

# Granular Material Interactive Manipulation: Touching Sand with Haptic Feedback

Bedřich Beneš  
Computer Graphics Technology  
Purdue University  
bbenes@purdue.edu

Enkhtuvshin Dorjgotov  
Computer Graphics Technology,  
Purdue University  
edorjgo@purdue.edu

Laura Arns  
Envision Center,  
Purdue University  
larns@purdue.edu

Gary Bertoline  
Envision Center,  
Purdue University  
bertoline@purdue.edu

## Abstract

We present a novel approach for a virtual granular material interactive manipulation with force feedback. A user can interactively change a height-field model of sand by dragging objects inside. The virtual sand behaves like real sand moving to the sides and falling back, filling holes and irregularities on the surface. The dragging object position is controlled by a haptic device that provides a position in 3D space that correspondingly changes the sand model. The sand model provides force feedback to the haptic device resulting in two principal forces; the repulse that has vertical direction and the viscous drag. The resulting force is delivered back to the haptic device. The user can sense the sand's response as the dragging object moves through the virtual sand. This results in the haptic-visual feedback providing a higher degree of plausibility than visual feedback alone.

**Keywords:** Sand, Tactile Devices, Virtual Reality, Simulating Natural Phenomena.

## 1 INTRODUCTION

Terrain modeling algorithms and techniques published in the area of computer graphics such as [1, 2, 3, 5, 6, 11, 10, 14] have focused mostly on obtaining visually realistic physically-based models. Little attention has been paid to real-time and interactive techniques for producing realistic shapes of terrains. Interactive manipulation techniques, such as [7, 8, 9, 13] mostly focus reaction of sand or soils to a certain user action. One of the first commercially successful applications providing interactive terrain modeling was the Metacreations Bryce 3D®.

We present a new algorithm for interactive modeling of sand shape using haptic devices. A user "touches" the virtual sand, which changes the shape correspondingly (see Figure 1). The potential users and applications of this algorithm are anyone that needs to rapidly and realistically model the shape of sand. There are many interactive modelers for different data types. For example, there is the NURBS sculptor in MAYA®, hair painter in the same software, particles can be modeled in 3D MAX® and MAYA®, etc. The authors are not aware of any application that allows interactive modeling of sand.

The paper begins with a description of the previously published modeling and simulating techniques that work with sand in the area of computer graphics.

In Section 3 we describe the real-time modeling and displaying of sand, and Section 4 describes the haptic device and the force feedback from the sand. Section 5 discusses the implementation issues and the last part shows results of our experiments and concludes the paper.



Figure 1: WSCG written in sand

## 2 RELATED WORK

The previous work can be roughly divided into two significant classes: the real-time algorithms and the non-real-time algorithms. We will first discuss the historically older group of non-real-time algorithms.

### 2.1 Non-Real-time Sand Manipulation

Probably the first paper dealing with sand simulation is the erosion simulation algorithm of Musgrave et al [10]. Material is described as a regular height field and is eroded by thermal and hydraulic erosion. Some parts are broken by the thermal shocks and fall down. Some amount of material is dissolved by running water and deposited to a different location. The shape morphology is described as a set of parameters that defines the material properties.

Sumner et al [13] introduced a technique for animating sand and soil motion that results from interactions

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

WSCG 2006 conference proceedings, ISBN 80-86943-03-8  
WSCG'2006, January 30 – February 3, 2006  
Plzen, Czech Republic.  
Copyright UNION Agency – Science Press

with tools. Their technique is able to simulate material displacement that results from footsteps and trails in the sand and mud.

An interesting approach for modeling wind ripples in sand was shown by Onoue and Nishita [7]. The sand particles reallocation is formed by the two principal factors - creep and saltation. These phenomena are described by a set of equations that is applied to the sand model represented as a regular height field.

Beneš and Forsbach [1] introduced a layered data structure that allows for representation of volumetric effects, such as caves, as well as for a surface erosion. They use a RLE-like data structure that represents layers of deposited material. This volumetric technique reports comparable speed of erosion simulation as in the case of height-fields that is significantly better than voxel-based approaches.

Zhu and Birdson [14] recently presented application of the Navier-Stokes equations to simulation of sand motion. Sand is represented as a cloud of particles and its motion is described by the physically correct equations. Special attention is paid to the surface tracking and rendering.

## 2.2 Real-time Sand Manipulation

The real-time manipulation algorithms are represented by the following previous work.

Li and Moshell in [9] described a physically-based model of real-time digging and caving. The model works with extended regular height-fields and describes the surface changes.

The papers [1, 13] were the motivation for the Virtual Sandbox of Onoue and Nishita [8]. Real-time manipulation of sand is described by the dual sand representation; as a height field and as a set of particles. Particles are used for sand that is elevated over the surface of the height-field and the height-field is used for deposited sand. The authors report interactive frame rates for sand manipulation and interaction.

Neidhold et al [11] recently introduced another algorithm for terrain shape morphology that deals with homogenous sand-like structures and allows for real-time hydraulic erosion simulation.

All of the previous work focuses on shape morphology and the interaction is limited to the user working with a mouse as an input device or to real-time displaying of erosion processes. We propose a haptic-device interaction that provides another level of fidelity to users who interact with the granular material model.

## 3 REAL-TIME SAND SIMULATION

One of the principal advantages of sand, from the viewpoint of modeling in computer graphics, is the fact that it cannot form concave structures. That is why sand is usually represented as a regular height-field; a

two dimensional matrix, where each vertex corresponds to the elevation of the given location. The elevation points are regularly sampled in two dimensional space, which simplifies algorithms for the terrain metamorphosis simulation. On the other hand this representation is space demanding. Regardless of the level of detail or the amount of terrain roughness the amount of space occupied is the same. We use regular height-fields to store the model of the sand.

There are two aspects of the sand manipulation to be considered. The sand relocation by user action (*displacement*) and sand returning to the stable position by gravity (*erosion*). Our algorithm is a two pass algorithm. In the first pass the user contacts the sand with a virtual tool and the dragging displaces a certain amount of the material. In the second pass the sand is eroded back. We will describe both passes in depth.

### 3.1 Sand displacement

As an object penetrates the sand surface it displaces a certain amount of the granular material. This displacement depends on the force vector that represents the drag direction, the collision area between the object, and on the object and the sand boundary shape. The force is responsible for the depth the objects enter the granular material, and its shape determines the amount of pushed sand. General calculation can be very demanding, but careful scene preparation can simplify it.

We use a sphere as the dragging object. It has the principal advantage of having the same shape facing the sand regardless of dragging direction. The amount of displaced material can be easily determined.

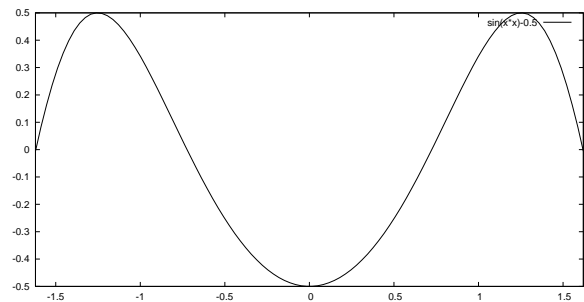


Figure 2: The cross-section of the filter applied to replace the sand

To make the calculation even faster we introduce a concept of two two dimensional erosion discrete filters. The first one has circular shape and corresponds to the area that penetrates the sand and all values in this filter are negative. We denote this filter by  $neg_{i,j}$ . The other filter has positive values, has a shape of circular rampart, and is located around the negative filter. It is denoted by  $pos_{i,j}$ . The sum of the integrals over both

filters is zero which reflects the volume preservation condition.

$$\sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} neg_{i,j} + \sum_{i=-\infty}^{\infty} \sum_{j=-\infty}^{\infty} pos_{i,j} = 0.$$

Both filters are displayed in one image in Figure 2. The inner part begins in the inflexion points of the curve and it is the *neg* filter. The rest is the outer part i.e., the *pos* filter. The discrete filters result from sampling a continuous function that has the form

$$neg_{x,y} = \sin(x^2 + y^2) - 0.53, \quad \sqrt{x^2 + y^2} > 1.2,$$

$$pos_{x,y} = \begin{cases} \sin(x^2 + y^2) - 0.53 & 1.63 < \sqrt{x^2 + y^2} \leq 1.2, \\ 0 & otherwise \end{cases}$$

When the dragged object penetrates the sand we first apply the circular negative filter. Based on the depth of the impact, the filter values are shifted by a constant coefficient and the sum of the values from this filter and the terrain is calculated. The result of this operation is one number which is the amount of the material that the sphere pushes out. At the same time, the negative values are added to the terrain that removes the material.

The removed material is distributed around the object according to the values in the positive filter in the second pass. The second pass is applied in the same manner as the first one i.e., applying the filter to the area around the object. The only difference is that the values from the filter are multiplied by the proportional part of the material detected in the first pass. In this way only the material detected in the first step is removed in the second one and that is why we need the two step algorithm. The requirement of the material reallocation is essential. The result of this operation is displayed in an example in the upper image of the Figure 3

### 3.2 Sand erosion

Sand seeks gravitational equilibrium. Any particle that is placed in a higher position tends to fall down and find the lowest possible point. This process corresponds to the thermal weathering and has been described by Musgrave et al [10] and later used by various authors [2, 8, 13]. The result of application of this technique is shown in the lower image of the Figure 3. We will briefly describe the simulation algorithm here.

Each point of the height field is processed. The actual height is compared with its eight neighbors and, if it exceeds at least one of them, the corresponding part of sand is removed down. Material is transported only to the neighboring elements so several passes are required to reach equilibrium.

Let's denote the height of the processed vertex by  $m$  and the height of the neighbors is  $m_i$ ,  $i = 1, 2, \dots, 8$ .

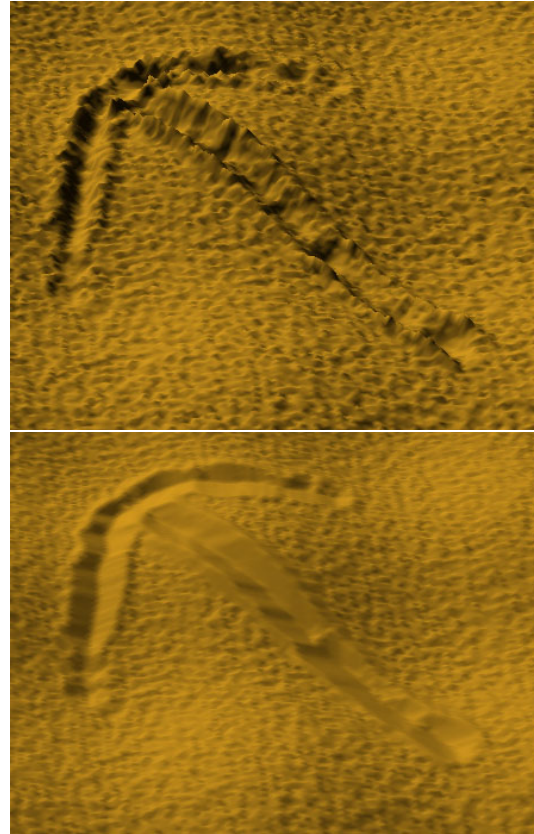


Figure 3: Example of the application of the positive and negative filters (up) and the same scene after erosion

There are two possible cases:  $m < m_i; \forall i = 1, 2, \dots, 8$  and the opposite one.

The first case corresponds to a vertex being a local minimum so there is no material that can be removed from it. In the second case at least one vertex lies lower than the actually processed vertex and the corresponding part of material is removed down. To get the correct distribution of the transposed sand, we first calculate the amount of material that can be reallocated. It is then redistributed according to the height of its neighbors. The vertices that are located lower than the others will receive a higher portion of the material. The total amount of the sand that is relocated is denoted by  $\Delta m$  and the redistribution is described by

$$\Delta m_i = \Delta m \frac{m_i}{sum}.$$

In this equation  $\Delta m_i$  is the sand that is moved to the  $i$ -th lower located vertex,  $sum$  is the sum of all differences to the lower located vertices, and  $m_i$  is the actual difference to the  $i$ -th neighbor.

There is an additional condition of the so called talus angle [10]. This condition postulates that sand does not move to the direction of a lower located neighboring vertex if the slope is smaller than the talus angle. This phenomenon reflects the inner tension of the granular material and roughly says that no erosion goes to infin-

ity. Depending on material, equilibrium is reached at a faster or slower pace. Especially in the case of sand, the talus angle is found very quick, as everyone who has visited the beach has probably experienced. When digging a hole in the sand, it is filled very fast but only the hole boundary is softened. This is shown in the Figure 4, where a scene was set up in such a way to demonstrate the phenomenon.

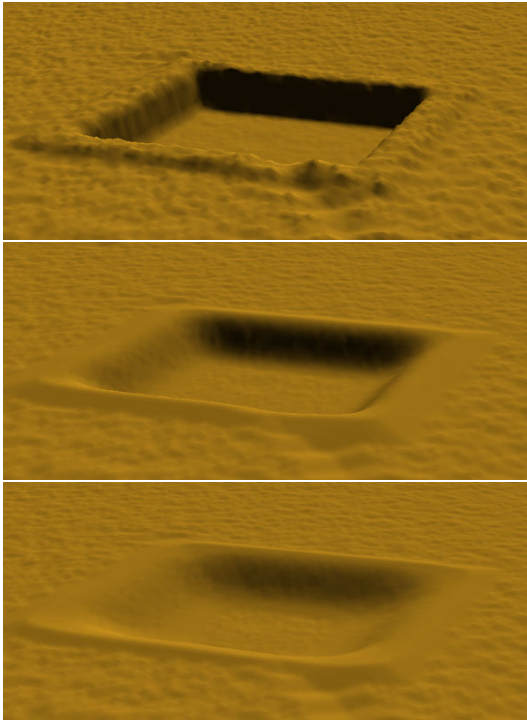


Figure 4: Sand finds the talus angle fast

Setting the value of the talus angle is essential for the speed of erosion. Erosion weakens exponentially and it does not make any sense to keep the angle close to zero. A small talus angle only keeps the algorithm running without a significant visual effect. On the other hand high talus angle values causes the granular material to look very rough. We have experienced a reasonable value of the talus angle to be around  $30^\circ$ .

#### 4 TOUCHING THE SAND

The fundamentally new approach in this paper is using haptic devices to interact with a material that is a subject of metamorphosis at the same time.

Sand is a special case of a granular material and is an extremely complicated subject that is studied in physics. This is primarily because the forces inside the material are not well-known yet. Granular material can be considered as a tight particle system. Each particle has its stickiness, surface area, mass, and shape. As an external force is applied the particles on the surface rotate and move and the forces are transferred inside the volume. This presents a vast amount of forces that are distributed inside the granular material. Obviously, the

force distribution depends heavily on the size of the particles and their stickiness (humidity is an important factor here). The above described qualitative model can be substituted by a quantitative one using reasonable simplifications.

#### 4.1 Forces

We propose a model of sand interaction to a force under the following conditions [12]. The penetrating object is a sphere. It has the major advantage that its interaction area is the same regardless of the direction the sphere moves. This allows us to drop-off area-to-sand calculations and this increases the speed of simulation. Sand is homogenous in our calculations. There are no layers with higher humidity, there are no crunchy shells on the surface, no cracks, nor bubbles or stones. Based on these assumptions we set up a set of equations that describe the force feedback of the sand to the penetrating object. The forces are then sent to the input of the haptic device which provides the corresponding feedback.

The resulting force has two important components. The vertical *repulse force* and the horizontal *friction* as can be seen in Figure 5.

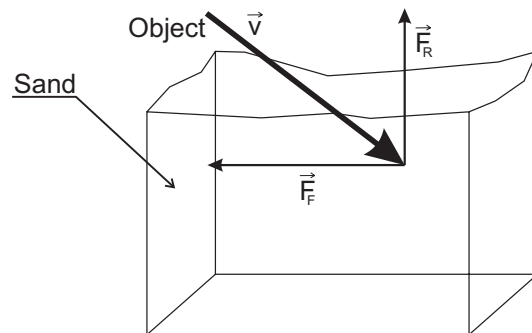


Figure 5: An object entering the sand is repulsed by the vertical force and dragged by friction

The *repulse force* results from the increasing tension with the depth of the sand. The repulse force has the opposite direction to the normal vector to the surface, close to the surface, and becomes vertical with increasing depth. The repulse function has the form

$$\vec{F}_r = -k_s e^{k_c d} - 1 \quad (1)$$

where  $k_s$ ,  $k_c$  are the material dependent constants and  $d$  is the depth of the layer of sand. This force has always vertical, or nearly vertical direction. We use the strictly vertical direction in our simulations, since the sand erodes quickly to the talus angle, as described in Section 3. The angle is small so the normal vector to the surface is nearly vertical. The difference is impossible to feel.

The second force is the inner *friction* of the sand that has only the vertical component. It is also a function of depth but it is higher with the velocity. The equation of friction is

$$\vec{F}_f = -k\vec{F}|\vec{F}_r|. \quad (2)$$



The force  $\vec{F}$  depends on the friction area, the velocity  $\vec{v}$ , and the size of the repulse force  $\vec{F}_r$ . The friction area is constant so  $\vec{F} = k_a \vec{v}$ . We can accumulate the constants  $k_f = k k_a$  obtaining

$$\vec{F}_f = -k_f \vec{v} |\vec{F}_r|. \quad (3)$$

The three constants depend on the implementation and the relative sizes of the other objects in the scene. Having the total sand area of size  $\langle -1, 1 \rangle^2$  and the sphere radius equal to 0.01, the  $k_s = 10^3$ ,  $k_c = 1/2$  and  $k_f = 10$ .

Figure 6 demonstrates the use of force on one stroke. The stroke starts with a strong vertical force that weakens. This causes more material to be removed at the beginning. The sand after it is touched by the sphere immediately erodes-out the differences.

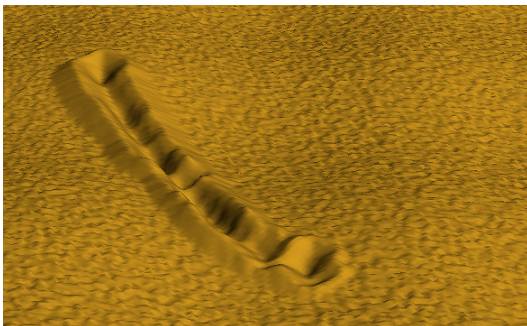


Figure 6: An example of the stroke with varying force applied. The stroke starts with a stronger force and ends with a smaller one as can be seen by the amount of the removed material

## 4.2 Haptic Devices



Figure 7: The 6DOF Phantom (left) and the 3DOF-Omega haptic device

Our sand drawing application currently works with the 3-DOF Omega(tm) haptic device and 6-DOF Phantom(tm) Desktop haptic device (Figure7), both provided by The Force Dimension(tm). These two haptic devices provide an affordable desktop solution that is suitable for our application. The Omega(tm) haptic device is connected to a PC by the USB port and provides resolution of 0.009mm and maximum force feedback

of 12N. The Phantom(tm) is connected via the parallel port and provides 0.03mm resolution, but goes up to 20N and has six degrees of freedom. Both devices provide full gravity compensation.



Figure 8: Scene setup

## 5 IMPLEMENTATION

The entire system is implemented in C++. We use OpenGL(r) for visualization and the CHAI 3D open-source haptic rendering library [4] for interaction. We have chosen the CHAI 3D library because of its transparent support for several haptic devices including the Omega(tm) family, the Delta(tm), and the Phantom(tm) devices. This library is provided in source code and is easily extendible for other haptic devices.

The sand drawing application consists of two major parts; the graphics (visualization) part and the haptic interactive device. Both parts work together to imitate drawing on sand with haptic feedback.

In each haptic rendering loop, we read the actual position of the haptic device in 3D space. We transform it as a position of the virtual device and if it is inside the sand surface we compute the responding forces. These forces are applied to the sand surface and it is indicated that the area should be eroded. The erosion step, that is actually slow, takes place outside the main haptic loop.

The update rates of these two loops are very different since a visual rendering needs a much slower refresh rate compared to a haptic update rate. While a 60Hz refresh rate is standard for visual rendering, the haptic rendering requires at least a 1kHz update rate to provide smooth haptic feedback to a human user. In our application, we use a high frequency timer for the haptic feedback so that the haptic rendering loop takes place in every millisecond. The commands in the haptic loop are not buffered. Once entered the callback application does not execute the same routine until it is terminated. If there is some piece of code that takes a long time, the newly generated position can be much farther. The haptic loop is no reentrant. It is important to keep all the complicated calculations out of the haptic loop and use it just for setting some variables. Actually, synchronization of the visual and the haptic loops presents an interesting problem in general.

Slowing down the application while inside the callback of the haptic loop is especially difficult when changing the depth of the virtual pointer. The force in a shallow depth is small and when deeper it is much higher. According to equation (1) the feedback is an exponential function of depth. Wrongly implemented haptic feedback can cause jumps and sudden abrupt forces. A very strong vertical force on a slow feedback could move the virtual pointer too deep into the granulate material. The haptic device produces a very strong back force in such cases.

There are not major issues in the rendering loop. One of the important parts, from the viewpoint of speed, is to update continuously only the areas that are really changed. To achieve this we divide the regular height field area into a set of 10x10 OpenGL display lists. Once the user makes a stroke, only the affected display lists are updated. Once the area is eroded (all the angles in the simulated area are smaller than the talus angle) and no material is moved, the display lists are not updated anymore. Technically this means that the system is slowest when there are many areas eroded (touched) at the same time. This is technically difficult, because the erosion reaches equilibrium fast.

## 6 CLOSING REMARKS

The system provides interactive feedback for height-fields up to 1000x1000 vertices (2MΔ). The system runs at 60fps on a Dell Precision M70 laptop running at 1.6GHz with the NVIDIA Quadro FX Go 1440 graphics card.

The haptic feedback does provide a higher level of fidelity than using only visual feedback. This can be easily tested by disabling the force feedback on the haptic device and manipulating the sand directly by mouse. We have tried a mouse but found the haptic feedback gives much better results.

The sequence of images in Figures 9-12 shows the successive modeling of a granular surface. The images are frames from the accompanying video. These images show the result of successive editing of granular material using our system.

We have demonstrated that implementation of interactive sand sculpting system with haptic feedback is possible and leads to better results than just visual feedback. We believe that future applications of our approach could be as a plug-in for some professional systems such as Maya(r), or 3D Studio MAX(r). We believe it would be much better to model complex scenes by simple haptic feedback and the user could then test complicated animation sequences in this way as well.

There are still many open areas and future work should focus on the following unsolved problems:

- dragging arbitrary shapes and correct repulse calculation based on the actual intersection area,



Figure 9: Sequence of images demonstrating successive editing of a granular material (image 1)

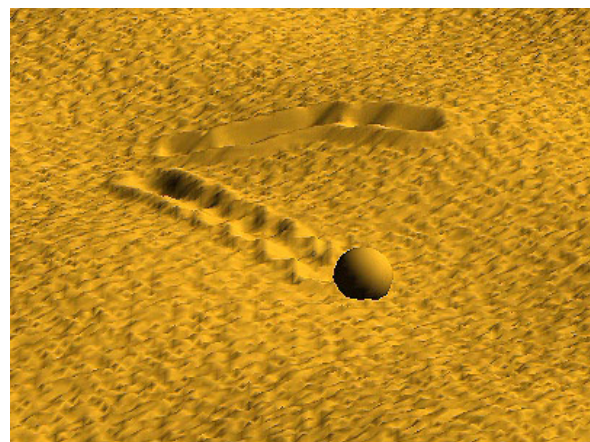


Figure 10: image 2

- correct material manipulation,
- including non-homogenous surfaces, shells, stones, wet sand, and
- including sand in the air (for example as free particles as done in the work of Onoue and Nishita [8]) and

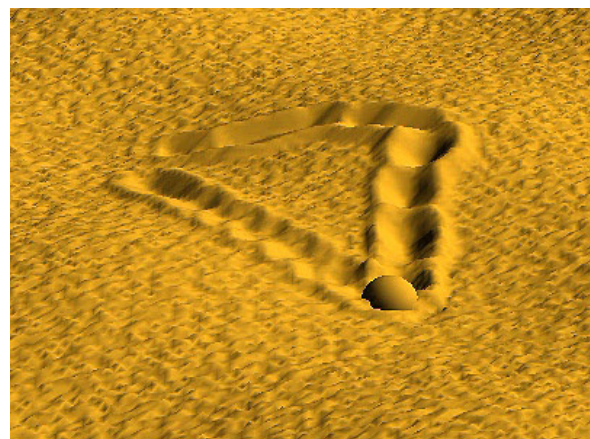


Figure 11: image 3





Figure 12: image 4

- improving control over the haptic device and the rendering loop.

There is one last interesting observation. The haptic feedback is incomparably better than just a visual feedback as we tested when just using a mouse instead of the Phantom(tm) device. On the other hand, using the haptic devices intensively, for more than four hours in the case of the final tuning the application, causes *very strong pain* in the arm next day. This is certainly an aspect that should be considered. The forces that a user has to support are up to 3 N, which means manipulating 3kg of material with one hand. Doing this for a long period of time is certainly good exercise for the user.

## REFERENCES

- [1] B. Beneš and R. Forsbach. Layered Data Structure for Visual Simulation of Terrain Erosion. In *SCCG '01: Proceedings of the 17th Spring conference on Computer graphics*, volume 25(4), pages 80–86. IEEE Computer Society, 2001.
- [2] B. Beneš and R. Forsbach. Visual simulation of hydraulic erosion. *Journal of WSCG*, 10(1):79–86, 2002.
- [3] N. Chiba, K. Muraoka, and K. Fujita. An erosion model based on velocity fields for the visual simulation of mountain scenery. *The Journal of Visualization and Computer Animation*, 9:185–194, 1998.
- [4] F. Conti, F. Barbagli, R. Balaniuk, M. Halg, C. Lu, and D. Morris. The CHAI libraries. In *Proceedings of Eurohaptics 2003*, pages 496–500, Dublin, IE, July 6–9 2003.
- [5] J. Dorsey, A. Edelman, H. W. Jensen, and H. K Pedersen. Modeling and Rendering of Weathered Stone. In *Proceedings of SIGGRAPH '99*, volume 25(4) of *Computer Graphics Proceedings, Annual Conference Series*, pages 225–234. ACM, ACM Press / ACM SIGGRAPH, 1999.
- [6] T. Ito, T. Fujimoto, K. Muraoka, and N. Chiba. Modeling rocky scenery taking into account joints. In *Computer Graphics International*, pages 244–247, 2003.
- [7] O. Koichi and T. Nishita. A Method for Modeling and Rendering Dunes with Wind-ripples. In *Proceedings of Pacific Graphics '00*, pages 427–428, 2000.
- [8] O. Koichi and Tomoyuki. Nishita. Virtual sandbox. In *Proceedings of Pacific Graphics '03*, pages 252–260. IEEE Computer Society, 2003.
- [9] X. Li and M. Moshell. Modeling Soil: Realtime Dynamic Models for Soil Slippage and Manipulation. In *Proceedings of SIGGRAPH '93*, volume 27(4) of *Annual Conference Series*, pages 361–368, 1993.
- [10] F.K. Musgrave, and C.E. Kolb, and R.S. Mace. The Synthesis and Rendering of Eroded Fractal Terrains. In *Proceedings of Siggraph '89*, volume 23(3) of *Annual Conference Series*, pages 44–50, 1989.
- [11] B. Neidhold, and M. Wacker, and O. Deussen Interactive physically based Fluid and Erosion Simulation. In *Proceedings of Eurographics Workshop on Natural Phenomena*, volume 1, pages 25–32, 2005.
- [12] G.N. Smith and G.N. Ian Smith. *Elements of Soil Mechanics*. Blackwell Science Profesional, 1998.
- [13] R. W. Sumner, J. F. O'Brien, and J. K. Hodgins. Animating Sand, Mud, and Snow. *Computer Graphics Forum*, 18(1):17–26, 1999.
- [14] Y .Zhu and R. Bridson. Animating sand as a fluid. *ACM Trans. Graph.*, 24(3):965–972, 2005.

