report on
Research Proficiency Examination

SAND CANVAS
Simulation and rendering of Sand Mixture as media of performance arts

by
NAFEES AHMED
nuahmed@cs.stonybrook.edu

VISUAL ANALYTICS AND IMAGING LAB
DEPARTMENT OF COMPUTER SCIENCE
STONY BROOK UNIVERSITY
Abstract Sand animation is a performance art technique in which an artist tells stories by creating animated images with sand. Though being a newer form of art technique, its popularity has expanded at a great rate during the past decade. With inspiration from this art medium, we intend to develop a digital interface that would allow an artist to produce an artful performance over virtual sand. Among many other important aspects of creating such digitized system, we identify and concentrate upon two of the major challenges; First, producing physically accurate and appropriate simulation of a huge number of sand particles with the time constraint of interactive interface; Second, rendering simulation data in a correctly and visually appealing manner. In this work, we study different possible methods of solving these two and judge their adaptability to our problem comparing their strengths, weaknesses and performance. At the same time, we explore other possible features that can enhance the experience of the artist in this virtual art environment, e.g. introduction of multi-colored sand.

1. Motivation

Through ages, mankind has always found ways to express its ideas, experiences, emotions and fantasies through many different artistic media. Ages and technologies have changed the format, but the desire and expressiveness of these thoughts have surpassed time. Paintings, sketches, sculptures, movies, theatres, graffiti and so on – all these different formats have their own unique abilities to convey the thoughts of an artist to the audience. The power of expression of these formats and their variety is only limited by the imagination of the artists, which inevitably, never falls short to amaze us with new things every now and then. One important branch of artistic media is known as Performance Art (1), where the artist tries to unveil a story by performing live in front of an audience. In past few decades, a new form of performance art has emerged into world canvas, where an artist utilizes very fine grained sand to tell a story by artfully manipulating them on a flat horizontal surface, using bare hand as the only tool. Because of being a new performance art technique, there is only a handful of performing sand artists around the world. Yet, the masterful works of artists like Ferenc Cakó (2), Kseniya Simonova (3), Ilana Yahav (4), Su Dabao (5) have taken the world by amazement and the appreciation has spread all around the world quite quickly. These artists utilize a powerful and expressive vocabulary of physical interactions between their hands and small granules of the homogenous sand to unfold a narrative through a progression of visual images produced with a seamless stream of physical gestures. In contrast with sketches or paintings,
which are produced with discrete pen or brush strokes, sand animation leverages the delicate structure of the artist’s whole hand (often both hands). These hand gestures are easy to learn, fast to perform, and economical to correct, which makes this medium suitable for exploration and brainstorming in addition to storytelling through live performance.

With the inspiration of this simple yet powerfully expressive performance media, our intention is to create a digitized version where an artist would manipulate the virtual sand with both hands. The justification of bringing this media into digital world has several aspects. First, real life sand art performance spaces are difficult to set up and maintain, which prevents many novices from getting started. Second, the digitized version has the flexibility of adding features like undo, copy, macros, record and playback which adds to the comfort of producing newer performances in faster and easier manner. Third, even though we take inspiration from the original media, the digitized version lets us explore newer features that can expand the expression power of the media as whole, i.e. rather than using homogenous sand, multicolored sand can be introduced for a whole new world of artistic possibilities. Fourth, increased availability of multi-touch display surfaces have spurred creation of many touch driven applications, one of which can be sand animation. Even though most of the multi-touch toolkits
fail to capture all the richness in human gestures, there are some devices e.g. Microsoft Surface (6) that can provide rich vision based input to serve our purpose. We would like to refer to such digital adaptation as SandCanvas.

2. WHAT IS SANDCANVAS?

The term SandCanvas (7), as introduced by Habib et al., refers to a multi-touch based digital canvas that allows one to produce sand animation. The device produced by them is the very first attempt to bring sand animation to digital environment. They introduced the concept, provided an analytical taxonomy of hand gestures that produces sand art and tried to produce similar facilities in the digital environment while providing additional features that automatically comes in any digital media. Before we discuss the functional performance and also spaces of improvement in their work, we take a look at the generic framework of such a system.

![Figure 2 Framework of SandCanvas](image)

Figure 2 shows the major outline. The system, overall has four major functional components,
1. Provision of input from user using both hands in true multi-touch fashion.
2. Handle gestures and interpret them to useful quantities to be fed into later phases.
3. With modified state and user interaction, do physical simulation of sand.
4. Render the result onto the screen.

**User Input**

A virtual sand art system requires a device that can capture human hand motion in detail and articulate manner. The interaction can vary from drawing shapes and outlines with tip of any finger to using the whole palm to wipe out the canvas in a smooth fashion. Adaptation of this wide range of human hand interaction can be tricky, since most of the commercially available multi-touch interfaces tend to restrict tracking of human touch to finger tips only. The common technologies (8) (9) involved are Capacitive Sensing, Resistive Sensing, Optical and Sound-Wave based. Capacitive, resistive and Sound-wave based techniques vary in different levels of picture clarity, touch sensitivity and durability, whereas limiting the touch facilities to only at fingertip level. Whereas computer vision based optical techniques provide the additional advantage of actually tracking deformable shapes of the human hand completely. Example of one such device is (6) and in this work (7), they also took advantage of a similar technique.

**Interaction**

The raw input of user hand movement needs to be analyzed and categorized to drive the interaction with virtual sand media. The work by Habib et al. analyzed and identified the basic motions and techniques that are required for a sand animation tool. These are classified into two broad categories,

1. Pouring Techniques: This handles addition of sand into the canvas.
2. Manipulation Techniques: This can be in the form of *Fingertip drawing, Finger Carving, Palm Rubbing, Hand Sweeping*, etc.

**Physical simulation of Sand**

With input through different gestures, we have forces acting upon the virtual sand media that should deform and behave in a way the real art sand behaves on the canvas to provide the artist the similar feel of creative process. There has been a great deal of effort in past few decades to create realistic simulation of sand in literature of computer graphics, but due
to the inherent complexity in behavior of granular materials, capturing its physical properties fully is computationally expensive, hence hard to adapt in real-time applications. As an early adaptation of sand canvas, this implementation (7) concentrates on a simplified yet intuitive simulation model based on height field approach as proposed by Sumner et al (10). Models based on height field provides a scalable solution that can run in real-time providing support for imprint of human hand on the virtual sand and also can account for corrosion. But, due to the limited expressiveness, this model fails to adapt momentum and other dynamic behavior of sand physics that provides the natural appearance and interaction behavior. Also, addition of newer features like producing animation with non-homogenous sand media is considered unadoptable according to present implementations.

This invites for a trivial opportunity of improvement. In this report, we identify and compare presently available methods of simulation of granular material and try to understand the ways of adapting and extending these methods according to our needs.

Rendering

At each time-step, the result of interaction is interactively rendered onto the screen, which effectively produces the final art work. Rendering of the sand implies the visual representation of the current state of virtual sand and the algorithm employed is largely motivated by the data-structure that represents sand and also the application in hand. Since, data representation and manipulation is mostly motivated by the kind of physical simulation algorithm employed, the methods of rendering is closely coupled with the simulation utilized. This implementation relied upon texturing to produce sand art from a height field representation, but since the simulation lacks computation for momentum and other physical phenomenon, it doesn’t extend the rendering facilities to cover those incidents. Inclusion of support for non-homogenous sand will also call for more advanced techniques in rendering than just using simplified textures.

With this view of the sand canvas framework and the present state of its implementation, in this report we concentrate on the following aspects of improvement,

1. A physical simulation model that is expressive enough to capture art-sand behavior more realistically.
2. To enable the artists a bigger canvas of creative space, allow non-homogenous sand to be simulated and rendered. Non-homogeneity can come from sand particles of different shape, size, texture and color.

The rest of the report is structured as follows, in Section 3, we discuss in detail the different algorithms of simulation of granular materials, compare them in terms of features, performance and adoptability. In Section 4, we discuss about different possible approaches taken for rendering simulated data and try to find the one that suits our requirement. Section 5 discusses choices made so far about the present state of work in progress.

3. **Simulation**

3.1 **How Sand Behaves?**

The art-sand, whose behavior we intend to simulate, falls into a more generic category known as granular materials. Granular materials are simple; they are large conglomerations of discrete macroscopic particles. If they are non-cohesive, then the forces between them are only repulsive so that the shape of the material is determined by external boundaries and gravity. If the grains are dry, any interstitial fluid, such as air, can often be neglected in determining many, but not all, of the flow and static properties of the system. Yet, despite this seeming simplicity, a granular material behaves differently from any of the other familiar forms of matter – solids, liquids or gases. In their work titled “Granular solids, liquids and gases” (11), Heinrich and Sidney, discussed in detail, the special behavior shown by granular materials in different situations. According to them, there are two particularly important aspects that contribute to the unique properties of granular materials: ordinary temperature plays no role, and the interactions between grains are dissipative because of static friction and the inelasticity of collisions. With these two major governing principals, granular materials show unique behavior at different ‘states’ and in this section, we try to summarize their observations and get a

![Figure 3](image-url)
guideline to what is required to simulate a canvas full of sand-art.

- When the granular material is held in a tall cylindrical container, such as a grain elevator or silo, no height-dependent pressure head occurs as it does with a normal fluid: the pressure at the base of the container does not increase indefinitely as the height of the material inside it is increased. Instead, for a sufficiently tall column, the pressure reaches a maximum value independent of the height.
- Sand at rest holds its surface through friction. Hence, unlike a fluid, if an object of higher density than sand is placed onto it, the object will not submerge.
- A sand pile will stay at rest until the surface slope reaches an angle lower than the angle of repose. Increasing the slope slightly above that will start an avalanche. Important thing to note is that the flow is only restricted to the surface granules, rather than impacting all the particles in the whole sand pile Figure 3.
- Flowing sand appears to behave like a fluid, but due to internal frictions between particles, they don’t directly follow fluid equations.
- Quick sweep of sand creates a splash and due to non-cohesion of particles, some of the granules float in air for short time before coming back to surface.

In case of sand art, the sand that is commonly used is very fine granuled and uniform. Hence, the simulation algorithm we plan to use should be able to simulate properties of sand with a scalability of handling this many particles within the give time constraint. Now we move our focus towards the discussion of methods adapted by graphics research for handling sand like granular materials. We can divide all the techniques into three broad categories,

- Discrete particle simulation
- Height Field base simulation
- Particle systems

3.2 DISCRETE PARTICLE SIMULATION
The most straightforward and detailed approach towards simulating granular materials is to consider each particle as individual unit and model all the forces as they continually collide with each other. Most of the techniques discussed in this section have a common approach; consider a single granule of sand as the unit of simulation, with a defined shape (generally a sphere), then
define the governing equation that would compute the cohesive and repulsive forces based upon their velocities and position. With all the forces computed, move the particles according to Newtonian physics at each time step. And in the end, provide handles for specific conditions to provide stability to the material as a whole.

Use of particles in simulation of natural phenomena has a long history in computer graphics. Early works like (12) developed techniques to animate and render irregular objects with particle systems. However, it was the introduction of particle systems with pairwise interactions that made it possible to simulate fluids and deformable objects. Miller and Pearce introduced the concept “Globular Dynamics” (13) where a connected particle system is utilized to simulate viscous fluids. Even though their model does not try to model granular materials directly, their approach towards particle interaction provided insight into a feasible simulation of stable granular media. They defined “Globule” to be the simulation unit in the particle system that is represented by a sphere. The globules as a whole represent the media and interactions between is computed as “soft” collision rather than stacking of marbles or billiard balls. “Soft Collision” means the force acting between two globules is a smoothly varying function of the distance between the two centers. The force acting on a globule is computed as,

\[
 f_p = \begin{cases} 
 p^\Delta \left[ S_r \left( \frac{b_1}{D^m} - \frac{b_2}{D^n} \right) - S_d \frac{(\vec{V}\Delta \cdot \vec{P}\Delta)}{D^2} \right] & \text{if } i \neq p \\
 0 & \text{if } i = p 
\end{cases}
\]

Where, \( \vec{P}\Delta \) is the vector formed by the positional difference of the two globules, \( \vec{V}\Delta \) is the difference in their velocities, \( D \) is the distance between them and \( m \) and \( n \) are constants which define the variation of forces with distance. The two scaling factor \( s_r \) (repulsion/attraction) and \( s_d \) (drag) attenuate the inter globule force based on distance. These factors describe a radius of influence for globules. A detailed explanation of these two factors is given in (13). Now, their generic mathematical model can define many different physical media behavior by putting right values of repulsion and attraction function. In case of granular materials like powder and sand, there is no acting force until there is contact and then a sudden continuous rise to illustrate the collision. In case of solids, when the distance between two globules is less than a threshold, there is a negative force (attraction) to keep them together (Figure 4). Figure 5 shows the simulation result of this model.
Because of the smoothly varying force function, the system converges to a stable state much easily than a hard decision system. Due to the adaptability of particle systems into simulation of physical materials and phenomena, similar concepts have been utilized in many different forms in last two decades. (14), (15) used particle systems for generic deformable materials. (16), (17) adapted Smoothed Particle Hydrodynamics (18) to compute the motion of deformable substances and liquids. Similar to SPH, Moving Particle Semi-Implicit (MPS) was also developed (19) along with its application to in-compressible fluid simulation in graphics. Although being effective for fluids and deformable objects, these techniques didn't directly incorporate the idea of simulating granular materials through particle physics of each individual grains. Adapting particle system based fluid simulations for granular materials have been a strong research topic of recent times and is discussed in details on section 3.4.

Bell et al. in their work “Particle based simulation of granular materials” (20) utilized the concept “Molecular Dynamics” that takes its motivation from Miller and Pearce’s work on “Globular
Dynamics” and extends to provide a usable solution of granular material simulation based on true particle level physics. They classified the particle based simulations into two distinct flavors. Event-Driven(ED), or hard-sphere algorithm, are based on the calculation of changes from distinct collisions between particles (typically spheres or polyhedral). The efficiency and utility of ED methods rely on the assumptions of binary collisions, short contact durations, and relatively long intervals between consecutive events. Hence such methods are well suited to simulate rapid flows, whose behavior resembles that of a gas. However in many cases of interest some or all of these assumptions are violated. In systems with strong dissipation the time between collisions may vanish, resulting in so-called inelastic collapse (21). Likewise, ED methods are inappropriate for systems with persistent contacts and frequent collisions such as heaps, hoppers, and shaken materials.

The other approach, which they identify as “Molecular Dynamics” or MD, also simulates particles discretely. But unlike ED methods, they allow small overlap between particles to simulate the elastic deformation of material elements. This overlap is then used along with other physical quantities to determine an appropriate contact force. This distinction allows MD methods efficiently model systems with persistent contacts and frequent collisions.

![MD Soft collision model](image)

*Figure 6 MD Soft collision model. (20)*

Let $\mathbf{x}_1$ and $\mathbf{x}_2$ be the centers of particles with radius $R_1$ and $R_2$ respectively. The overlap $\xi$ of the particles is defined as,
\[ \xi = \max(0, R_1 + R_2 - |x_2 - x_1|) \]

The normalized line of centers,
\[ \vec{N} = \frac{x_2 - x_1}{|x_2 - x_1|} \]

To compute the normal contact force, they restrict the force rules to the form,
\[ \vec{F}_n = f_n \vec{N} \]

Where,
\[ f_n + k_d \xi + k_r \xi = 0 \]

With viscous damping coefficient \( k_d \) and elastic restoration coefficient \( k_r \). \( k_r \) controls particle stiffness and \( k_d \), dissipation during collisions. In the simplest case setting \( \alpha = 0, \beta = 1 \) gives a damped linear spring force,
\[ f_n + k_d \dot{\xi} + k_r \xi = 0 \]

Here, \( k_d \) and \( k_r \) can be chosen to match experimental data, or manually tuned to produce desired response. For detailed discussion about values of \( \alpha \) and \( \beta \), refer to (20).

Normal forces between particles account for their collision and repulsion. But to properly simulate the behavior of sand at rest, an important force to model is the shear force or friction. The basic physical approach towards modeling friction is to consider it as a force opposing the tangential velocity,
\[ \vec{F}_t = -k_t \vec{V}_t \]

In this work, they point out that computation of this shear force can only slow movement in the tangent direction, not stop or reverse it. Hence even though this is computationally expensive, it doesn’t provide adequate support for simulating heap formation, angle of repose etc. A common solution is to introduce a virtual spring when particles first touch (22). Another approach purely relies on particle geometry, using non-spherical particles to create roughness. In their final implementation they prefer this method, because in spite of its simplicity, the result it produces is considerably good. Non-spherical particles are often constructed of polygons (23) or spheres connected through springs (23). Due to the geometric structure, the normal contact forces provide the simulation of static friction required for the material. Figure 7 shows example of tetrahedron and cube shapes constructed from spheres.
Since, in this work they model each particle separately and does a close approximation of all contact forces, the resulting behavior of the granular material closely follows the real world. Figure 8 and Figure 9 shows frames from simulation. Table 2 provides insight into simulation scale and running time.
Figure 9: A bulldozer sweeping through sand debris. (20)

Table 1: Timing data of simulation from (20), on a set of 3Ghz class PCs (20)

<table>
<thead>
<tr>
<th>Simulation</th>
<th>Round Particles</th>
<th>Frames</th>
<th>Min. / Frame</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hourglass</td>
<td>109,708</td>
<td>1600</td>
<td>3.18</td>
</tr>
<tr>
<td>1000 Rings</td>
<td>110,000</td>
<td>460</td>
<td>3.73</td>
</tr>
<tr>
<td>Splash</td>
<td>186,892</td>
<td>480</td>
<td>3.41</td>
</tr>
<tr>
<td>Avalanche</td>
<td>294,820</td>
<td>720</td>
<td>26.40</td>
</tr>
<tr>
<td>Bulldozer</td>
<td>310,149</td>
<td>300</td>
<td>17.40</td>
</tr>
</tbody>
</table>

The simulation data unveils two-fold information about this method; first, due to the fact that the interaction between individual particles is considered, a huge range of physical behavior is covered. Second, this kind of perfection comes with a price of extensive computational requirement. According to their implementation, this is still a long way away from real-time implementation. With a view to adapt this method for interactive application, Harada implemented a GPU version of Bell’s method (24). Later, Rungjiratananon utilized the same methods for real-time interaction between sand and water (25). Even though they were able to achieve an interactive rate (~30fps), the material was coarsely grained (16k particles). With the
amount of particles we expect in art-sand, even with highly parallel algorithms and GPU support, this method will not scale to that level.

Another interesting approach (26) towards bringing down the simulation time was introduced by Alduan et al. They pointed that, in simulation of granular materials, there are two types of forces that work on the granules; the computationally expensive internal forces, e.g. collision and less expensive external forces, e.g. gravity. Since, there is a distinct difference in their scalability, for performance reasons, they propose a method where internal forces would be simulated a much lower resolution, and from there higher resolution particles would be generated and simulated with external forces giving the appearance of real media. Figure 10 and Figure 11 show example from their simulation. Even though, they still cannot achieve interactive rate, the concept of decoupling internal and external force computation gives the added advantage of finding a balance between simulation accuracy and computation time.

![Figure 10](image1.jpg)  
*Figure 10: Left: scene with 50000 particles simulated with full internal force computation (all particles are LR guides). Middle: scene with 2500 particles simulated with full internal force computation (all particles are LR guides). Right: scene computed with spatial force decomposition; the internal granular behavior of the 2500 LR particles in the middle figure is interpolated to a high-resolution, and external forces are computed on 50000 HR particles. This method obtains a speed-up of 11x compared to the scene on the left, with comparable behavior. (26)*

![Figure 11](image2.jpg)  
*Figure 11: Three frames of a simulation with over 2 million sand particles. (26)*

Discussion of this method can be summarized by following points,
• Simulation of each individual particle with soft collision can produce simulation results showing sand flow, heap formation, angle of repose and realistic interaction with rigid bodies.
• Normal contact forces are computed through direct collision of the particles. Shear contact forces are simulated passively using non-spherical geometric shape of particles.
• Particles can be heterogeneous.
• Due to high computational cost, this method doesn’t scale well for material with huge number of particles, hence not directly suitable for interactive applications.

3.3 Height Field Based Simulation
Because of computational complexity inherent to simulating granular materials by considering individual particles, there has been other approaches considered that can provided application specific real-time performance. One such approach is, trying to simulate sandy terrain with simple interactions with rigid bodies. The core principal behind such methods is to represent the terrain as a uniform grid of scalar values that represent the height at each of the grid points. Figure 12 shows an example height field grid that can represent any terrain. Depending upon how those parameters are defined for physical simulation and how it is rendered, it can represent many surface types. But here in this discussion, we are going to concentrate on sand terrains only.

![Figure 12](image)

Figure 12: The uniform grid forms a height field that defines the ground surface. Each grid point within the height field represents a vertical column of ground material with the top of the column centered at the grid point. (10)
The most closely related early work is that of Li and Moshell (27). They developed a model of soil that allows interactions between the soil and the blades of digging machinery. Soil spread over a terrain is modeled using a height field. Soil that is pushed in front of a bulldozer’s blade is modeled as discrete chunks (Figure 13). Although they discount several factors that contribute to soil behavior in favor of a more tractable model, their technique is physically based and they arrive at their simulation formulation after a detailed analysis of soil dynamics. As the authors note, actual soil dynamics are complex and their model, therefore, focuses on a specific set of actions that can be performed on the soil, namely the effect of horizontal forces acting on the soil causing displacements and soil slippage.

![Figure 13 Pattern of soil movement ahead of the blade. (27)](image)

With a view to enable more visually correct simulation of surfaces of sand/soil, Chanclou et al. proposed a model (28) based on physical laws. In their approach, they modeled the surface as an elastic sheet that can adapt to interactions of moving rigid bodies and take imprints to produces visual artifacts of sand interaction. Figure 14 shows their approach to the model and Figure 15 is an example result of how moving rigid bodies produces visual imprint on loose soil surface.

![Figure 14 Granular modeling of loose soil: with particle and with a plastic carpet (28)](image)
Even though their work didn’t utilize the concept of height field based simulation directly, their approach toward appearance driven physical simulation opened up some possibilities. Sumner et al. extended (10) the appearance based approach with displacement and erosion, to model the contact mark which is caused by the penetrated objects. Summary of the algorithm produced by them is presented below.

- A terrain is represented by a uniform grid structure with scalar height values at the center of each grid cells. Initially the values are set to match the scene by procedural generation or external data loading.
- The height field gets deformed when some rigid geometric object pushes the ground at contact.
- The collision algorithm determines whether a rigid object has collided with the ground surface. For each column, a ray is cast from the bottom of the column through the vertex at the top. If the ray intersects a rigid object before it hits the vertex, then the rigid object has penetrated the surface and the top of the column is moved down to the intersection point. The computational cost of the ray intersection tests are reduced by
partitioning the polygons of the rigid body models using an axis-aligned bounding box hierarchy (29).

- Using a vertex coloring algorithm, a contour map with the distance from each column that has collided with the object to the closest column that has not collided.

- Ground material from the columns that are in contact with the object is either compressed or distributed to surrounding columns that are not in contact. \( \Delta h = \alpha m \) denotes the amount of material to be distributed where \( \alpha \) denotes the compression ratio and \( m \) represents the total amount displaced. All the material from contact area is distributed to the columns in the first non-contact ring.

- After the distribution is done, a continual erosion process is carried out through the terrain to ensure stability. The erosion algorithm examines the slope between each pair of adjacent columns in the grid; if the slope is greater than the threshold then the ground material is moved from the higher column down the slope to the lower column. Ground material is moved by computing the average difference of height.

- For added visual perfection, particles are generated on contact.

With this framework in hand, an artist can tune the provided parametric variables to simulate behavior of different types of terrain ranging from sand, mud, snow and others. Table 2 shows example of such variable values. Figure 17 shows results from their simulation.
Table 2 Table of parameters for the three ground materials.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Sand</th>
<th>Mud</th>
<th>Snow</th>
</tr>
</thead>
<tbody>
<tr>
<td>liquidity ($\theta_{\text{stop}}$)</td>
<td>0.8</td>
<td>1.1</td>
<td>1.57</td>
</tr>
<tr>
<td>roughness ($\sigma$)</td>
<td>0.2</td>
<td>0.2</td>
<td>0.2</td>
</tr>
<tr>
<td>inside slope ($\theta_{\text{in}}$)</td>
<td>0.8</td>
<td>1.57</td>
<td>1.57</td>
</tr>
<tr>
<td>outside slope ($\theta_{\text{out}}$)</td>
<td>0.436</td>
<td>1.1</td>
<td>1.57</td>
</tr>
<tr>
<td>compression ($\alpha$)</td>
<td>0.3</td>
<td>0.41</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Figure 17 Images from video footage of a human runner stepping in sand and a simulated runner stepping in sand, mud and snow. (10)

With this method of simulation they can visually emulate the interaction of a rigid body imprint on a terrain of material like sand or snow. Their simulation covers for displacement and a basic model of erosion. But they do not account for any sort of momentum based calculation; also the velocity of the rigid object has no impact on the displacement of the sand. Because of this simplicity in modeling sand, their implementation runs in real-time.

The work by Sumner inspired a series of development along this path. Benes and Forsbach (30) proposed a layered data structure for visual simulation of terrain erosion. They use thermal erosion which is originally introduced by Musgrave et al. (31) to calculate the motion of granular material. Their algorithm could capture some more realistic features, such as subsurface erosion in caves and erosion of material consisting of different densities. However, their method is unable to deform the ground surface as it contacts with a solid object. Benes and Roa (32)
significantly improved the algorithm of Onoue and Nishita (33) for simulating wind-ripples. The sand movement is extended by the detection of fixed objects. The sand can only be moved by wind and not a solid object.

Onoue and Nishita produced a modified version of Sumner’s work that takes into count the velocity of the rigid body in contact with the terrain. Figure 18(a) show the overall framework of their work, which still includes all the basic steps from the original work. What differs is their approach towards the implementation.

Figure 18 (a) Framework diagram (b) Rendering of polygon onto the ground surface to help optimize ray-casting collision detection (c) Anti-aliasing on the ray casting method (34)

The most important addition they bring to the framework is the updated distribution mechanism. After a collision is detected and the contact contour is identified, for each of the
column under contact, the material to be distributed, $\Delta h$ is factored into two orthogonal components,

$$\text{Vertical component, } \Delta h_v = \min\left(\left| V_{objz} \right| \Delta t, \Delta h \right)$$

$$\text{horizontal component, } \Delta h_h = \Delta h - \Delta h_v$$

Where, $V_{objz}$ is the velocity of the object in vertical direction and $\Delta t$ is the time interval of a simulation step. The vertical component of displaced material is then distributed according to Sumner’s method using compression factor $\alpha$. The horizontal displaced material is distributed to reflect the motion of the object in contact. Figure 19 shows the impact of a rigid body movement on the height field. For detailed pseudo-code of the algorithm, refer to (34).

![Figure 19: Distributing granular material. The red arrow shows the direction of the object's motion. The black arrows depict the distribution behavior of the granular material. The dotted lines show the height states of columns of the ground granular material before it collides with the motion object. (a) The motion object moves in vertical direction and the displaced material tends to equally distribute to the neighbors. (b) The motion object moves in horizontal direction and the displaced material tends to distribute in the direction of object’s horizontal motion. (34)](image)

To provide optimization to ray-casting based collision detection, they utilize an image space based technique of polygon indexing. Figure 18(b) shows their approach where the polygonal rigid body is rendered orthogonally onto the bottom surface with each polygon having their unique color. That allows any column to index the nearest polygon face that is in contact with the height field in very fast manner. Also, to

![Figure 20: A teapot thrown on mud. (34)](image)
avoid aliasing problem, the texture at the bottom is sampled at twice the resolution, [Figure 18c]. The simulation result of a teapot thrown onto a patch of sand is shown in Figure 20.

The same authors took another approach (35) on improving over Sumner’s work that involved a different and more expressive data-structure for representing sandy terrain. Rather than having a single scalar value representing a unit in height field grid, they represented the terrain with a linked list structure called Height Span Map. Figure 22 shows a cross section of HS Map, which asserts the point that now, ground material is allowed to be discontinuous with the original terrain field and hence allowing further interaction with the rigid object. Figure 21 shows the result of a simulation where a concave rigid object like a bucket is used to carry off a portion of sand from the terrain to produce more realistic interaction.

To summarize the discussion on height field based approach,

- This method is simple enough to be adapted in real-time applications.
- The appearance driven technique can facilitate our cause of producing artist interfaces.
- Good for handling sharp imprints of rigid body contact on otherwise non-moving terrain/canvas.
- Can cover for erosion, momentum with visual correctness.

But,

- The data structure is too restrictive to allow generic physical behavior to be modeled; their empirical
nature prevents them from faithfully simulating high-speed scenarios where significant energy is exchanged among particles and objects.

- A scalar valued grid is only good for homogenous media. Simulating with mixture of sand of different color and properties is non-trivial with the same approach and not yet researched properly.
- All these method utilize uniform grid structure which doesn’t take advantage of localized interaction.

3.4 AS CONTINUUM USING PARTICLE SYSTEM

In section 3.2 we discussed briefly how particle system is being widely adapted to simulate continuous deformable materials. A particle system is essentially a representation of some volume with a set of points. Each point in the particle system stores information about the material at that location. Generally no direct connectivity between points is considered necessary. But the way they interact with each other and external forces, define the physical behavior of the media. Early works (12) utilized particle systems to simulate deformable fuzzy objects. Later, introduction of Smoother Particle Hydrodynamics (SPH) (18) created a series of works for simulating fluids. Desburn and Cani (36) introduced SPH to the computer graphics community in 1996 and later Muller et al. popularized the particle method for solving fluid flows (17). Zhu and Bridson (37) took the continuum approach of simulating sand as a fluid based upon SPH particle system, there by decoupling the resolution of the simulation from the grain size. Lenaerts and Dutre (38) adapted this method to the setting of sand-water interaction. Narain et al. (39) built on this continuum approach allowing non-cohesive behavior, globally coupled frictional handling and efficient two-way interaction with solids. Since much of these works were based upon SPH, we provide a brief introduction to the concept in the following paragraph.

SPH was first created to study fission in rotating stars by Lucy (40) and Gingold and Monaghan. It has since been applied to astrophysical simulations such as star formations, solar systems and supernovae.
Basically, SPH is an interpolation technique to provide a continuous field from the discrete particle definition. Using kernels $W$, continuous properties $A(x)$ can be derived as a weighted sum of the properties $A(x_j)$, defined in all particles $p_j$ with volume $V_j = m_j/\rho_j$:

$$A(x) = \sum_j V_j A(x_j) W(x - x_j, h)$$

The function $W(x - x_j, h)$ is a radially symmetric kernel function with a finite support range (or smoothing length) $h$. $W$ must be even (i.e. $W(\vec{r}, h) = W(-\vec{r}, h)$) and normalized over space:

$$\int_\Omega W(\vec{r}, h) d\vec{r} = 1$$

Where, $\vec{r} = x - x_j$. One such smoothing kernel is the Poly6-kernel proposed by Muller et al. (17).

$$W(r, h) = \begin{cases} 315 \frac{1}{64\pi h^9} (h^2 - r^2)^2 & 0 \leq r < h \\ 0 & \text{otherwise} \end{cases}$$

In SPH the derivatives of a discretized function $A$ can be obtained using the derivatives of the smoothing kernel:

$$\nabla A(x) = \sum_j V_j A(x_j) \nabla W(x - x_j, h)$$

$$\nabla^2 A(x) = \sum_j V_j A(x_j) \nabla^2 W(x - x_j, h)$$

Here, the continuous function can be any property stored within each particle, e.g. density, velocity.

The advantage of using a kernel with a finite support is that the given equation and its derivatives only have to be evaluated for neighboring particles instead of all particles. Which reduces the running time of SPH from $O(N^2)$ to $O(NM)$.

Before we continue discussion about adaptation of fluid solvers into granular material simulation paradigm, a quick overview of the fluid simulation itself; There are currently two main approaches to simulating fully three dimensional water in computer graphics: Eulerian grids and Lagrangian particles. Grid-based methods (recent papers include [Carlson et al. 2004; Goktekin et al. 2004; Losasso et al. 2004]) store velocity, pressure, some sort of indicator of
where the fluid is or isn’t, and any additional variables on a fixed grid. Usually a staggered “MAC” grid is used, allowing simple and stable pressure solves. Particle-based methods, exemplified by Smoother Particle Hydrodynamics, dispense with grids except perhaps for accelerating geometric searches. Instead fluid variables are stored on particles which represent actual chunks of fluid, and the motion of the fluid is achieved simply by moving the particles themselves. The Lagrangian form of the Navier-Stokes equations (sometimes using a compressible formulation with a fictitious equation of state) is used to calculate the interaction forces on the particles. The primary strength of the grid-based methods is the simplicity of the discretization and solution of the incompressibility condition, which forms the core of the fluid solver. Unfortunately, grid-based methods have difficulties with the advection part of the equations. On the other hand, particles trivially handle advection with excellent accuracy—simply by letting them flow through the velocity field using ODE solvers—but have difficulties with pressure and the incompressibility condition. SPH methods in particular cannot tolerate non-uniform particle spacing, which can be difficult to enforce throughout the entire length of a simulation. Zhu and Bridson utilized a semi-Lagrangian method where they adapted PIC (41) and FLIP (42) for simulating incompressible flow in SPH.

With a view to simulate sand using fluids, they make the following assumptions,

- The pressure inside a sand volume is similar to the pressure required to make the velocity field incompressible. This is certainly false in real-life, hence their simulation do not follow a standard property of sand, that is a sand column has constant maximum
pressure irrespective of its height. They claim that even with this, the simulation results can be acceptably good.

- Where the sand flows, the frictional stress is the point on the yield surface that most directly resists the sliding.

They decompose the sand domain in regions moving rigidly and regions of shearing flow. To do this, the frictional stress \( \sigma_f \) and the rigid stress \( \sigma_r \) are computed for each particle using the gradient of displacement in each particle. The frictional stress for particles in regions of shearing flow is given as:

\[
\sigma_f = -\mu_f \frac{D}{\sqrt{1/3|D|}}
\]

Where, \( \mu_f \) is the friction coefficient. The strain rate tensor \( D = (\nabla u + \nabla u^T)/2 \) is evaluated for each particle using SPH:

\[
\nabla u = \sum_j V_j v_j \Delta t \cdot \nabla W(x_j - x_i, h_j).
\]

Particles in regions moving rigidly can be found by testing the rigid stress \( \sigma_r \):

\[
\sigma_r = -\frac{\rho D h^2}{\Delta t}
\]

Against the Mohr-Coulomb conditions, which determines material yielding,

\[
\sqrt{3}\sigma_f < \mu_f \sigma_r + c
\]

Figure 24 Stanford bunny is simulated as water and as sand by Zhu and Bridson’s method. (37)
After this, a search for clusters of particles marked as rigid is made. Two rigid particles belong to the same cluster when at least one path can be constructed between those particles over neighboring particles within a support range. Forces on cluster particles are accumulated to a total force and torque and the particle cluster is then moved as a rigid body. The remaining particles are updated according to the flow condition of the fluid solver. Figure 23 depicts the framework summary.

Toon Lenaerts in his work on “Unified particle simulation and interaction” (43) concentrated on interaction of sand with water to produce a general framework than can handle wet materials when water mixes with sand or other porous substances. Even though they didn’t directly work on media like mixture of sand with multiple properties, their work is a big cue on how to approach such simulation problem. Their work heavily relies on fluid simulations based on SPH and hence they also extended on sand simulation developed by Zhu and Bridson. In this work, they identified one key aspect about simulation of mixed media, simulation using Eulerian approach doesn’t hold in this case, because defining multiple substances in a single grid often requires the same grid resolution for each substances, also during simulation interaction within a single grid has be handled separately. Hence, they opted for Lagrangian approach. In a Lagrangian setting interactions can be dealt with between neighboring particles, which is far more flexible than defining interfaces in a grid.

Figure 25: A Lagrangian simulation setup. Fluids and objects are sampled (43)
The second fact they stress on is that SPH fluid algorithms (17) (36) cannot guarantee incompressible fluids. Using a compressible fluid solver not only results in a loss of volume, but also greatly influences the stability and the dynamics of the particle simulation. Becker and Teschner introduced Tait’s equation in the graphics community to faster react to these density deviations. This effectively avoids small explosions of particles in compressed regions.

![Figure 26: Mixture of sand and water using unified SPH simulations. (38)](image)

Lenaerts and Dutre’s (38) work on mixture media wasn’t intended to be for interactive purpose. The work by Rungjiratananon et al. created a GPU implementation (25) of sand-water interaction using SPH based fluid solver and Molecular Dynamics based sand physics. Due to the limitation of scalability provided by the sand simulator, their work can only run at interactive rate with considerably fewer particles (~20K) than desired. Yet, it provides insight into the possibility of having an interactive version of mixture media simulator where both sand and water or non-homogeneous sand is simulated using particle systems only.

Narain et al. (39) in their work identified the shortcomings of SPH based methods and tried to solve the incompressibility issues. According to them, their work fully resolves the internal pressure and frictional stresses, producing visually noticeable behaviors of granular material, which includes freely dispersing splashes without cohesion, constant rate of flow in an hourglass.

LBM
To summarize the discussion on SPH based granular material simulator, this method can decouple physical simulation from the resolution of sand material, providing optimization in simulation time. The most recent works can ensure physical behavior very close to that of simulation using each individual particle and also readily adopts mixture media due to Lagrangian setup. Right now, there is no interactive implementation that fully solves this problem, but availability of real-time SPH based fluid solvers hint towards possibility of having one.

With success of SPH in simulating granular materials, there can be question about if other approaches of fluid simulation like Lattice Boltzman Model can be applicable here. But until now, to our best of knowledge, there is no known algorithm that does that.

4. RENDERING
At each time step, once the simulation is done, the sand needs to be rendered. Techniques and challenges involved in the visualization of sand is closely coupled with the data structure that represents virtual, hence the simulation technique itself. In this section, we are going to discuss
the techniques adapted by the works presented earlier in this report, find out if those are suitable for the task we have in hand and consider other methods from graphics literature.

4.1 Direct Render of Particles
The simplest of all representation of granular materials is the methods adapted as in (20), where we store positions are properties of each individual particles. With access to positions of all the particles, rendering them can be as straightforward as modeling each of them as very small sized spheres. The roughness of surface can be brought in by considering non-spherical particle shapes or bump maps over each particle. But since their simulation weren’t at interactive rate, they had the opportunity of utilizing the added advantages of offline rendering algorithms. In the real-time GPU implementation (25) of Bell’s method, particles were rendered as solid spheres using point sprites, and the visual quality is enhanced by a screen-space ambient occlusion technique similar to (44). Availability of particle positions simplifies the rendering task a lot, and we can also take advantage of the acceleration from the graphics hardware. But due to the lack of scalability of the simulation itself, this method eventually proves to be off little value.

4.2 Rendering of Height-Field
We move our interest to the other interactive simulation paradigm where sand volume is modeled as a height field. The rectangular grid that identifies the height field produces a polygonal mesh on the top with each scalar value at the grid points denoting the height of the surface. This surface is then textured to provide the right look and feel. To produce the

Figure 29: Example render from particle based simulation by (21)

Figure 30: Real-time render from particle based simulation by (26)
impression of right depth, density and motion, blending and other techniques are utilized. (34) exploited the texture sliding method as introduce in (35).

![Image](image_url)

**Figure 31**: Example of rendering of height field using texture sliding method. (7) (34)

Now, even though this method is quite favorable for real-time application both in terms of simulation and rendering, inclusion of mixture media causes problem in both. Height fields are not expressive enough to adapt to non-homogeneous sand and also, having multi valued grid cells will not directly fit into the current scheme of texturing the polygons directly.

### 4.3 Rendering of Particle System

The most lucrative of all the simulation methods were particle systems, where the sand volume was represented as set of particles rather than each individual particle. And this also allows us to simulate mixture of sand is possibly within a favorable time-scale. Rendering of simulation data like SPH is an interesting research area itself and there has been some considerable effort on tackling this problem.
First, let us understand the problem closely. When we simulate a media using SPH, we are representing a volume containing possible millions of grains with only few particles in the magnitude of 10k only. Even though we have particles and their positions, rendering them directly doesn’t provide us with the right result. To get around this problem, one common approach adapted by the graphics community is polygonization of sand volume. To do this, first they think of a continuous function that represents the volume itself based upon SPH approximation. Then, an ISO surface extraction algorithm is utilized to produce the polygonal version of the volume. Extraction of surface from a volume data is a well-researched problem by itself and in this case the favored method is Marching Cubes (45).

![Color field and iso-surface](image1)
![Marching Cubes mesh](image2)

**Figure 32**: (a) The color field defined by a set of particles in SPH set-up. (b) The surface mesh extracted using Marching Cubes. (43)

After the surface has been extracted, the rest of the job is to provide the right texture. However, a procedural sand texture requires texture coordinates consistent over time which is not trivial. Bargteil et al. (46) , Kwarta et al. (47) and Mihlef et al. (48) proposed methods for texturing fluids and visualizing the flow. The later one is important for sand volumes since they often show sand sliding down.
The method of using MC for rendering SPH data can provide fast enough result for real-time application. But, introduction of mixture media produces an odd situation for the polygons. Figure 34 shows a standard case of having multiple SPH simulated granules together. The blue and red granules have different color and properties. The dotted lines show the surfaces extracted out of them. Since, these surfaces make hard decisions on the boundaries; they do not represent the media on those places truthfully.

Hence, directly adapting the present interactive rendering solutions might not be feasible.

The efforts that wanted to work with sand mixture took a common way around to this problem by eliminating the need of extracting surfaces altogether. One solution to this problem as utilized by Lenaerts (38) and Narain et al. (39), is to up-sample millions of grains from SPH particles and move them separately in response to physical stimuli. Then render them directly as individual particles. There is a two-fold problem to this method; first, ensuring correct physical
behavior for high resolution particles can cause problems. Second, trying to simulate and render such huge amount of particles might not scale well for art-sand. As quoted in (39) “the issue of sampling and rendering millions of grains from a continuum representation of granular material is an interesting research problem in itself, and can further enhance the visual appearance of our key contribution on simulation. Further independent investigation of this problem is valuable.”

Since, the issue of rendering sand mixture from particle system simulation is quite new in graphics research; there hasn’t been enough effort to try different possibilities to find the right way to solve the problem. Apart from trying to adapt to polygon rasterizers, there are other methods that can help render the simulation of sand. Two such possibilities are point based rendering schemes and direct volume rendering. Because of lack of exploration along this track for sand rendering, it will require close consideration and tests to find out their quality, performance and applicability in this issue. We put it on the list as part of our future work.

5. Conclusion

In this report we presented a generic framework for digitized version of sand art. As part of the implementation process, we draw our focus onto two of the important steps – simulation and rendering. First we discussed about methods available as part of graphics research to simulate granular materials. We found particle based method to be most expressive, but not suitable for real-time applications. Height field based approaches perform well within their limit, but they lack the power of extensibility to add newer features. Methods based on particle system have the potential of finding the right balance, but generally the algorithm is hard to adapt directly. In terms of visualization, rendering height-field and particles directly is easily implementable in interactive applications, even though the present texture based approach for height-field cannot directly adapt to sand mixture. For particle systems, there are known good methods based upon iso-surface extraction that work well in interactive setting, but again, they fail to support sand mixture. With a view to support non-homogeneous media, we intend to explore other techniques in rendering and also for simulation, which can produce visually appealing results within the restriction of interactive rates.
REFERENCES


28. Physical models of loose soils dunami all marked by a moving object. **Chanclou, Benoit, Luciani, Annie and Habibi, Arash.** s.l. : Computer Animation, 1996.


