

風紋・砂丘を含む砂漠景観の表示法

尾上 耕一[†] 西田 友是[†]

A Rendering Method on Desert Scenes of Dunes with Wind-ripples

Koichi ONOUE[†] and Tomoyuki NISHITA[†]

あらまし 本論文ではコンピュータグラフィックス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。次に砂丘と風紋を組み合わせて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 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、砂埃を考慮している。最後にレンダリング結果として砂丘表面上の風紋、砂丘の影、風紋の影、の有無を比較した画像を作成しそれぞれの効果を示した。

キーワード 砂漠, 砂丘, 風紋, Level of Detail, 影, 砂埃

1. はじめに

近年、コンピュータグラフィックスを用いて雲、海、山などの自然景観を表示しようとする研究が多くなされている。一方、風による砂の動きによる砂丘の形成や、一定方向にある程度以上の強さの風が吹くことで形成される縞模様である風紋の形成については長い間研究がなされてきた [1]。近年、それらを単純なモデルを用いて表現することが試みられている [2], [3]。本研究はこうしたモデルを用いた砂丘と風紋のモデリングと風紋を考慮した砂丘の高速かつリアルなレンダリングをするためのシステムの作成を目的としている。

砂丘や風紋の形成モデルとしては代表的なものが二つ提案されている。西森らは砂の運動について二つのスケールの単純なモデルを提案した [2]。彼らは小規模モデルでは風力がある閾値を越えたとき風紋が自然に形成されることを示した。また、大規模モデルによって砂丘の形成を論じた。大内らは砂粒子のサイズを二種類に分けて考えることによって風紋の断面での砂粒子の分布をモデル化した [3]。しかし、砂粒子の微妙なサイズの違いは、砂丘と風紋を 3 次元表示したときの外見にはほとんど影響を及ぼさないと考えられる。

本研究ではモデリング法として西森らの方法 [2] を用いた。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。彼らのモデルを採用した理由は、物理的に厳密ではないが外観としては十分な砂丘と風紋の形状が得られ、また物理シミュレーションを行うより計算時間が少なくすむと考えられるからである。ただし、彼らの目的は砂丘と風紋形成のモデル化にあったので、彼らは 3 次元表示を行っていない。

そこで、われわれは風紋を考慮した砂丘のレンダリング法を提案する。本研究ではハイトフィールドのメッシュで表された砂丘表面へ風紋パターンをバンプマッピングすることによってレンダリングを行う。このときの計算時間を節約するために、ポリゴン数の削減や LOD (Level of Detail) を考慮する。また、バンプマッピングの計算と同時に砂丘の影と風紋の影の計算も行う。

映画などで砂漠を含むシーンでは、地を這うような砂埃がよく見られる。砂漠景観を含むアニメーションでは、よりリアルな表現を行うためには砂埃の表示も必要である。そこでパーティクルシステムを用いた砂埃の表示も行う。

[†] 東京大学大学院新領域創成科学研究科, 東京都
Graduate School of Frontier Sciences, The University of
Tokyo, Hongou 7-3-1, Bunkyo-ku, Tokyo, 113-0033 Japan

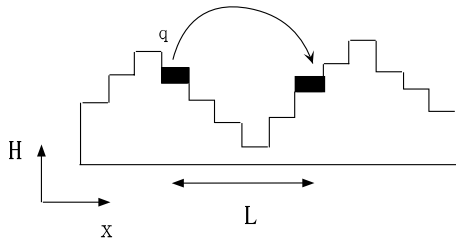


図 1 跳躍運動
Fig. 1 The saltation movement.

2. 砂丘と風紋のモデリング

2.1 風紋の形成

風紋のハイトフィールドを $H(x, y)$ で表す。風による二種類の砂の運動をモデル化する。第一は跳躍運動で、式 (1) から (3) のように各点の高さに比例した飛距離 L を一定量 q の砂が飛ぶものとする。

$$H_{n'}(x, y) = H_n(x, y) - q, \quad (1)$$

$$H_{n'}(x + L(H_n(x, y)), y) = H_n(x + L(H_n(x, y)), y) + q, \quad (2)$$

$$L = L_0 + bH_n(x, y). \quad (3)$$

ここで、 L_0, b は飛距離のパラメータであり n は時間ステップである。1 ステップに跳躍と転がり運動の計算が含まれるので n' は n と $n+1$ の中間の時間ステップである。図 1 は砂粒の跳躍運動を示したものである。第二は転がり運動で、次式のように H の各点 (x, y) での砂を周囲に等方的に分散させる。

$$H_{n+1}(x, y) = H_{n'}(x, y) + D \left[\frac{1}{6} \sum_{NN} H_{n'}(x, y) + \frac{1}{12} \sum_{NNN} H_{n'}(x, y) - H_{n'}(x, y) \right]. \quad (4)$$

ここで、 D は拡散係数、 NN は (x, y) に縦横に隣接した点、 NNN は斜めに隣接した点を表す。図 2 および 3 はこの場合の砂の動きを示したものである。

この二つの計算をハイトフィールド全体について繰り返し適用することにより、風紋の動きの時間変化がシミュレートされる。この計算では周期境界条件を用いる。

2.2 砂丘の形成

砂丘のハイトフィールドも風紋の場合と同様に $H(x, y)$ で表す。砂の運動については、転がり運動は風紋の場合と同じモデルを用いる。跳躍運動につい

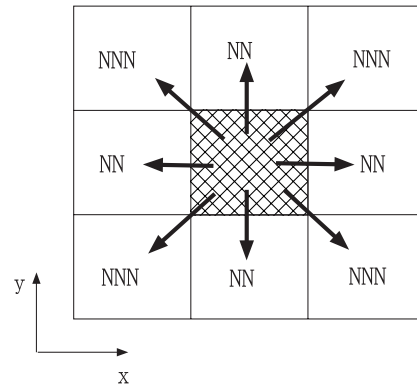


図 2 転がり運動
Fig. 2 The creep movement.

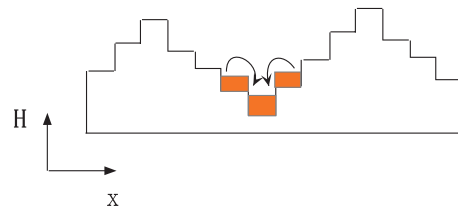


図 3 転がり運動 (断面)
Fig. 3 The creep movement shown in cross section.



図 4 砂丘の場合の跳躍運動
Fig. 4 The saltation movement at dunes.

ては、風紋の場合より規模が大きくなるので、斜面の傾きと風向を考慮してモデル化する。まず、次式のように飛砂量 q は砂丘の風上側では大きく、風下側では小さくする。

$$q = q(\nabla \cdot H_n(x, y)) = q_0 + b' \tanh(\nabla \cdot H_n(x, y)). \quad (5)$$

ここで、 q_0 と b' は飛砂量のパラメータである。

また、砂丘の風上側で跳躍した砂はすぐ斜面に衝突するので飛距離 L は小さくし、風下側では大きくする。

$$L = L(\nabla \cdot H_n(x, y)) = L_0 - b \tanh(\nabla \cdot H_n(x, y)). \quad (6)$$

砂丘での砂の跳躍運動を図 4 に示す。この図では矢印の太さが飛砂量を示している。

風紋の場合と同様にハイトフィールド全体について跳躍と転がり運動についての計算を繰り返すことにより、砂丘が形成される。

3. 風紋を考慮した砂丘のレンダリング

3.1 概要

砂丘と風紋はハイトフィールドで表されている。砂丘はハイトフィールドメッシュの各四角形を OpenGL を用いてレンダリングする。メッシュの各四角形を表示する際に風紋をバンプマッピングする。レンダリングプロセスの概要は次のようになる。

(1) 風紋のハイトフィールドから法線ベクトルを計算する。

(2) 砂丘の各四角形 F について

(a) F が視点に対して裏向きか視界 (view frustum) の外にある場合、 F は表示されない。

(b) そうでないとき、LOD の手法を用いて視点と F の距離に応じた解像度の風紋を F にバンプマッピングする。同時に影の計算も行う。

(3) 砂丘表面上に砂埃を表示する。

次節以降でレンダリングについての詳細を述べる。

3.2 砂丘の表面上への風紋のマッピング

砂丘の表面に風紋のパターンを表現する単純な方法としては、(1) 風紋を表現できるほど砂丘表面を構成するポリゴンを細かくする方法、(2) 風紋を上から見た場合の画像を、砂丘の表面にテクスチャマッピングする方法、の二つが考えられる。しかし、(1) の方法では、ポリゴンの数が増えるため描画に時間がかかり、(2) の方法では描画は高速だがテクスチャにリアルさが欠ける。そこで、提案法では砂丘表面を構成する各ポリゴンについて、風紋ハイトフィールドをバンプマッピングすることによって砂丘表面上の風紋の凹凸を表現する。すなわち、図 5 のように、(a) 風紋のハイトフィールドからハイトフィールドの各頂点の法線ベクトルを求める (得られた結果を法線マップと呼ぶ)。 (b) その法線ベクトルを砂丘の斜面にあわせて傾ける。 (c) 傾けた法線ベクトルと光線ベクトルから砂丘表面の輝度を求める。これにより、ポリゴン数を増やさずに砂丘の表面上に風紋が表現できる。

3.3 計算時間の節約

前節の方法で砂丘を構成するすべてのポリゴンにバンプマッピングをしようとすると計算に多くの時間がかかってしまう。そこで、バンプマッピングを適用する砂丘のポリゴン数を削減し、バンプマッピング自体の計算時間を短縮するために LOD を使用した。

3.3.1 ポリゴン数の削減

視線ベクトルと法線ベクトルの内積の正負判定に

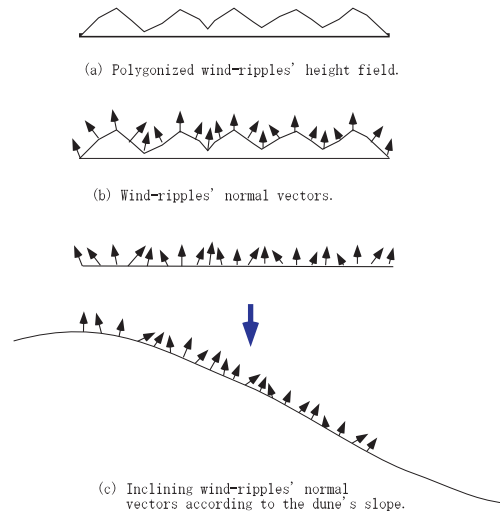


図 5 風紋の砂丘へのバンプマッピング

Fig.5 Bump mapping wind-ripples onto dunes.

よって、視点に対して裏向きのポリゴンを削除する。また、視界に入らないポリゴンへの風紋のバンプマッピングの計算を省略する。これは、視線とメッシュを xy 平面等に投影して容易に判定できる。

3.3.2 LOD を用いたバンプマッピング

LOD とは物体の細かさのレベルのことで、一般に視点から近い物体には細かい幾何情報をもたせ、視点から離れた物体には粗い幾何情報をもたせることでレンダリング結果の画像に影響なくレンダリング時間を短縮するために用いられる。レベルに応じてポリゴンの数を減らす方法は多く研究されているが、提案法では風紋をバンプマッピングで表しているの、風紋の解像度をレベルに応じて変化させることを考える。具体的には視点から近い部分にあるポリゴンに対応した風紋の解像度は高くし、視点から遠い部分にあるポリゴンに対応した風紋の解像度は低くする (図 6 参照)。具体的には以下の方法をとる。

3.3.1 節で求めた可視面に対してさらに LOD を考慮して計算量を節約する。この LOD はポリゴンのスクリーンへの投影面積から決める [5]。砂丘を表示する前にあらかじめ風紋の法線マップ^{注1)}を LOD のレベル数用意しておく (例えば、 $256 \times 256, 128 \times 128, 64 \times 64, 32 \times 32$ の 4 段階)。最も高いレベル (最も解像度が高い) の法線マップは風紋の

(注1): 風紋のハイトフィールドから求めた法線ベクトルの二次元配列のこと。

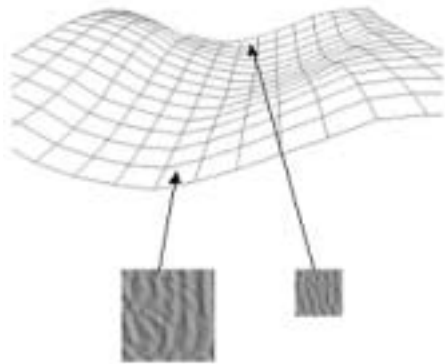


図 6 視点からの距離に応じた風紋の解像度の変化
Fig. 6 Changes of resolutions according to distances from viewpoint.

ハイトフィールドから求める．他のレベルの法線マップは，一つ上のレベルの法線マップの隣接した4つの法線ベクトルの平均をとることによって得る．そして，各ポリゴンの LOD に対応したレベルの風紋の法線マップを用いて前節の方法でバンプマッピングの計算をする．

3.4 砂丘の影の計算

リアルな画像の生成には影は不可欠である．影の計算法としては，物体を水平な平面に投影したものを影とする方法 [6]，シャドウボリュームを用いた方法 [7]，シャドウマップを用いた方法 [8] ~ [10]，レイトレーシング法，などが代表的なものである．物体を水平な平面に投影したものを影とする方法は，物体から水平な地面に落ちる影しか表現できず，砂丘の斜面への適用は困難である．またシャドウボリュームを用いた方法は光源とポリゴンからシャドウボリュームを生成するため，ポリゴン数が多くなると計算に時間がかかる．シャドウマップ法はハードウェアの機能を用いて非常に高速に影を生成でき，また Z バッファの値を利用するので，Z バッファ法で表示できるものなら物体の形状を考慮することなく影の計算ができる．しかし，バンプマッピングによって表現された物体上の凹凸の影までは表示できない．

そこで，砂丘の影の計算はレイトレーシング法を用いて行う．これは，計算速度よりもリアルさや精度を重視しているためである．また，レイトレーシング法を用いることにより砂丘の影だけでなく，風紋によってできる影の計算も同時に行うことができるという

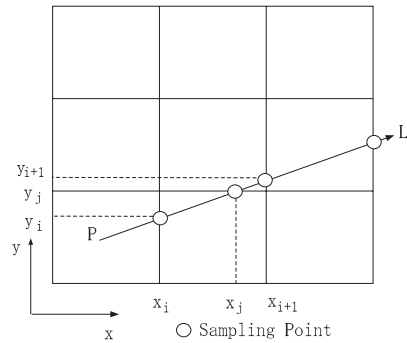


図 7 影の計算におけるサンプリング点の決定
Fig. 7 The determination of sampling points in the shadow calculation.

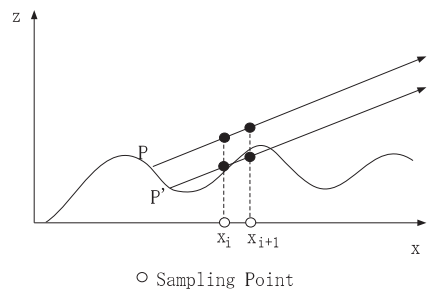


図 8 ハイトフィールドでの影の計算
Fig. 8 The shadow calculation in the height field.

利点がある．風紋の凹凸を考慮した砂丘の影の計算法 [11] を用いることもできるが，砂丘の影の大きさに比べて風紋の凹凸は微小なのでここでは考慮しない．ここではレイと各ポリゴンの交差判定を行うのではなく，メッシュ上でレイとハイトフィールドの交差判定を行うことでレイトレーシング法を簡略化した．つまり以下のように計算した．

ハイトフィールドでのレイトレーシング法による影の計算は次のように行う．風紋をバンプマッピングする時に砂丘ハイトフィールド上の各点 P について， P から光源に向かってレイを飛ばし， xy 平面上でレイとハイトフィールドのポリゴンメッシュとの交点をサンプリング点とする (図 7 参照)．サンプリング点とそこでのレイの高さはともに増分法により効率よく計算できる．そして，サンプリング点でのハイトフィールドとレイの高さを比較することによって P が影に入るかどうかを判定する．例えば，図 8 の場合， x_i, x_{i+1} をサンプリング点， L を光源方向としたとき，点 P には光が当たり，点 P' は影に入る．

3.5 風紋の影の計算

砂丘の影の計算 (3.4 節) で点 P が光に照らされていると判定されたとき、さらに P が風紋によってできる影に含まれているのかも判定する。ここではその判定に地平線計算に基づく方法 [12] を用いた。この方法は前処理を用いた近似的な影の計算方法である。この方法を用いた理由は、前述の砂丘の影の計算と組み合わせるのが容易であるし、風紋の影がレンダリング結果に占める割合は小さいので近似的な方法でも問題ないと考えられるからである。この方法をグラフィックスハードウェアの機能を利用して高速化した Heidrich らの方法 [13] もあるが、使用できるハードウェアの種類が限られるのでここでは用いていない。以下に風紋の影の計算のための前処理と影の判定について説明する。

3.5.1 風紋の影の計算のための前処理

風紋の影を表示するために風紋ハイトフィールドメッシュの各格子点で、いくつかの方向についてその方向にハイトフィールドのほかの部分が見えるかどうかという情報を前もって計算しておく（この計算を地平線計算と呼ぶことにする）。

(1) 風紋ハイトフィールドの各格子点 P について (2), (3) を行う。

(2) 図 9 のように xy 平面上で縦横斜め 8 方向の各方向 D について (3) を行う。

(3) 格子点 P から方向 D に向かって順に格子点をたどり、 P と各格子点との間の高さの最大の傾きを求める（これを以後地平線の傾きと呼ぶことにする）。

例えば、図 10 の場合、点 $P(x, y)$ から x 軸正方向に $(x-1, y)$ までの各格子点^(注2)について、 P からその点までの傾きを計算したとき、 PP' 間の傾きが最大となるのでこの傾きを記憶しておく。

3.5.2 風紋の影の判定

砂丘の影の計算で点 P が砂丘の影に含まれないと判定された場合、地平線計算の結果を用いて P が風紋の影に含まれるかどうかを判定する。そのためにまず、光線ベクトルをオブジェクト座標系に変換する。次に、変換された光線ベクトルと地平線計算で求めた 8 サンプル方向の地平線の傾きから線形補間によって求めた光線方向での地平線の傾きを比較する。その結果、光線ベクトルの傾きが地平線の傾きより大きければ P は

(注2): すなわち、点 $P(x, y)$ から $(x+1, y), (x+2, y), \dots, (XSIZE-1, y), (0, y), (1, y), \dots, (x-1, y)$ という順に点をたどる。ここで、 $XSIZE$ は風紋ハイトフィールドの x 方向のサイズである。

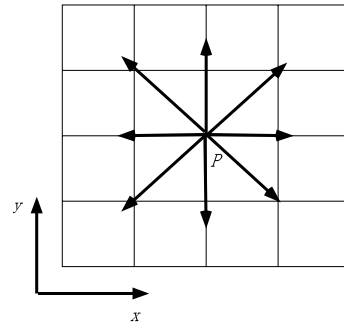


図 9 ハイトフィールドでの地平線計算

Fig. 9 The calculation of horizon in the height field.

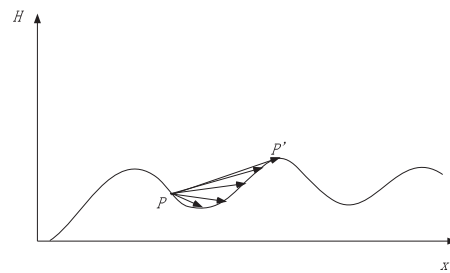


図 10 ハイトフィールドでの地平線計算 (断面)

Fig. 10 The calculation of horizon in the height field shown in cross section.

光に照らされ、地平線の傾きの方が大きければ P は風紋の影に含まれるということになる。

3.6 砂の粒状感の表現

風紋をバンプマッピングしただけでは砂丘の表面上に滑らかに風紋ができるのみで砂の粒状感は表現できない。そこで、粒状感を出すために砂表面の画像をテクスチャマッピングする。砂表面の画像とは水平に砂を敷き詰めたものを真上から、砂の粒状感が認識できる程度の距離で見たもののことである。この砂表面の画像を砂丘にテクスチャマッピングしておき、そこに風紋のバンプマッピングを施す。これによって砂の粒状感を表現する。ここでは、砂表面のテクスチャは乱数で初期化したハイトフィールドを真上から見下ろしたときの画像を用いた (図 11 参照)。

3.7 砂埃の表示

砂漠景観の表現力を高めるために風が吹いているときによく見られる砂埃の表示を行う。ここで想定している砂埃とは、砂丘全体から見ると局所的にできる地を這うような砂埃である。このため、砂丘の起伏によってできる風の乱流や渦は無視できると考え、風紋形成モデルを使用して次のように砂埃の表示を行った。

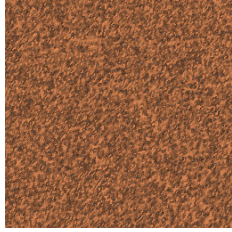


図 11 砂表面のテクスチャの例
Fig. 11 An example of the sand surface texture.

砂埃を構成する砂粒の数は非常に大きいので、砂粒の座標を1つ1つ計算して表示しようとするると計算時間が膨大なものとなる。そこで砂粒の密度が中心で高く中心から離れるにしたがって低くなるような球状の集合体を考え、この球状の砂粒集合体を複数表示することによって近似的に砂埃を表現した。1つの砂粒集合体の表示には視線と垂直な透明な小さい四角形(これをビルボードと呼ぶ)を用いる。このビルボードには球状の砂粒集合体を投影し、光の減衰率は砂粒の密度に比例するものとして、背景の色と混合して表示する。このとき OpenGL のテクスチャマッピング機能とアルファブレンディング機能を利用して高速に表示することができる。

砂丘上に砂埃を表示するために、風紋ハイトフィールドの格子点に対応する砂丘上の各位置から砂丘表面上に沿うようにビルボードを飛ばす。風は x 軸に沿って吹いていると仮定するので、あるタイムステップ t でのビルボードの位置 (x_t, y_t, z_t) は次の式で計算する。

$$\begin{aligned} x_t &= x_{t-1} + \Delta x + \alpha x, y_t = y_{t-1} + \alpha y \\ z_t &= H_{dune}(x_t, y_t) + \Delta z + \alpha z \end{aligned} \quad (7)$$

ここで Δx は x 軸方向にビルボードが1タイムステップで平均的に進む距離である。 $H_{dune}(x_t, y_t)$ は (x_t, y_t) での砂丘の高さを表す。 Δz は砂丘表面上からのビルボードの高さである。 $\alpha x, \alpha y, \alpha z$ は砂埃の動きに乱れを持たせるための乱数である。ビルボードの大きさは風紋ハイトフィールドの各位置の高さに比例した値に設定する。

4. 計算結果

図 12 から図 18 にレンダリングの結果を示す。図 13 は図 12 の砂丘に風紋をバンプマッピングし、砂表面のテクスチャをマッピングしたものである。風紋生成時に用いたパラメータは $L_0 = 7.3, q = 0.6, b = 2.0$

であり、砂丘生成時に用いたパラメータは $L_0 = 7.0, q_0 = 1.5, b = 2.0, b' = 2.0$ である(パラメータの設定は西森らの論文[2]を参考にした)。ここでは 64×64 の配列で砂丘を表し、砂丘を構成する各四角形に最大 64×64 のサイズの風紋をマッピングしている。このとき LOD を4段階に分け、レベルに応じて $64 \times 64, 32 \times 32, 16 \times 16, 8 \times 8$ のサイズの風紋マッピングした。背景の空は、基底関数を用いた高速な空の色の計算法[14],[15]を用いてレンダリングを行った。図 12 および 13 は LOD 等を使用した結果である。図 13 を LOD 等を使用しないでレンダリングしてみた結果の画像と比較したところ、2枚の画像の差分画像の輝度値の平均が255段階で3.92となり、LODを使用しても結果の画像に変化がほとんどないことがわかった。この例では、LOD等を用いない場合、風紋をバンプマッピングする計算時間は約15秒であったが、3.3.1のポリゴン数の削減法を用いることによって、計算時間は約9.7秒となった。さらにLODを用いることによって計算時間は約1.5秒となり、LOD等を用いない場合に比べて計算時間は約1/10に短縮できた。ただし、ここでは影の計算時間は含めていない。図 14 および 15 は太陽高度を下げた場合の影の効果を示す例である。影なしの場合の計算時間は約1.5秒、砂丘の影を付けた場合の計算時間は約9.1秒であった。図 16 および 17 は風紋の影の効果を示したものである。風紋の影も付けた場合の計算時間は約10.2秒(前処理6.8秒)であった。風紋の影も付けても砂丘の影だけの計算よりそれほど計算時間がかからないことがわかる。実際の砂漠景観では風紋だけでなく人や動物が歩いた足跡がつくことがある。そこで、砂丘表面上の足跡を Sumner の方法[16]を用いて生成した。足跡と砂埃も加えてレンダリングした結果は図 18 のようになった^(注3)。これらの画像サイズは 720×480 で、使用した CPU は Pentium III 1GHz、ビデオカードは Nvidia GeForce3 である。

5. おわりに

本論文では砂丘と風紋のモデリング法を議論し、それらを組み合わせるシステムを提案した。提案したシステムにより、砂丘の表面に風紋をマッピングしたレンダリングを、リアルでかつ高速に実行できるようになった。本論文の特徴を以下にまとめる。

(注3): 砂埃は静止画では存在がわかりにくいので下記の動画を参照されたい。(http://nis-lab.is.s.u-tokyo.ac.jp/~onoue/sand/)

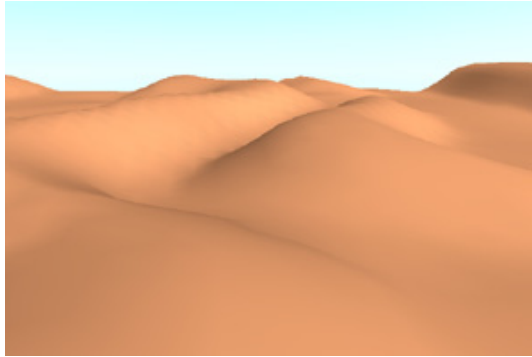


図 12 風紋がない場合の砂丘
Fig. 12 Dunes without wind-ripples.

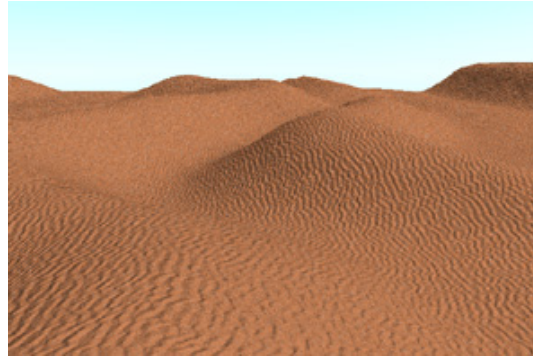


図 13 風紋がある場合の砂丘
Fig. 13 Dunes with wind-ripples.



図 14 影がない場合の砂丘
Fig. 14 Dunes without shadows.



図 15 影をつけた場合の砂丘
Fig. 15 Dunes with shadows.

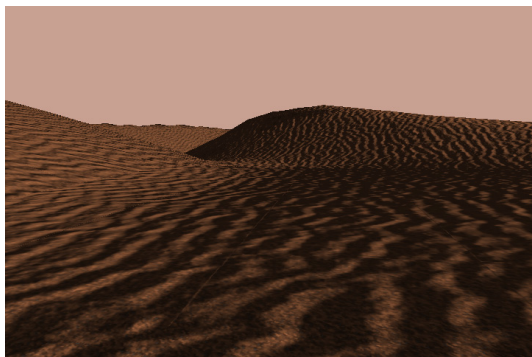


図 16 風紋の影がない場合の砂丘
Fig. 16 Dunes with shadows (without wind-ripples' shadows).

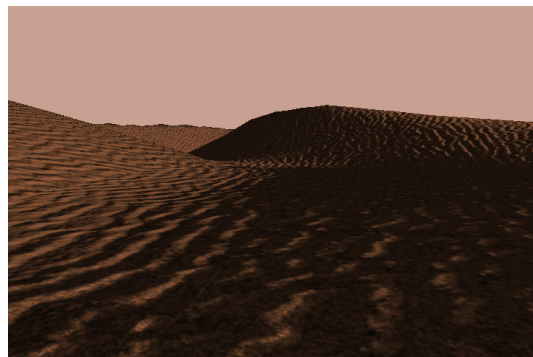


図 17 風紋の影をつけた場合の砂丘
Fig. 17 Dunes with shadows (with wind-ripples' shadows).

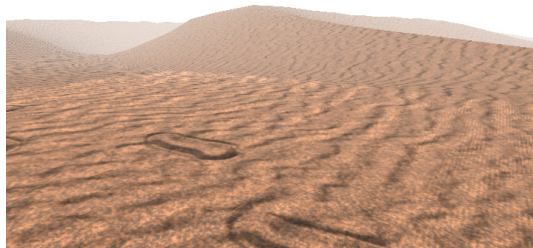


図 18 砂埃と足跡も表示した場合の砂丘
Fig. 18 Dunes with sand smoke and foot prints.

(1) ハイトフィールドを用いた砂丘と風紋形成のモデルを使用した。

(2) 砂丘の表面上に風紋を表現するためにバンプマッピングと LOD を組み合わせる方法を用いた。風紋の法線マップを LOD に応じて複数用意することによって効率的に計算できた。

(3) 砂丘の影の計算を簡略化したレイトレーシング法によって行った。LOD を利用したことによって効率的にリアルな影も表示できた。

(4) 風紋によってできる細かい影の表示も行った。地平線計算を利用して砂丘の影だけの場合とそれほど変わらない時間で風紋の影も計算できた。

(5) 風紋形成の結果を利用して砂丘表面上の砂埃の表示を行い、よりリアルな表現を可能にした。

今後の課題としてはレンダリングのさらなる高速化などが考えられる。

謝 辞

最後に本論文について有意義な議論をしてくださった土橋助教授（北海道大学）に深謝いたします。

文 献

- [1] K. Pye and H. Tsoar, *Aeolian Sand and Sand Dunes*, pp.88-220, Unwin Hyman, London, 1990.
- [2] H. Nishimori and N. Ouchi, "Formation of Ripple Patterns and Dunes by Wind-Blown Sand," *Physical Review Letters*, Vol.71, No.1, pp.197-200, July 1993.
- [3] N. Ouchi and H. Nishimori, "Modeling of wind-blown sand using cellular automata," *Physical Review*, B 52, 5877-5880, 1995.
- [4] J. Blinn, "Simulation of wrinkled surfaces," *Computer Graphics (SIGGRAPH '78 Proceedings)*, Vol.12, No.3, pp.286-292, August 1978.
- [5] L. Williams, "Pyramidal Parametrics," *Computer Graphics*, Vol.7, No.3, pp.1-11, July 1983.
- [6] J. Blinn, "Me and my (fake) shadow," *IEEE Computer Graphics and Applications*, Vol.8, No.1, pp.82-86, 1988.
- [7] F. Crow, "Shadow algorithms for computer graphics," *Computer Graphics (SIGGRAPH '77 Proceedings)*, Vol.11, No.2, pp.242-248, 1977.
- [8] L. Williams, "Casting Curved Shadows on Curved Surfaces," *Computer Graphics (SIGGRAPH '78 Proceedings)*, Vol.12, No.3, pp.270-274, August 1978.
- [9] W. Reeves, D. Salesin, and R. Cook, "Rendering Antialiased Shadows with Depth Maps," *Computer Graphics (SIGGRAPH '87 Proceedings)*, Vol.21, No.4, pp.283-291, July 1987.
- [10] M. Segal, C. Korobkin, R. Widenfelt, J. Foran, and P. Haerberli, "Fast Shadows and Lighting Effects Using Texture Mapping," *Computer Graphics (SIGGRAPH '92 Proceedings)*, Vol.26, No.2, pp.249-252, July 1992.
- [11] T. Noma, K. Sumi, "Shadows on Bump-Mapped Surfaces in Ray Tracing", *The Visual Computer*, Vol.10, No.4, pp.330-336, 1994.
- [12] N. Max, "Horizon mapping: shadows for bump-mapped surfaces," *The Visual Computer*, 4(2), pp.109-117, July 1988.
- [13] W. Heidrich, K. Daubert, J. Kautz, H. Seidel, "Illuminating Micro Geometry Based on Precomputed Visibility", *Computer Graphics (SIGGRAPH 2000 Proceedings)*, pp.455-464, July 2000.
- [14] T. Nishita, T. Shirai, K. Tadamura, and E. Nakamae, "Display of The Earth Taking into Account Atmospheric Scattering," *Computer Graphics (SIGGRAPH '93 Proceedings)*, Vol.27, No.4, pp.175-182, August 1993.
- [15] Y. Dobashi, T. Nishita, K. Kaneda, and H. Yamashita, "A Fast Display Method of Sky Colour Using Basis functions," *The Journal of Visualization and Animation*, Vol.8, pp.115-127, 1997.
- [16] R. Sumner, J. O'Brien, J. Hodgins, "Animating Sand, Mud, and Snow". *Proceedings of Graphics Interface '98*, June 17-21, pp.125-132.