールドより低解像度のハイトフィールドをいくつか準備しておき、物体が視点から遠い位置に存在する場合等には低い解像度のハイトフィールドを用いて地面の変形の計算を行うことによって地面の変形を効率的に行うことができる。

適用例として砂、泥、雪でできた地面に靴で足跡をつけた場合の計算結果を図2に示す。これらの結果が示すように、パラメータを変えることによって様々な種類の地面を表現することができる。

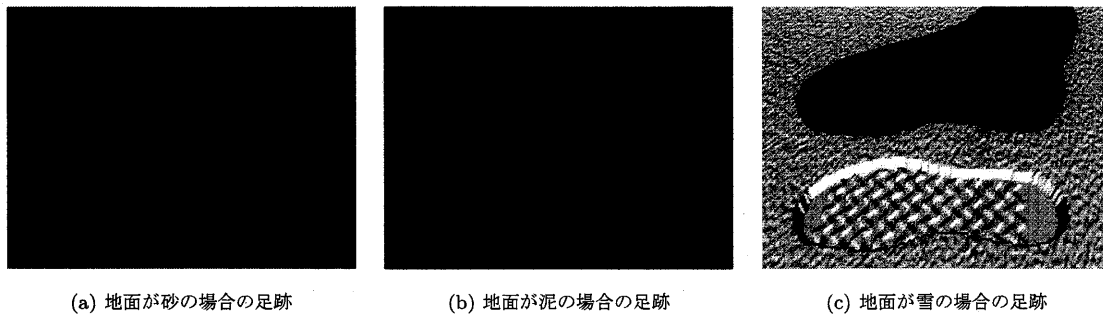


図 2: 様々な種類の地面に残る足跡の例

5 物体による柔らかい地面のインタラクティブな変形法

近年、バーチャルリアリティシステムのようなインタラクティブアプリケーションが広く普及してきている。本発表では、このようなアプリケーションに利用可能な、粒状物質で構成された地面に物体が接触したときの地面の変形アルゴリズムを提案する。提案法では物体と物体上の粒状物質を、物体存在領域を表す高さのスパン（ハイトスパンと呼ぶ。図3の矢印(↓)）のリストを各要素として持つ二次元配列（HS(Height Span) マップ）で表す。各ハイトスパンは次の三つの要素で構成される。(1) 物体の上向き面の高さ、(2) 下向き面の高さ、(3) その位置での物体上の粒状物質の高さ（図3ではスパン上の太線）。これにより地面と同様に高さの変化だけを考えればよくなるので粒状物質の動きを効率的に計算できる。HS マップの例を図4に示す。提案法はハイトスパンマップ（HS マップ）を用いることによって、凹物体を含む様々な形状の物体による地面の変形と、物体上に存在する粒状物質を表現できる。提案する変形アルゴリズムは4節と同様に次の3ステップからなる：(1) 物体と地面の衝突判定、(2) 粒状物質の移動、(3) 急斜面での粒状物質の崩落。衝突判定はここではHS マップを用いて行う。

レンダリングはハイトフィールドから生成したポリゴンメッシュを描画することにより行う。そして、崩落が起こった部分で崩落の方向に応じた粒状物質の動きを表現する。そのために本節ではテクスチャスライド法（図5）を提案する。まずポリゴンメッシュの各頂点 $\mathbf{v}(x, y, z)$ にテクスチャ座標 $\mathbf{T}(u, v) = (\omega_x x, \omega_y y)$ を与える。ここで、 ω_x と ω_y は各ポリゴンにマッピングされるテクスチャ画像のサイズを決定するパラメータである。本節の実験では、 $\omega_x = \omega_y = 0.125 (= 32/256)$ とした。そして、各ポリゴンにテクスチャ座標のオフセット \mathbf{T}_{offset} を与える（図5参照）。各タイムステップで、崩落が起こったセルについて \mathbf{T}_{offset} に $\Delta\mathbf{T}_{offset}$ を加える。 $\Delta\mathbf{T}_{offset}$ は次の式で計算する。

$$\Delta\mathbf{T}_{offset} = -\gamma Q_{collapse} \mathbf{E}, \quad (1)$$

ここで、 $\mathbf{E}(e_x, e_y)$ は水平面上での崩落方向であり、あるセルからその隣接するセルに向かう8方向の中の一つである。 $Q_{collapse}$ は崩落

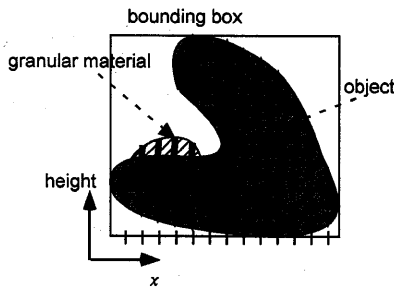


図 3: HS マップ (断面図)

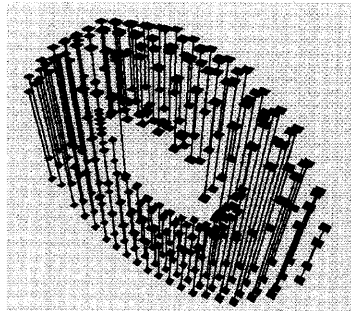


図 4: ドーナツ形物体の HS マップ

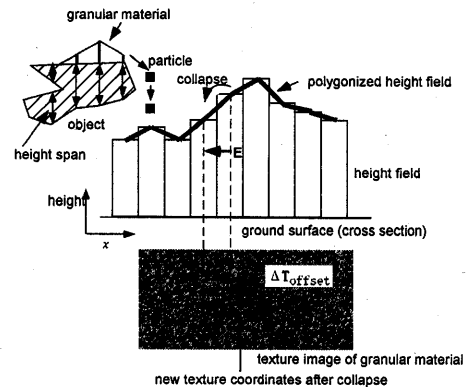


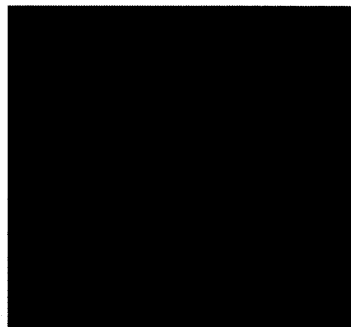
図 5: テクスチャスライド法

量であり、 γ は粒状物質の外見上の移動する速さを決めるパラメータである。本節では $\gamma = 0.05$ とした。あるセルから複数のセルに崩落が起こる場合は、式 (2) を各崩落についての崩落方向 \mathbf{E} について計算し、対応する各ポリゴンについて $\Delta \mathbf{T}_{offset}$ を加算する。ポリゴンをレンダリングするには、各ポリゴンについて各頂点のテクスチャ座標に \mathbf{T}_{offset} を加算する。

砂の上でバケツを操作した結果を図 6 に示す。バケツは引きずられ (図 6(a))、砂をすくい (図 6(b))、砂を落としている (図 6(c))。これらの結果はインタラクティブに作成した。このときの平均フレームレートは約 10fps であった。ここでは 256×256 の解像度のバイトフィールドを用いた。そして、HS マップの解像度は 30×30 であった。計算時間の計測には、Dell Precision530(CPU: Pentium4 2.26GHz, GPU: GeForce4 MX420) を用いた。



(a) バケツを砂の上で引きずった例



(b) 砂に埋もれたバケツの例



(c) バケツから砂を落としている様子

図 6: バケツによる砂の変形

6 粒子要素法を用いた粒状物質の飛散アニメーション

近年、水などの自然物と物体との相互作用を考慮したアニメーションに関する研究が行われている。その中でも物体の衝突による地面の変形についてはすでに多くの研究が行われているが、衝突時の粒状物質の飛散までは表現できていない。そこで、本論文では物体が粒状物質で構成された地面に衝突したときの粒状物質の飛散アニメーションの計算方法を提案する。

本節では粒状物質の飛散の計算に粒子要素法 (DEM) を拡張したものをを用いている。DEM では粒子の集合の挙動について、各粒子の移動と回転をその粒子に接触している粒子から受ける反発力を用いて計算する。しかし、粒子要素法は地面を構成する全粒子に関する挙動を逐一計算するので、本節で対象とする砂地のような広い空間をシミュレートすると計算時間が膨大になる。この問題を解決するために、本節では粒子要素法を拡張し、ここではいくつかの粒子を一つの大きい粒子 (DE と呼ぶ) で代表して粒子の集合の動きの計算を行うこととする。まず、物体衝突地点の周辺に DE を配置し (図 7) する。そして、DE 間の接触力については DEM を用いて計算する。DE と物体との衝突判定には前節で提案した HS マップを用いる (図 8)。しかし、空中で粒子がまとまったまま移動するのは不自然である。したがって、物理現象との違いが顕著に現れるのは粒子が飛散した場合であると考えられる。そこで、空中に飛散した

DEについては粒子の大きさにまで分解する。DE数の増大を防ぐため、飛散したDEが着地した時には、地面に存在しているDEに飛散していたDEを融合する。

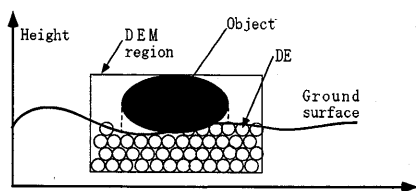


図 7: DE の配置

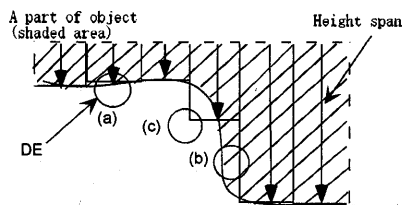


図 8: DE と物体の衝突判定

適用例として、図 9 に、Bunny モデルを地面に落とした例を示す。物体が地面から力を受けた結果、着地後に Bunny が倒れる。計算時間は 1 タイムステップの計算に約 2.0 秒かかった。計算時間の計測には、Dell Dimension 4500C (CPU: Pentium4 2.26GHz, GPU: GeForceFX 5200) を用いた。



図 9: Bunny を自由落下させた例

7 まとめ

粒状物質を対象としたアニメーションを効率的に生成する手法を提案した。まず、風紋・砂丘を含む砂漠景観のモデリングおよびレンダリング法を提案した。次に砂・泥・雪等、様々な種類の柔らかい地面における物体の痕跡を、Zバッファ法を用いた物体と地面の衝突判定を行い効率的に計算する方法を示した。また、物体による粒状物質のインタラクティブな変形法を提案した。HS マップを用いることにより物体による粒状物質の運搬や地面に埋もれる物体を表現することが可能となった。そして、テクスチャスライド法により粒状物質の崩落を高速にレンダリング可能となった。最後に粒子要素法を拡張した方法により物体衝突時の粒状物質の飛散アニメーションを効率的に計算する方法を提案した。

参考文献

- [1] K. Onoue and T. Nishita. A method for modeling and rendering dunes with wind-ripples. *Pacific Graphics*, pp. 427–428, October 2000.
- [2] K. Onoue and T. Nishita. An efficient method for displaying marks on soft grounds created by objects. *Journal of IIEEJ*, Vol. 32, No. 4, pp. 328–335, 2003.
- [3] K. Onoue and T. Nishita. A rendering method on desert scenes of dunes with wind-ripples. *Journal of IEICE*, Vol. J86–DII, No. 2, pp. 282–289, 2003.
- [4] K. Onoue and T. Nishita. Virtual sandbox. *Pacific Graphics*, pp. 252–259, October 2003.
- [5] K. Onoue and T. Nishita. Animating spattering granular material using distinct element method. *Journal of IIEEJ*, Vol. 33, No. 4-B, pp. 523–530, 2004.