

粒子要素法を用いた粒状物質の飛散アニメーション

東京大学大学院新領域創成科学研究科

尾上 耕一

東京大学大学院新領域創成科学研究科

正会員 西田 友是

〈あらまし〉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、本稿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。提案法では、各粒子の挙動を粒子間の衝突と反発によって表す方法である粒子要素法を用いて地面の変形を計算する。これにより、物体による地面の変形と粒状物質の飛散を表現することができる。さらに、粒状物質が物体に及ぼす力も考慮して物体の動きを計算できる。提案法の有効性を示すため、適用例として剛体を砂地に落とした時の砂の飛散を計算した結果を示す。

キーワード：粒状物質，飛散，アニメーション，自然現象，粒子要素法

〈Summary〉 In recent years there have been many studies on animations that take account of interactions between objects and natural objects such as water. Among others, some researchers propose deformation methods for ground surfaces when they are in contact with objects. But these methods cannot represent granular material spattering at an impact of an object. In this paper, we propose an algorithm which can represent the animation of granular material spattered by objects. In our proposed method, the ground surface is deformed by using the distinct element method which calculates the motion of each grain by collisions and reactions among grains. The deformation of the ground surface and spattering of granular material are represented by this method. Moreover our method can calculate the motion of objects taking account of forces exerted on the objects by granular material. In order to demonstrate the validity of our method, we show a result of calculating spattering sand when a rigid object falls onto a sand surface.

Key words: Granular material, Spattering, Animation, Natural phenomena, Distinct element method

1. はじめに

コンピュータグラフィックスを用いたアニメーションに関する研究は従来から行われてきたが、最近では水や地面などの自然物と物体との相互作用によるアニメーションに関する研究が行われている。水に物体を落としたときの物体と水の動きに関する研究としては、例えば、O'Brienら¹²⁾、Kunimatsuら⁷⁾、Takahashiら¹⁸⁾によるものがある。物体による地面の変形に関する研究も盛んに行われている。本稿のテーマである地面の変形に関する研

究は、アニメーション以外にもドライブシミュレーターやコンバットシミュレーター等に広く応用されている。このようなアプリケーションで必要になる物体と地面(粒状物質)の相互作用による効果としては以下のものが挙げられる。

1. 地面への物体の接触跡の生成(足跡、タイヤ跡等)
2. 物体による粒状物質の運搬(バケツやブルドーザーで砂をすくう等)
3. 物体への粒状物質の付着と落下(足の裏に砂が付着し、足を持ち上げた後で落下する等)
4. 物体の移動による粒状物質の飛散(物体が落下したとき、物体を地面上で高速に引きずったとき等)
5. 粒状物質から力を受けることによる物体の動き(物体を地面に自由落下させたり放り投げたりする場

“Animating Spattering Granular Material Using Distinct Element Method” by Koichi ONOUE (Graduate School of Frontier Sciences, The University of Tokyo), Tomoyuki NISHITA (Member).

合)

6. 濡れた粒状物質の表現 (砂の城のモデリング等)
7. 埃状の粒状物質の動き (非常に細かい粒状物質が空気の動きによって舞い上がり煙状になる場合)
8. 風による粒状物質の動き

これらのエフェクトの一部を扱った研究はすでに存在している。Liらの方法⁸⁾は2を扱える。Chanclouら²⁾の方法とOnoueら¹⁵⁾の方法は1を目的としたものである。Sumnerら¹⁷⁾の方法は、1と3を目的としたものである。Onoueら¹⁶⁾の方法は、1と2を目的としたものである。4を実現可能な粉粒体シミュレーションの手法もある⁹⁾が、計算コストが高い。6に関連した方法は特に粘土のモデリング等に見られる^{9,13)}。Chenら³⁾の方法は7を目的としたものである。8については、物体の存在しない場合については、風による風紋の形成の研究をOnoueらが行った¹⁴⁾。従来法の中で本研究と関連の深いものについては次節で詳しく紹介する。

本稿では、これらのエフェクトの中で、特に4と5を表現することを目的とする。主な従来法では地面を連続体として扱っていたが、粒状物質が飛び散る場合には連続体として扱うのは困難である。そのために、本稿では粒子間の衝突と反発によって表す方法である粒子要素法 (DEM: Distinct Element Method)^{4),19)}を拡張した方法を用いて各粒子の動きを計算する。これにより、物体による地面の変形と粒状物質の飛散を表現することができ、さらに、粒状物質が物体に及ぼす力も考慮して物体の動きを計算できる。ただし、粒状物質が水を含む場合は、水を含まない場合と飛散の様子が異なってくるので、本稿では粒状物質が水を含まない場合を対象とする。

2. 関連研究

物体の衝突による地面の変形を扱った研究はすでにいくつか存在する。

まずLiらはハイトフィールドを用いた、物理法則に基づく土の崩落モデルを提案した⁸⁾。彼らは土の崩落とブルドーザーによる土の掘削、積載、運搬をリアルタイムに計算した。Chanclouらは地面を弾性のあるシートとして表現し、地面に残る物体の痕跡を生成する方法を提案した²⁾。Sumnerらは砂・泥・雪など、様々な種類の地面の上に行ける物体の接触の痕跡を計算した¹⁷⁾。また、物体が地面から離れた後に物体の底面に付着した粒状物質が落下する様子も表現した。Onoueらはグラフィックスハードウェアを用いて物体と地面の衝突判定を行い、リアルタイムに物体の痕跡を生成する方法を提案した¹⁵⁾。また、層状のデータ構造を用いて物体上の粒状物質を表現した¹⁶⁾。Chenらは前処理で流体計算を行うことにより

自動車を通った後に舞い上がる土埃を表現した³⁾。

ただし、これらの従来法では基本的に地面をハイトフィールドで表しているだけなので、物体が衝突したときの粒状物質の飛び散りは表現できない。本稿ではこの粒状物質の飛び散りを表現できる手法を提案する。

参考のためにその他の地面のモデリングに関する研究についても挙げておく。さまざまな種類の地面の中でも、特に雪を扱った方法がいくつか発表されている。西田らは雪をメタボールを使ってモデル化し雪が様々な物体に積もる様子を表現した¹¹⁾。Fearingは物体と地面への降雪と積雪のモデル、および雪の安定性のモデルを提案した⁵⁾。地面の大きなスケールのモデリングに関しては、Musgraveらが地形の生成法を提案した¹⁰⁾。Benesらは地形侵食のビジュアルシミュレーションを行うための地形の階層表現を提案した¹⁾。また、Onoueらは風紋と砂丘を含む砂漠地形の生成とレンダリングを行った¹⁴⁾。

3. データ構造

本節では提案法で用いているデータ構造に関して、地面、物体と物体上の粒状物質、粒子要素法に用いるデータそれぞれについて説明する。

3.1 地面

提案法では地面をハイトフィールド、すなわち高さをもつセルの二次元配列で表す。ここで、高さはある水平な基準面 (高さ0) への距離とする。各セルの高さは単位面積当たりの地面の物質の量に等しい。ハイトフィールドの初期状態は様々な方法で決定することができる。本稿では初期状態の高さを乱数によって決定したが、実際の地形データや他のモデラーの出力を利用することも可能である。

3.2 物体と物体上の粒状物質

提案法ではポリゴン物体を扱う。そして、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 (図1の矢印 (↓)) のリストを各要素として持つ二次元配列 (HS マップ) で表す。ここでは、二次元配列の各要素のことをハイトフィールドの場合と同様にセルと呼ぶことにする。HS マップはOnoueら¹⁶⁾が用いたデータ構造であるが、Onoueらは各スパン (ハイトスパンと呼ぶ) を次の三つの要素で構成していた。(1) 物体の上向きの面の高さ、(2) 下向きの面の高さ、(3) その位置での物体上の粒状物質の高さ (図1ではスパン上の太線)。本稿では各スパンに次の二つの要素を加える。(4) 物体の上向きの面の法線ベクトル、(5) 下向きの面の法線ベクトル。これは、後述する粒子要素法の計算に必要な。なお、HS マップは物体が回転した時には再計算する必要がある。HS マップを用いる利点は、物体上の粒状物質のレンダ

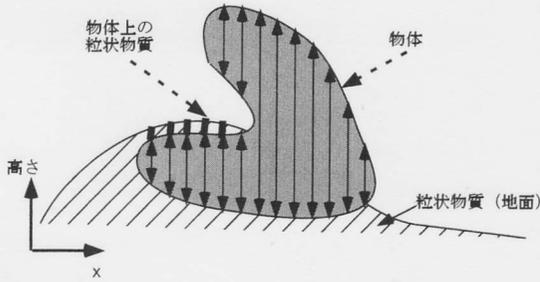


図1 HSマップ(断面図)
Fig. 1 HS map (in cross section)

リング、および物体と粒子の衝突判定の両方を一つのデータ構造で行えることである。

3.3 粒子要素法

本稿では地面の変形計算に粒子要素法^{4),19)}を拡張したものを用いている。粒子要素法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、本稿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、本稿では粒子要素法を拡張した、以下の方法を提案する。まず、ここではいくつかの粒子を一つの大きい粒子で代表してシミュレーションを行うこととする。ここで、粒子と代表した粒子を区別するために、いくつかの粒子を代表した大きい粒子をDE(Distinct Element)と呼ぶことにする。また単純化のため、DEは球形であるとみなす。DEを用いると物理現象を厳密には再現できなくなってしまうが、本稿では地面の内部(地表面より下の部分)での粒子の動きをDEの動きで表せるものとみなす。地面内部での粒子間の摩擦相互作用を考慮するために、DEの角速度も計算することにする。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。

なお、各DEはパラメータとして位置、速度、角速度、半径、質量、飛散中を示すフラグをもつ。

物体上とその周囲の粒状物質の動きが問題となるので、物体上と衝突後の物体周辺のみ粒子要素法の計算を行う。この計算を行う領域のことを本稿ではDEM領域と呼ぶ。単純化のために、本稿ではDEM領域を直方体の形状とする。

4. 地面変形アルゴリズム

本節では提案する地面変形アルゴリズムを説明する。提案法では物体の動きをシミュレーションかキーフレームアニメーションのどちらかで決める。キーフレームアニメーションで決める場合は物体がDEから力を受けないものとする。

地面変形の計算は以下のような順で行う。

衝突判定: 物体と地面(ハイトフィールド)の衝突判定を行う。

DEの配置: 衝突範囲周辺で粒子要素法を行うためのDEを配置する。

粒子要素法の計算: DE間、およびDEと物体との衝突判定を行い、衝突要素間での力を計算することによって、粒状物質の移動を計算する。

データの再構築: DEの位置から、ハイトフィールドとHSマップの粒状物質の高さを更新する。次節以降で各ステップについて詳しく説明する。

4.1 物体と地面の衝突判定

HSマップから物体底面の高さが求められるので、それとハイトフィールドの高さを比べることによって衝突判定を行う¹⁰⁾。衝突した場合は次節の処理を行う。衝突しなかったら、次節の処理はスキップする。ただし、すでにDEが配置されている場合は、本節の処理は必要ない。

4.2 DEの配置

物体が地面に衝突したときにまだDEが配置されていない場合、物体の周辺にDEを配置する。このために、まずDEM領域を物体のバウンディングボックスに等しくなるように設定する。そして、DEM領域と地面の両方に含まれる領域にDEを面心立方格子状に配置する。

ここで面心立方格子状にDEを配置するのは、できるだけ地面中のDEの密度を高くして実際の地面に近づけるためである。また、初期状態ではDEの半径と質量はすべて等しく設定する。

DEM領域の境界に位置するDEに閾値以上の力が加わったら、動的にDEを追加配置してDEM領域を拡張する。追加配置されたDEはDEM領域の境界部で局所的に面心立方格子を形成するように配置する。図2に拡張後のDEM領域を示した。

4.3 粒子要素法の計算

粒子要素法の計算の流れは次の通りである。まずDE間の衝突判定を行い、次に衝突したDE間に働く力を計算し、最後に各DEの状態(速度、位置、角速度)を更新する。

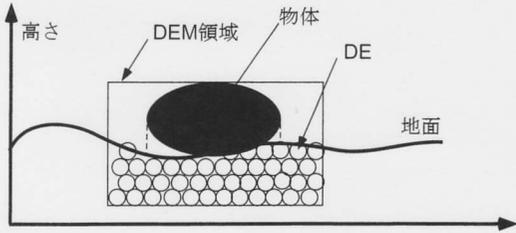


図2 DEの配置
Fig. 2 Configuration of DEs

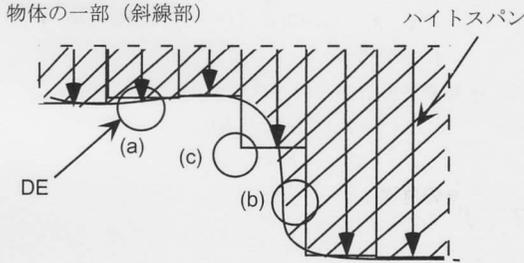


図3 DEと物体の衝突判定
Fig. 3 Collision detection between DE and an object

4.3.1 DEの衝突判定

DEを球とみなしているので2つのDE (G_1 と G_2)の衝突判定は、まず、交差点距離 d_n を、

$$d_n = r_1 + r_2 - \|c_1 - c_2\| \quad (1)$$

で計算し、 $d_n > 0$ の場合、衝突しているとして判定できる (図4参照)。ここで、 c_1, c_2 は各DEの中心の位置ベクトルであり、 r_1, r_2 は各DEの半径である。このとき、次節で説明する接触力の計算に必要な衝突ベクトルは次の式で求まる。

$$u_n = \frac{c_2 - c_1}{\|c_2 - c_1\|} \quad (2)$$

衝突判定を効率的に行うために、本稿では空間をボクセル分割する。各ボクセルにそこに含まれるDEを記憶しておき、あるDEについて衝突判定を行う時は、そのDEが含まれるボクセルとその隣接ボクセルに含まれるDEとの間でのみ衝突判定を行う。

DEと物体との衝突判定はHSマップを用いて以下のように行う (図3参照)。

1. DEがHSマップのどのセルに含まれるかを求める。
2. DEがハイトスパンの上面、あるいは下面と交差しているかどうかを調べる (図3(a))。この場合、 d_n はハイトスパンの高さとDEの中心位置と半径から求まり、 u_n はハイトスパンがもつ法線ベクトルとする。

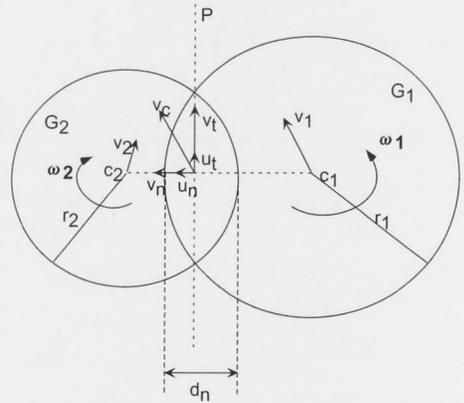


図4 DE相互の衝突
Fig. 4 Contact between two DEs

3. 隣接するセルのハイトスパンと交差しているのかも調べる。これには次の二つの場合が考えられる。

- DEの中心の高さが隣接ハイトスパンの上下面の間に位置する場合 (図3(b))、ハイトスパンの側面との衝突判定を行う。 d_n はハイトスパンの位置とDEの中心位置と半径から求まり、 u_n はハイトスパンの側面の法線ベクトルとする。
- 上の場合以外 (図3(c)) は、(2)と同様にハイトスパンの上面と下面との衝突判定を行う。

4. DEが含まれるセルと隣接するセルの両方に衝突した場合、 d_n と u_n はそれぞれの衝突計算から得られる値の平均をとることとする。

4.3.2 接触力の計算

付録Aに示したモデルを用いて2つのDE間に作用する力を計算する。

DEが物体と衝突した場合は、HSマップから求めた衝突位置での物体の法線ベクトルを u_n として、付録Aの計算を行う。物体の動きをキーフレームアニメーションで決める場合は、物体の質量を無限大とみなす。

4.3.3 DEの状態更新

あるDE(G_1)の状態更新は次の3種類の力：(1) G_1 に衝突したDEから受ける力、(2)重力、(3) G_1 が物体に衝突した場合には物体から受ける力、を考慮して次のように行う。

1. G_1 が受ける力を上記の(1)から(3)の力の総和として求める。
2. 力の大きさとDEの速さが閾値より小さい場合はDEの状態を更新しない。これはシミュレーションの安定性を高めるためである。
3. G_1 に衝突したDEと物体から受ける接線方向の力

(F) (G_1 が物体に衝突した場合) からトルクの総和を求める。

4. 運動方程式にしたがって、DE の速度、位置、角速度を更新する。
5. 空中に存在していた $DE(DE_{air})$ が着地した場合、すなわち地面にある $DE(DE_{ground})$ と衝突した時には次の処理を行う。
 - (a) DE_{air} の質量を DE_{ground} に加え、 DE_{ground} の半径をそれに応じて増やす。
 - (b) DE_{ground} の半径が大きくなった場合は、半径の小さい複数の DE に分割する。すなわち、体積の等しい 2 つの DE に分割し上下に接するように配置する。
6. DE_{air} が DEM 領域の外に位置する場合は次の処理を行う。
 - (a) ハイトフィールドの高さと DE_{air} の位置を比較することによって着地するかどうかを判定する。
 - (b) 着地する場合はハイトフィールドの高さを DE_{air} の体積分増やす。

ここで、5. の処理について補足しておく。5.(a) で空中にある DE は既に分割されているため地上にある DE に比べて小さい。したがって、融合しても地上にある DE の半径の変化は微小である。したがって、融合後に他の粒子と交差しても粒子の挙動の変化は小さいと考えられる。また、5.(b) で DE を分割したときに局所的に高さが増加するが、次節で述べるようにレンダリング時にハイトフィールドの平滑化を行うので分割時の変化は緩和される。

4.4 データの再構築

DEM 領域でのハイトフィールドとハイトスパン上の粒状物質の高さを DE の高さ (DE の中心の高さ + DE の半径) から求める。本稿で用いた方法は次のものである。

1. ハイトフィールドの各セル C について以下の処理を行う。
 - (a) C に含まれる DE について、高さが低いものから順に以下の処理を行う。
 - i. DE が一番低い場合は C の高さを DE の高さに設定する。
 - ii. そうでない場合は、DE の高さ h と C の高さ h_c を比較する。
 - iii. 高さの差が閾値より小さい場合は C の高さを DE の高さに設定する。
 - iv. 高さの差が閾値以上あり、DE の速度 v が上向きで速さが閾値以上ある場合、DE は空中に飛び散ったとみなしてハイトフィールドやハイトスパンの高さには加えない。このとき

DE を複数の細かい DE に分解し、各 DE に飛散中を示すフラグを立てておく。そして、分解後の各 DE の速度 v' は v とのなす角がある定数以下になるように v を乱数で微小変化させたものとする。また、分解後に DE 同士が交差しないように、各 DE の位置を v' の方向にシフトさせる。

- (b) DE がハイトスパンより上に位置する場合には、ハイトスパン上の粒状物質の高さについて (a) と同様の処理を行う。

2. 1. の方法で得られたハイトフィールドは、一度 DE に離散化したため凹凸が目立つので、DEM 領域内のハイトフィールドを平滑化する。

上の 1., 2. の結果、最も高い DE の高さがハイトフィールドとハイトスパンの高さとなる。

5. 計算結果

本節では地面が砂で構成されている場合の提案法による計算結果を示す。

計算結果の作成に用いたレンダリング方法は次の通りである。ハイトフィールドはポリゴンメッシュとして描画する。物体上の粒状物質の場合は、各ハイトスパン上の粒状物質を接続してポリゴンメッシュを生成し、描画する。空中に存在する粒子は点で描画する。レンダリングには OpenGL を用いた。

本節で示す例では、 128×128 の解像度のハイトフィールドを用いる。実装の単純化のため、ハイトフィールドのセルの 1 辺のサイズを単位長さとした。そして、DE の初期半径を 0.5、初期質量を 0.05 とし、1 つの DE に 30 個の粒子が含まれることとした。また、摩擦係数 μ を 0.1 とした。また、1 タイムステップを 0.02 秒とした。

ここでは、質量の異なる剛体球 (半径 15) を自由落下させた例を示す。まず、剛体球の質量が軽い場合 (DE の初期質量の 60 倍) の結果を図 5 に示す。この場合は球の質量が軽いため、砂がほとんど飛び散らない。次に剛体球の質量が重い場合 (DE の初期質量の 200 倍) の結果を図 6 に示す。球の質量が重いため、剛体球が衝突した位置の周辺の砂が多く飛び散っている。そして、図 7 は砂に剛体球を斜め下方向に投げ入れた場合の結果 (20 タイムステップごと) である。剛体球の前方に多くの砂が飛散していることがわかる。最後に別の物体を用いた例として、Bunny モデルを地面に落とした例を図 8 に示す。物体が地面から力を受けた結果、着地後に Bunny が倒れる。これの結果は紙面の都合で動画から一部のフレームを抽出したものだが、実際の動画では飛散した砂と物体や地面との衝突の表現が実現できた。

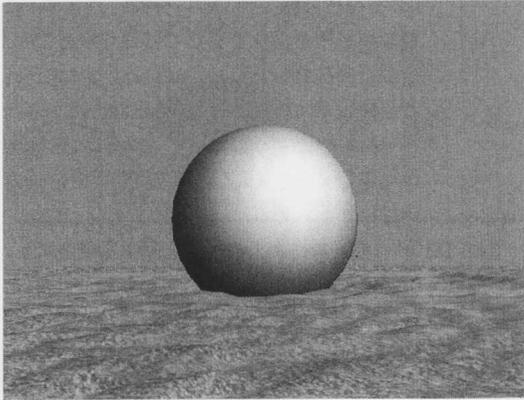


図5 剛体球を自由落下させた例 (1)
Fig. 5 Results of dropping a rigid sphere (1)

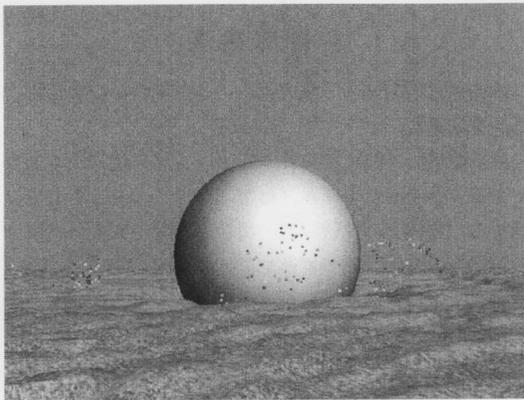


図6 剛体球を自由落下させた例 (2)
Fig. 6 Results of dropping a rigid sphere (2)

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。

6. おわりに

本稿では物体が地面に衝突したときの粒状物質の飛散アニメーションを生成する手法を提案した。提案法には以下のような特徴がある。

- ・粒子要素法を次のように拡張し物体周辺のみ適用することで、物体の衝突による粒状物質の飛散を表現できた。
 - いくつかの粒子をまとめたものをDEとし、DEの挙動を計算する。
 - 空中に飛散した場合はDEを分割して粒子の大きさにし、飛散した粒子が着地すると地面に存在するDEに融合することにより、粒子の飛散を表現

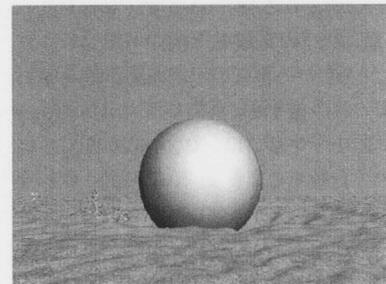
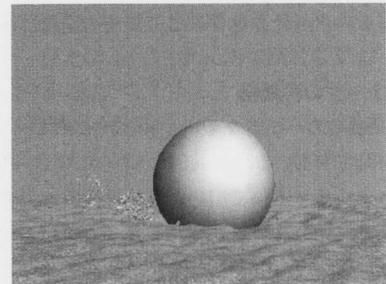
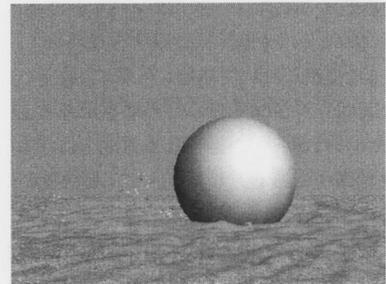
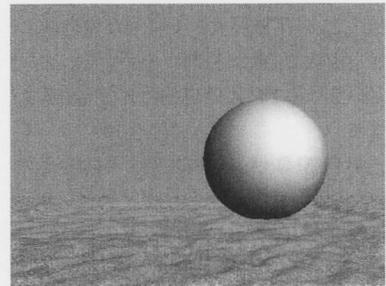
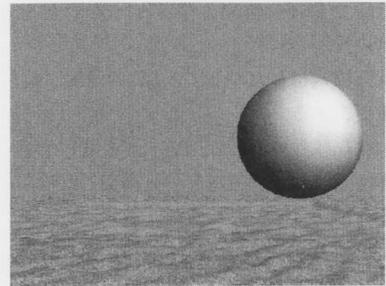
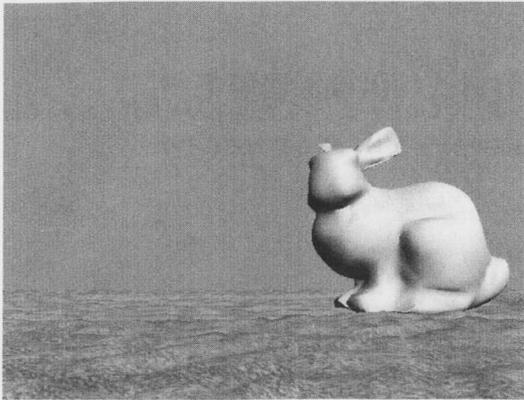
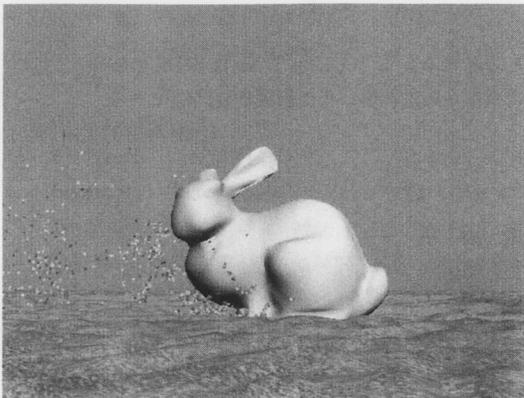


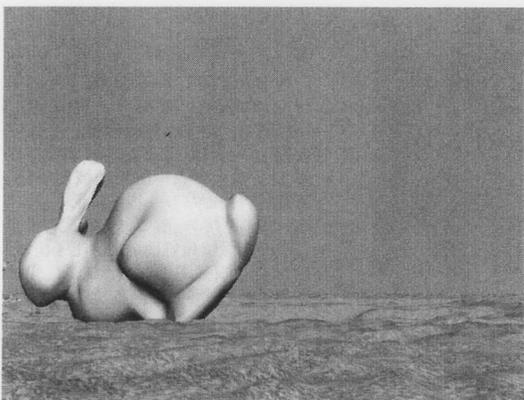
図7 砂に剛体球を斜め下方向に投げ入れた例
Fig. 7 Results of throwing a rigid sphere into sand



(a) 衝突時



(b) (a) の 35 タイムステップ後



(c) (b) の 90 タイムステップ後

図 8 Bunny を自由落下させた例
Fig. 8 Results of dropping a rigid bunny

する。

— HS マップを用いて、複雑な物体と DE との衝突判定を行う。

- 粒子が物体に及ぼす力を考慮することにより、物体の動きもシミュレーションできた。
- 粒子の集合を地面と空中のパーティクルに分類することによって、粒状物質の飛散を効率的にレンダリングできた。

埃状の粒状物質の動きや、風や物体衝突時の風圧による粒状物質の動きは本稿では扱わなかったが、これらは近年著しく進歩している、流体シミュレーションの手法を組み合わせることで、実現可能であると考えられる。これらを今後の課題としたい。

参考文献

- 1) B. Benes and R. Forsbach: "Layered data representation for visual simulation of terrain erosion", SCCG '01, pp. 80–85 (2001).
- 2) B. Chancelou, A. Luciani, and A. Habibi: "Physical Models of Loose Soils Dynamically Marked by a Moving Object", Computer Animation '96, pp. 27–35 (1996).
- 3) J. X. Chen, X. Fu, and E. J. Wegman: "Real-time simulation of dust behavior generated by a fast traveling vehicle", ACM Transactions on Modeling and Computer Simulation, Vol. 9, No. 2, pp. 81–104 (1999).
- 4) P. A. Cundall and O. D. L. Strack: "A discrete numerical model for granular assemblies", Geotechnique, Vol. 29, pp. 47–65 (1979).
- 5) P. Fearing: "Computer modelling of fallen snow", ACM Computer Graphics (Proc. SIGGRAPH '00), pp. 37–46 (2000).
- 6) J. A. Ferrez: "Dynamic Triangulations for Efficient 3D Simulation of Granular Materials", PhD thesis, Swiss Federal Institute of Technology Lausanne (2001).
- 7) A. Kunimatsu, T. Watanabe, H. Fujii, T. Saito, K. Hiwada, T. Takahashi, and H. Ueki: "Fast simulation and rendering techniques for fluid objects", Computer Graphics Forum (Proc. EUROGRAPHICS 2001), Vol. 20, No. 3, pp. 57–66 (2001).
- 8) X. Li and J. M. Moshell: "Modeling soil: Realtime dynamic models for soil slippage and manipulation", ACM Computer Graphics (Proc. SIGGRAPH '93), pp. 361–368 August (1993).
- 9) M. Matsumiya, H. Takemura, and N. Yokoya: "A virtual clay modeling system for 3d free-form design using implicit surfaces", IPSJ Journal, Vol. 42, No. 5, pp. 1151–1160 (2001).
- 10) F. K. Musgrave, C. E. Kolb, and R. S. Mice: "The synthesis and rendering of fractal terrains", ACM Computer Graphics (Proc. SIGGRAPH '89), pp. 41–50 July (1989).
- 11) T. Nishita, H. Iwasaki, Y. Dobashi, and E. Nakamae: "A modeling and rendering method for snow by using metaballs", Computer Graphics Forum (Proc. EUROGRAPHICS '97), Vol. 16, No. 3, pp. 357–364 August (1997).
- 12) J. F. O'Brien and J. K. Hodgins: "Dynamic simulation of splashing fluids", Computer Animation '95, pp. 198–205 April (1995).
- 13) Y. Oda, K. Muraoka, and N. Chiba: "Particlebased visual simulation of virtual clay", IPSJ Journal, Vol. 42, No. 5, pp. 1142–1150 (2001).
- 14) K. Onoue and T. Nishita: "A method for modeling and rendering dunes with wind-ripples", Pacific Graphics, pp. 427–428 October

- (2000).
- 15) K. Onoue and T. Nishita: "An efficient method for displaying marks on soft grounds created by objects", J. of IEEE, Vol. 32, No. 4, pp. 328-335 (2003).
 - 16) K. Onoue and T. Nishita: "Virtual sandbox", Pacific Graphics, pp. 252-259 October (2003).
 - 17) R. W. Sumner, J. F. O'Brien, and J. K. Hodgins: "Animating sand, mud, and snow", Computer Graphics Forum, Vol. 18, No. 1, pp. 3-15 (1999).
 - 18) T. Takahashi, H. Fujii, A. Kunimatsu, K. Hiwada, T. Saito, K. Tanaka, and H. Ueki: "Realistic animation of fluid with splash and foam", Computer Graphics Forum (Proc. EUROGRAPHICS 2003), Vol. 22, No. 3, pp. 391-400 (2003).
 - 19) 粉体工学会編: 粉体シミュレーション入門, Chapter 3, pp. 29-82, 産業図書 (1998).

付録 A DE 間の接触力のモデル

DE が衝突した際に 2 つの DE (G_1 と G_2) 間に作用する力のモデルを説明する。ここでは、 G_2 が G_1 に及ぼす力の計算法を示す。

まず、衝突時の力の計算に必要なベクトルをいくつか定義しておく (図 4 参照)。 \mathbf{u}_n を法線とし二つの DE の交差円を含む平面を P とする。衝突点での相対速度は次式で与えられる。

$$\mathbf{v}_c = \mathbf{c}_2 - \mathbf{c}_1 + \mathbf{u}_n \times (r_2 \boldsymbol{\omega}_2 + r_1 \boldsymbol{\omega}_1) \quad (\text{A-1})$$

\mathbf{v}_c を \mathbf{u}_n に投影したベクトル \mathbf{v}_n が相対速度の法線成分になる。

$$\mathbf{v}_n = (\mathbf{v}_c \mathbf{u}_n) \mathbf{u}_n \quad (\text{A-2})$$

また、 \mathbf{v}_c を P に投影したベクトル \mathbf{v}_t が相対速度の接線成分になる。

$$\mathbf{v}_t = \mathbf{v}_c - \mathbf{v}_n = \mathbf{v}_c - (\mathbf{v}_c \mathbf{u}_n) \mathbf{u}_n \quad (\text{A-3})$$

これを積分すると接触した DE の表面上の距離 \mathbf{d}_t が得られる。

$$\mathbf{d}_t = \int \mathbf{v}_t dt \quad (\text{A-4})$$

これらのベクトルを用いて、Cundall のモデル⁴⁾を用いて DE 間に働く力 \mathbf{F} を計算する。 F^n と F^t をそれぞれ法線、接線方向に働く力とすると、

$$\mathbf{F} = F^n (d_n \dot{\mathbf{d}}_n) \mathbf{u}_n + F^t (d_t \dot{\mathbf{d}}_t) \quad (\text{A-5})$$

と表される。ただし、

$$\|F^t\| \leq \mu \|F^n\| \quad (\text{A-6})$$

を満たすものとする。ここで、 μ は摩擦係数である。Cundall のモデルでは、 F^n と F^t はそれぞれ、

$$F^n = k_n d_n + c_n \dot{\mathbf{d}}_n \quad (\text{A-7})$$

$$F^t = k_t \mathbf{d}_t + c_t \dot{\mathbf{d}}_t \quad (\text{A-8})$$

$$k_n = k_t = \frac{m_{eff}}{t_c^2} (\pi^2 + \ln(e_n)^2) \quad (\text{A-9})$$

$$c_n = c_t = -\frac{m_{eff}}{t_c} \ln(e_n) \quad (\text{A-10})$$

$$m_{eff} = \left(\frac{1}{m_i} + \frac{1}{m_j} \right)^{-1} \quad (\text{A-11})$$

と表される。ここで、 m_i と m_j は各 DE の質量、 e_n はエネルギー散逸係数、 t_c は衝突継続時間である。

(2004 年 2 月 12 日受付)

尾上 耕一



平 12, 東京大学理学部卒。平 14, 同大学院新領域創成科学研究科修了。現在、同大学院新領域創成科学研究科博士後期課程に在学中。主として、コンピュータグラフィックス、特に自然物の表示に関する研究に従事。情報処理学会会員。

西田 友是 (正会員)



昭 46, 広島大学工学部卒業。昭 48, 同大学院工学研究科修了。同年、マツダに入社。昭 54 年、福山大学工学部講師。昭 59, 同助教。平 2, 同教授。平成 10, 東京大学理学部教授。平 11, 同大学院新領域創成科学研究科教授となり、現在に至る。コンピュータグラフィックスの研究に従事。工学博士。情報処理学会、電子情報通信学会、画像電子学会、ACM、IEEE 各会員。