

Shallow Sand Equations: Real-Time Height Field Simulation of Dry Granular Flows

Kuixin Zhu, Xiaowei He, Sheng Li, Hongan Wang and Guoping Wang

Abstract—Granular media is the second-most-manipulated substance on Earth, second only to water. However, simulation of granular media is still challenging due to the complexity of granular materials and the large number of discrete solid particles. As we know, dry granular materials could form a hybrid state between a fluid and a solid, therefore we propose a two-layer model and divide the simulation domain into a dilute layer, where granules can move freely as a fluid, and a dense layer, where granules act more like a solid. Motivated by the shallow water equations, we derive a set of shallow sand equations for modeling dry granular flows by depth-integrating three-dimensional governing equations along its vertical direction. Unlike previous methods for simulating a 2D granular media, our model does not restrict the depth of the granular media to be shallow anymore. To allow efficient fluid-solid interactions, we also present a ray casting algorithm for one-way solid-fluid coupling. Finally, we introduce a particle-tracking method to improve the visual representation. Our method can be efficiently implemented based on a height field and is fully compatible with modern GPUs, therefore allows us to simulate large-scale dry granular flows in real time.

Index Terms—shallow sand equations, dry granular media, real-time simulation

1 INTRODUCTION

Granular media, after water, is the second-most-manipulated substance by man on Earth [1]. Strictly speaking, a granular material is a collection of large number of discrete solid particles with interstices filled with a fluid or gas [2]. It shows distinctive behaviors that manifest itself either like that of a solid or a fluid. For instance, flour can both pack like a solid and flow like a fluid. Researchers in computer graphics have simulated various scenarios of granular flows, such as snow [3], [4], debris [5], [6], [7] and dry/wet sands [8], [9], [10]. It has been shown that the dynamic behaviors of granular flows not only depend on the particle mass, but also on other factors such as particle size, particle shape, water saturation, etc. How to simulate a large-scale granular flow that contains millions or even billions of granules has long been believed to be a tough challenging for numerical simulation.

In this paper, we consider how to simulate dry granular flows efficiently. If the granule size is large enough (e.g., size $d > 100\mu\text{m}$) and the viscosity of the interstitial fluid is small, it is reasonable to neglect all subtle forces such as electrostatic, aerodynamic and capillary forces [11]. Therefore, we can model a dry granular flow by only considering the transfer of momentum and energy dissipation, where the cohesion is simply neglected. Within continuum mechanics, a dry granular flow can either be modeled approximately

as a unilateral incompressible fluid [8] or a Drucker-Prager elastoplastic material [9], both of which achieved realistic simulation results. However, for large-scale scenarios, it is still impractical to apply these models for real-time applications.

Motivated by the success of the shallow water equations used for simulating large-scale fluids, we develop a set of governing equations, which we call *shallow sand equations*, to model the dynamics for dry granular flows whose vertical length scale is much smaller than the horizontal length scale. The term “shallow” is used here with a motivation from the S-H model [12] which applies a shallowness assumption when investigating the idealized problem of a finite mass of material released from rest on a rough inclined plane. Due to the fact that a dry granular material could form a hybrid state between a fluid and a solid, we divide the simulation domain into two layers where one is treated as a shallow fluid while the other as a static solid. The shallow sand equations can then be derived by integrating three-dimensional governing equations along its vertical depth. All components of our method can be efficiently solved on GPU in parallel, achieving real-time simulations for large-scale dry granular flows. We also present a ray casting algorithm for one-way solid-fluid coupling and a particle-tracking method to improve the visual effects. Our method is able to simulate large-scale dry granular flows and fluid-solid interactions in real time, as shown in Figure 1 and 17.

To summarize, the contributions of this paper are

- A set of shallow sand equations that realistically capture complex behaviors of dry granular flows, such as sand pilings with different friction angles.
- A ray casting algorithm for efficient one-way solid-fluid couplings.

- K. Zhu is with the School of Electronic and Computer Engineering, Peking University ShenZhen Graduate School, Peking University.
E-mail: zkx@pku.edu.cn
- X. He is with the Institute of Software, Chinese Academy of Sciences.
E-mail: xiaowei@iscas.ac.cn
- S. Li is with the Peking University.
E-mail: lisheng@pku.edu.cn
- H. Wang is with Institute of Software, Chinese Academy of Sciences.
E-mail: hongan@iscas.ac.cn
- G. Wang is with the Peking University.
E-mail: wgp@pku.edu.cn

Manuscript received X X, XXXX; revised X X, XXXX.



Fig. 1. Butterfly. A butterfly is drawn with a wooden stick, which is then flattened by a long wooden block.

- A particle-based tracking method with adjustable resolution of particles to enrich visual details for dry granular flows.

2 RELATED WORK

Particle-based approaches Due to the particle nature of granular flows, researchers in computer graphics have mainly applied particle methods to model granular materials. Luciani et al. [13] directly modeled interactions between individual particles. Bell et al. [14] presented an efficient discrete element method to simulate granular flows and two-way coupling with direct interactions between rigid-motion-constrained collections of spheres. Rungjiratananon et al. [15] combined SPH and DEM together to model interactions between fluids and granular materials. Lenaerts and Dutré [16] instead proposed a unified smoothed particle hydrodynamics (SPH) framework to simulate both fluids and granular materials. Alduán and Otaduy [17] also modeled granular flows under the SPH framework, but extended to model the force of friction with a predictive-corrective scheme. Besides, Alduán et al. [18] tried to reduce the computational cost with a spatial decomposition of the computation of internal and external forces. Chang et al. [19] proposed a modified Hooke's law to handle the friction between grains. Longmore et al. [20] implemented a discrete element method on GPU which greatly improves the simulation performance. To improve the visual quality, Ihmsen et al. [21] simulated granular flows on coarsely sampled particles first, then embedded high-resolution particles using a pure Lagrangian method. The idea of coupling highly resolved particles with the base simulation motivates us to embed a high resolution of particles at the top of the height field to enrich visual details. To uniformly handle deformable bodies and granular materials, Yan et al. [22] introduced a multiphase fluid SPH framework. He et al. [10] has recently proposed an elastoplastic material model based on projective peridynamics that can be easily extended to simulate both dry and wet sand.

Continuum approaches Although particle-based methods have been widely used in computer graphics, the disadvantages are obvious. Simulating a large-scale scenario containing millions of particles is usually computationally prohibitive for real-time applications. Therefore, other researchers instead use a continuum approach to model the granular media as a continuous material. Zhu and Bridson [23] first implemented a modified Particle-in-Cell method and simulated sand as an incompressible fluid. Narain et al. [8] improved their work by assuming that the dry sand is unilaterally incompressible and solved its

dynamics as a quadratic optimization problem with KKT conditions, which effectively removes the cohesion artifacts associated with incompressibility. To improve the performance, Nkulikiyimfura et al. [24] realized a GPU version of Zhu and Bridson's approach. Klár [9] used the Material Point Method (MPM) equipped with the Drucker-Prager plastic flow model for sand animation, producing appealing behaviors of sand flowing and piling. Daviet and Bertails-Descoubes [25] proposed a semi-implicit discrete scheme for the realistic simulation of large-scale free-flowing granular materials. Gao et al. [26] recently accelerated MPM simulation with GPU optimization. To reduce the computational cost, Yue et al. [27] proposed to hybridize the discrete and continuum approaches to simulate granular materials. In some sense, our method is quite similar to MPM since both methods apply an Eulerian grid together with embedded particles to solve granular flows. However, since MPM uses 3D Eulerian grids to solve the dynamics of granular flows, its performance still cannot satisfy the requirements of real-time applications.

Height-field-based approaches An alternative solution to further improve the simulation efficiency is to use a height field for sand animation. The most widely used theory was developed by Savage and Hutter [12], which is termed as the S-H model. However, the S-H model is only valid when the "downhill" velocity and its variation are much larger than those in the lateral direction [28]. Several improvements were proposed by other researchers under the shallowness assumption [29], [30]. In contrast to the S-H model, the Hungr model is proposed with a strain-related expression for the earth pressure coefficient based on conventional loading and unloading experiments with soil, which is only targeted at simulating rapid landslides [31], [32]. In computer graphics, Li and Moshell [33] applied analytic methods based on soil properties and Newtonian physics to model soil slippage and soil manipulations. Sumner et al. [34] used deformable surface ground to model sand, mud and snow. Onoue and Nishita [35] proposed a non-physical editing method to deform the ground surface for rigid body interactions. Zhu and Yang [36] also tried to accelerating the sand animation by simulating sand as a surface flow. As far as we know, none of these methods are suitable for realistic real-time applications yet, e.g., digital sand painting.

Digital arts For real-time digital arts, performance should always be considered at first priority. Therefore, Kazi et al. [37] optimized the work of Sumner et al. [34] and accelerated the computing by using graphics hardware. Chen and Wong [38], [39] proposed a method for auto

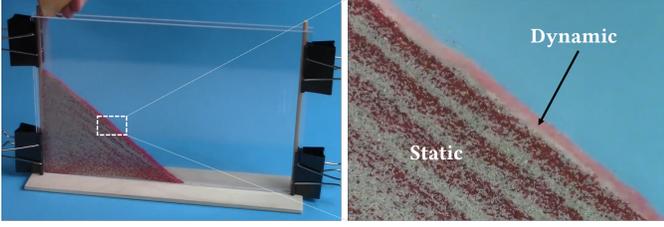


Fig. 2. An experiment demonstrating the distinctive behaviors of dry sand. ©SIMONS FOUNDATION

stylized sand art drawing based on image editing. Recently, Chen et al. [40] developed a GPU-based real-time 3D oil painting system. However, none of existing systems are able to create physically plausible sand arts yet. We hope our method can provide a possible solution for digital sand arts.

3 SHALLOW SAND EQUATIONS

In this section, we derive the shallow sand equations motivated by the theory to shallow water equations as well as the distinctive behaviors of dry granular flows.

3.1 Motivation

We first give a brief introduction of the shallow water equations. As we know, the two-dimensional shallow water equations are derived from depth-integrating the three-dimensional Navier-Stokes equations. In case no frictional or viscous force is considered, the corresponding equations are written as follows

$$\begin{aligned} \frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) + \frac{\partial}{\partial y}(hv) &= 0 \\ \frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}\left(\frac{hu^2}{2}\right) + \frac{\partial}{\partial y}(huv) &= -gh\frac{\partial s}{\partial x}, \quad (1) \\ \frac{\partial}{\partial t}(hv) + \frac{\partial}{\partial x}(huv) + \frac{\partial}{\partial y}\left(\frac{hv^2}{2}\right) &= -gh\frac{\partial s}{\partial y} \end{aligned}$$

where h is the water depth, s is the vertical coordinate of the water surface which can be expressed as the sum of the water depth h and the ground elevation b , i.e., $s = h + b$, u and v are the horizontal velocities of the fluid and g is the gravity. The first equation is given according to the mass conservation law while the second two equations are given according to the momentum conservation law. The derivation process makes an assumption of the fluid domain that its vertical length scale is much smaller than its horizontal length scale, indicating that the horizontal velocity can be assumed to be constant along the depth of the fluid.

Let us investigate the major features of dry sands now. Figure 2 shows a typical experiment demonstrating the distinctive behaviors of dry sands. First, unlike water, the majority of the sand remains static during the sand piling process, indicating the shallowness assumption used in the S-H model is violated. Second, by observing the dynamic process, it can be noticed that sand flowing occurs only near the free surface area. These two features motivate us to simulate a dry granular media by dividing it into two layers: a *dilute layer* wherein particles can flow freely like a fluid, and a *dense layer* wherein particles behave as a solid,

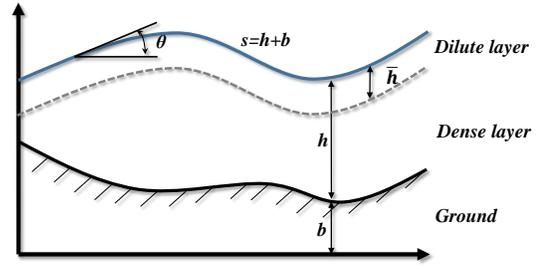


Fig. 3. An illustration of our two-layer model for dry granular materials.

as shown in Figure 3. In the dilute layer, we further assume the pressure magnitude is proportional to the sand depth, similar to a shallow water. In the dense layer, we assume the pressure magnitude is constant and independent of the sand depth, similar to a rigid body. With these assumptions, we will derive the shallow sand equations for dry granular flows in the following section.

3.2 Derivation

Considering a sand bed whose depth of the dilute layer is denoted as a variable \bar{h} and assuming the density ρ to be constant along the z -axis, the pressure at an arbitrary depth z is written as

$$p(z) = \begin{cases} \rho g(s - z), & s - \bar{h} < z \leq s \\ P_0, & b \leq z \leq s - \bar{h} \end{cases}, \quad (2)$$

where we have assumed the atmospheric pressure is 0 and P_0 is a constant value. Integrating the pressure gradient along the z -axis, the horizontal pressure force density imposed on the whole vertical column can be simplified as

$$\mathbf{F}^p = -\frac{1}{\rho} \int_b^s \begin{bmatrix} \partial p / \partial x \\ \partial p / \partial y \end{bmatrix} dz = -g\bar{h} \begin{bmatrix} \partial s / \partial x \\ \partial s / \partial y \end{bmatrix}. \quad (3)$$

Next, we consider how to calculate the frictional force based on a Coulomb friction model. According to the definition of friction, it should always resist the relative motion of the dilute layer against the dense layer. Therefore, the frictional force exerted on the dilute layer can be formulated as

$$\mathbf{F}^f = -\mu f(\mathbf{F}^p) \mathbf{m}, \quad (4)$$

where μ is the friction coefficient, \mathbf{m} is a normalized vector of the horizontal velocity, i.e., $\mathbf{m} = (\frac{u}{\sqrt{u^2+v^2}}, \frac{v}{\sqrt{u^2+v^2}})$, and $f(\cdot)$ indicates that the magnitude of the friction depends on the pressure force. To determine $f(\cdot)$, we consider a wood block sliding on a ramp with an inclination angle of θ , as shown in Figure 4(a). By taking some simple operations, the relationship between the horizontal component of the normal pressure force and the frictional force can be expressed as $|F^t| = \mu \cot \theta |F^n|$. Similarly, we assume the pressure force \mathbf{F}^p and the frictional force \mathbf{F}^f in the sand bed also satisfies the above relationship, i.e., $\|\mathbf{F}^f\| = \mu \cot \theta \|\mathbf{F}^p\|$, where θ should be the dihedral angle in a three-dimensional space, as shown in Figure 4(b). Since $\cot \theta$ can be formulated as

$$\cot \theta = \frac{1}{\sqrt{(\partial s / \partial x)^2 + (\partial s / \partial y)^2}}, \quad (5)$$

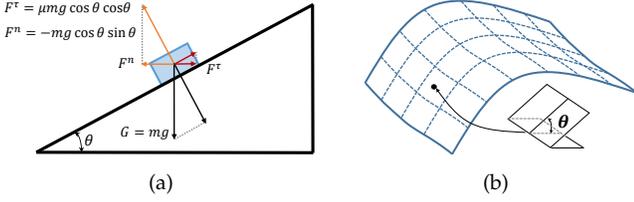


Fig. 4. (a) A wood block sliding on a ramp with an inclination angle θ ; (b) An illustration of the dihedral angle for a point on the surface.

the final formulation for the frictional force can be derived by cancelling out the denominator

$$\mathbf{F}^f = -\mu g \bar{h} \mathbf{m} \quad (6)$$

Therefore, the shallow sand equations are written as

$$\begin{aligned} \frac{\partial h}{\partial t} + \frac{\partial}{\partial x}(hu) + \frac{\partial}{\partial y}(hv) &= 0 \\ \frac{\partial}{\partial t}(hu) + \frac{\partial}{\partial x}\left(\frac{hu^2}{2}\right) + \frac{\partial}{\partial y}(huv) &= -g\bar{h}\left(\frac{\partial s}{\partial x} + \mu \mathbf{m}_x\right), \\ \frac{\partial}{\partial t}(hv) + \frac{\partial}{\partial x}(huv) + \frac{\partial}{\partial y}\left(\frac{hv^2}{2}\right) &= -g\bar{h}\left(\frac{\partial s}{\partial y} + \mu \mathbf{m}_y\right) \end{aligned} \quad (7)$$

The value of μ , which should not be greater than the coefficient of kinetic friction μ_k , will be dynamically calculated (see Section 4.2 for details). \mathbf{m}_x and \mathbf{m}_y represent the x and y components of \mathbf{m} , respectively. In our model, we also assume the coefficient of static friction is equal to μ_k . Be aware that u and v represent the depth-averaged velocities, therefore the momentum contribution from the dilute layer will be averaged on the whole column.

4 SIMULATION OF DRY GRANULAR FLOWS ON A HEIGHT FIELD

In this section, we will discuss how to numerically solve the shallow sand equations. We introduce a uniform two-dimensional grid $I_{i,j} := (i\Delta L, j\Delta L)$ with ΔL representing the grid spacing, and denote the cell averages of the solution as $\mathbf{U}_{i,j} = (h, hu, hv)_{i,j}^T$. By applying a central-upwind semi-discretization, we split Equation 7 into the following steps

- 1 Advection (§4.1). At the beginning of each simulation step, we first take an explicit advection as follows

$$\mathbf{U}'_{i,j} = \mathbf{U}_{i,j} - \Delta t \left(\frac{\mathbf{H}_{i+\frac{1}{2},j}^t - \mathbf{H}_{i-\frac{1}{2},j}^t}{\Delta L} - \frac{\mathbf{H}_{i,j+\frac{1}{2}}^t - \mathbf{H}_{i,j-\frac{1}{2}}^t}{\Delta L} \right), \quad (8)$$

where \mathbf{H} represents the numerical flux between adjacent grid cells, e.g., $\mathbf{H}_{i+\frac{1}{2},j}$ measures the flux between $I_{i,j}$ and $I_{i+1,j}$.

- 2 Applying the pressure force (§4.2). Then, we explicitly apply the pressure force by temporarily ignoring the frictional force

$$\mathbf{U}_{i,j}^* = \mathbf{U}'_{i,j} + \Delta t \mathbf{F}_{i,j}^{p,t}, \quad (9)$$

- 3 Applying the friction (§4.2). Finally, we calculate a compatible frictional force implicitly, i.e., $\|\mathbf{F}_{i,j}^{f,t+\Delta t}\| \leq \mu_k \|\mathbf{F}_{i,j}^{p,t}\|$, to get the final solution

$$\mathbf{U}_{i,j}^{t+\Delta t} = \mathbf{U}_{i,j}^* + \Delta t \mathbf{F}_{i,j}^{f,t+\Delta t}. \quad (10)$$

More details on how to take each step will be discussed in the following sections.

4.1 Advection

To obtain a stable and accurate simulation, we apply a second-order central-upwind scheme proposed by Kurganov and his colleagues [41] to advect \mathbf{U} . For completeness, we give the exact definition of $\mathbf{H}_{i+\frac{1}{2},j}^t$ as follows

$$\begin{aligned} \mathbf{H}_{i+\frac{1}{2},j}^t &= \frac{a_{i+\frac{1}{2},j}^+ \mathbf{T}_{i,j} - a_{i+\frac{1}{2},j}^- \mathbf{T}_{i+1,j}}{a_{i+\frac{1}{2},j}^+ - a_{i+\frac{1}{2},j}^- + \delta} \\ &\quad + \frac{a_{i+\frac{1}{2},j}^+ a_{i+\frac{1}{2},j}^-}{a_{i+\frac{1}{2},j}^+ - a_{i+\frac{1}{2},j}^- + \delta} [\mathbf{U}_{i+1,j} - \mathbf{U}_{i,j}], \end{aligned} \quad (11)$$

where δ is a predefined positive value to avoid a division-by-zero error, \mathbf{T} is defined as $\mathbf{T} = (hu, \frac{hu^2}{2}, huv)$, $a_{i+\frac{1}{2},j}^+$ and $a_{i+\frac{1}{2},j}^-$ represent the one-sided local speeds of propagation along the x direction, which are formulated as

$$\begin{aligned} a_{i+\frac{1}{2},j}^+ &= \max \left\{ u_{i,j} + \sqrt{gh_{i,j}}, u_{i+1,j} + \sqrt{gh_{i+1,j}}, 0 \right\}, \\ a_{i+\frac{1}{2},j}^- &= \min \left\{ u_{i,j} - \sqrt{gh_{i,j}}, u_{i+1,j} - \sqrt{gh_{i+1,j}}, 0 \right\}. \end{aligned} \quad (12)$$

Note that $a_{i+\frac{1}{2},j}^+$ and $a_{i+\frac{1}{2},j}^-$ may be zero if both $h_{i,j}$ and $u_{i,j}$ are zero. Therefore, the threshold δ helps stabilize the simulation. Other numerical fluxes, such as $\mathbf{H}_{i-\frac{1}{2},j}$, $\mathbf{H}_{i,j+\frac{1}{2}}$ and $\mathbf{H}_{i,j-\frac{1}{2}}$, can also be calculated in the same way as $\mathbf{H}_{i+\frac{1}{2},j}$. According to the definition of \mathbf{H} , we can verify that the total volume is preserved.

In our current implementation, we only store the values for h , hu and hv , a simple way to compute the horizontal velocity could be using $u = (hu)/h$. However, this method may cause numerical instability if the value of h is too small or even zero. To overcome this difficulty, we introduce a prescribed positive number $\varepsilon = \max \{\Delta x^4, \Delta y^4\}$ and compute the horizontal velocity using the following formulas

$$u = \frac{\sqrt{2h}(hu)}{\sqrt{h^4 + \max(h^4, \varepsilon)}}, \quad v = \frac{\sqrt{2h}(hv)}{\sqrt{h^4 + \max(h^4, \varepsilon)}}. \quad (13)$$

Notice that Equation 13 is equivalent to $u = (hu)/h, v = (hv)/h$ for large values of h . But when h is small, to preserve positivity for h , we should re-calculate hu and hv with $hu = h \cdot u$ and $hv = h \cdot v$. Therefore, the entire algorithm can remain consistent provided that the following condition is satisfied

$$\Delta t \leq \min \left\{ \frac{\Delta x}{4a}, \frac{\Delta y}{4b} \right\}, \quad (14)$$

where a and b are the maximum one-sided local speeds of propagation along the x and y direction, respectively.

$$a := \max_{i,j} \left\{ \max \left\{ a_{i+\frac{1}{2},j}^+, a_{i+\frac{1}{2},j}^- \right\} \right\} \quad (15)$$

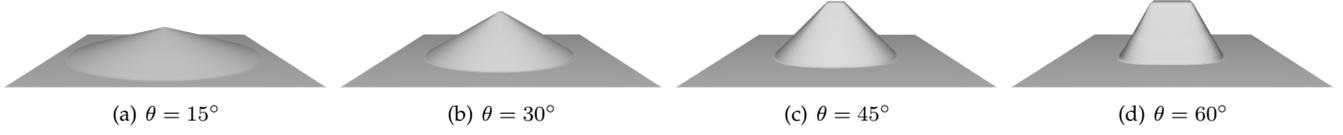


Fig. 5. Sand piles simulated with four different friction angles $\theta = 15^\circ, 30^\circ, 45^\circ, 60^\circ$.

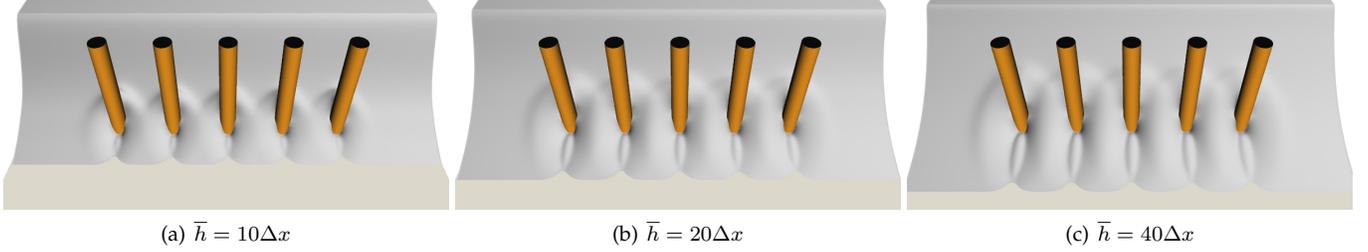


Fig. 6. Dry granular flows with three different constant values of \bar{h} .

$$b := \max_{i,j} \left\{ \max \left\{ b_{i,j+\frac{1}{2}}^+, b_{i,j+\frac{1}{2}}^- \right\} \right\} \quad (16)$$

Please refer to [41] for a strict proof of the above positivity preserving property.

4.2 Update velocity

To correctly solve the momentum equations, we apply a prediction-correction scheme to update the velocity for each grid cell. At the prediction step, we only consider the pressure force and take an explicit forward Euler method. By applying a central difference scheme, we compute the pressure force as follows

$$\mathbf{F}_{i,j}^{p,t} = -g \min(h_{i,j}^t, \bar{h}_{i,j}) \begin{bmatrix} \frac{s_{i+1,j}^t - s_{i-1,j}^t}{2\Delta L} \\ \frac{s_{i,j+1}^t - s_{i,j-1}^t}{2\Delta L} \end{bmatrix}, \quad (17)$$

where g is always set to $9.8m/s^2$. Note that the depth of the dilute layer should not be greater than the total depth. Therefore, we use $h_{i,j}$ instead of $\bar{h}_{i,j}$ to compute the pressure force if $h_{i,j} \leq \bar{h}_{i,j}$.

After adding the pressure force and getting an intermediate velocity $(hu, hv)_{i,j}^*$, we check how much frictional force should be imposed for each grid cell. In continuum mechanics, according to the principle of maximum plastic dissipation [8], [42], the actual frictional stress is the one which maximizes the rate of dissipation of kinetic energy. By applying this rule, the actual frictional force for each grid cell can be obtained by solving the following optimization problem

$$E = \min_{\mu \in [0, \mu_k]} \frac{1}{2} \left\| (hu, hv)_{i,j}^* - \Delta t \min(h_{i,j}, \bar{h}_{i,j}) \mu g \mathbf{m}_{i,j}^* \right\|^2 \quad (18)$$

Since the two vectors $(hu, hv)_{i,j}^*$ and $\mathbf{m}_{i,j}^*$ are parallel, we can solve Equation 18 by simply comparing the values between $e_1 = \|(hu, hv)_{i,j}^*\|$ and $e_2 = |\Delta t \min(h_{i,j}, \bar{h}_{i,j}) \mu_k g|$. If e_2 is larger, an appropriate value between 0 and μ_k can be set for μ to minimize E to be zero. Otherwise, a maximum value of the frictional force can be reached. To summarize,

the frictional force for each grid cell can be expressed as follows

$$\mathbf{F}_{i,j}^{f,t+\Delta t} = - \begin{cases} \|(hu, hv)_{i,j}^*\| \mathbf{m}_{i,j}^* / \Delta t, & e_1 \leq e_2 \\ \mu_k \min(h_{i,j}, \bar{h}_{i,j}) g \mathbf{m}_{i,j}^*, & e_1 \geq e_2 \end{cases} \quad (19)$$

In Figure 5, we demonstrate four different sand piles with four different friction angles $\theta = 15^\circ, 30^\circ, 45^\circ, 60^\circ$, where the correct angle of repose is recovered. Figure 6 also demonstrates the simulation results of using different \bar{h} . It can be noted that the larger value of \bar{h} we use, the more rapid the dry granules flow.

5 COLLISION WITH KINEMATIC OBJECTS

At the first glance, it seems a variety of solid-fluid interaction techniques developed for shallow water equations [43], [44] can be applied to handle the collision between kinematic objects and the granular media. However, the distinctive behaviors of granular materials will cause problems in calculating forces in a traditional way, e.g., we cannot calculate the buoyancy exerted on the kinematic object by using Archimedes' principle any more. To allow efficient but still plausible colliding between kinematic objects and the granular media, we propose a one-way coupling method to simulate fluid-solid interactions, i.e., the motion of a granular flow can only be affected by kinematic objects but will not have a retroaction on the motion of kinematic objects. Consider a kinematic object that is partially submerged under the sand surface, as shown in Figure 7(a), we assume the overlapped area of the granular media will move instantly at the same speed of the kinematic object. The depth-averaged velocity after taking the overlapping area into account can then be derived from the momentum conservation law as

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} = \frac{1}{\bar{h}} \left[\int_{\Omega_s} (\mathbf{V} + \omega \times \mathbf{r}) dz + \int_{\Omega_f} \begin{pmatrix} u \\ v \\ 0 \end{pmatrix} dz \right], \quad (20)$$

where Ω_s and Ω_f represent the overlapping and non-overlapping areas, respectively, \mathbf{V} is the translational velocity of the kinematic object defined at its mass center \mathbf{c} , ω

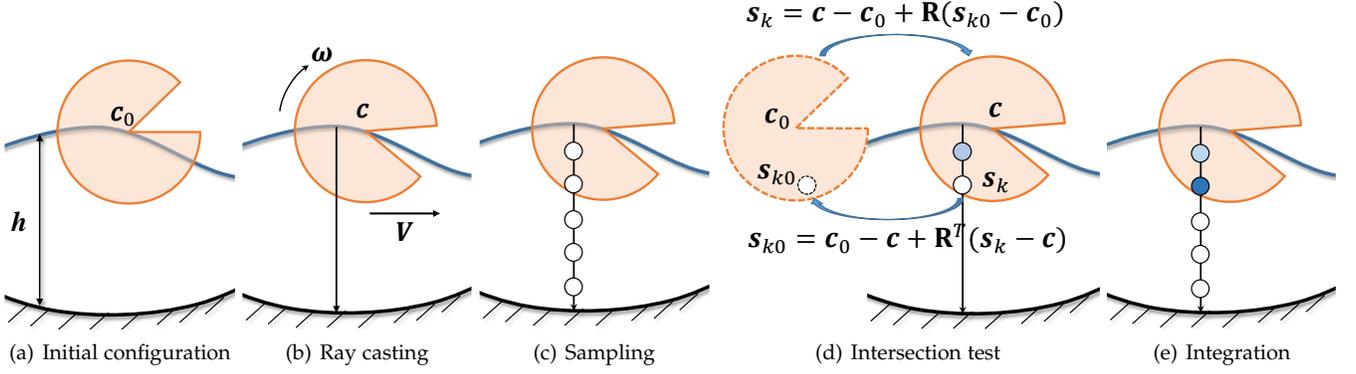


Fig. 7. An illustration of our ray casting algorithm for the one-way solid-fluid coupling.

is the angular velocity and $\mathbf{r} = \mathbf{x} - \mathbf{c}$ represents the relative position of point \mathbf{x} to the mass center. Note that the vertical velocity \tilde{w} may not be zero due to a vertical movement of the kinematic object. Since we only store the depth-averaged horizontal velocities for the granular media, we should transfer the vertical velocity to the horizontal components in order to get physically-plausible vertical interactions. By invoking the shallow sand equations, we actually have

$$\tilde{w} = \frac{\partial h}{\partial t} = -\frac{\partial}{\partial x}(hu)' - \frac{\partial}{\partial y}(hv)'. \quad (21)$$

It means if we know horizontal velocities $(hu)'$ and $(hv)'$ in advance, we can uniquely calculate the vertical velocity \tilde{w} with Equation 21. However, if we only know \tilde{w} , the values of $(hu)'$ and $(hv)'$ cannot be uniquely calculated from \tilde{w} . To address this problem, we simplify the calculation by assuming a uniform transfer of \tilde{w} to all horizontal directions. With this assumption, we first discretize Equation 21 for a grid cell (i, j) into the following form

$$\tilde{w}_{i,j} = -\frac{(hu)'_{i+1,j} - (hu)'_{i-1,j}}{2\Delta L} - \frac{(hv)'_{i,j+1} - (hv)'_{i,j-1}}{2\Delta L} \quad (22)$$

Our assumption implies $(hu)'_{i+1,j} = -(hu)'_{i-1,j} = (hv)'_{i,j+1} = -(hv)'_{i,j-1}$, therefore we have

$$(hu)'_{i+1,j} = -\frac{\Delta L}{2}\tilde{w}_{i,j} \quad (23)$$

The final value of $(hu)_{i,j}$ can then be formulated as

$$(hu)_{i,j} = h_{i,j}\tilde{u}_{i,j} + \frac{\Delta L}{2}(\tilde{w}_{i+1,j} - \tilde{w}_{i-1,j} + \tilde{w}_{i,j+1} - \tilde{w}_{i,j-1}). \quad (24)$$

The value of $(hv)_{i,j}$ can also be calculated in a similar way. Figure 8 shows how Equation 24 can improve the simulation result by taking the vertical component \tilde{w} into account.

Till now, the only unresolved problem is how to calculate \tilde{u} , \tilde{v} and \tilde{w} with Equation 20. Motivated by the volume ray casting algorithm for rendering [45], we propose a ray casting algorithm for efficient solid-fluid coupling, which consists of four steps as follows

- 1 **Ray casting.** For each grid cell, a ray is shot from the surface to the ground.
- 2 **Sampling.** Along the ray within the granular media, equidistant sampling points are selected. We

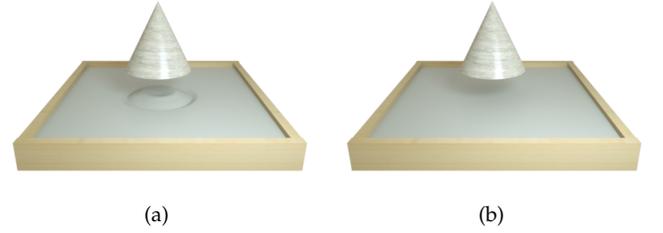


Fig. 8. A comparison between (a) incorporating and (b) without incorporating the vertical motion of the kinematic object.

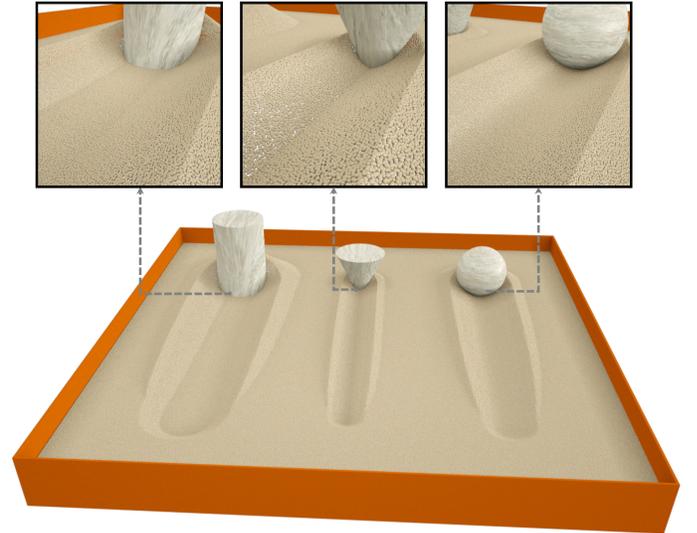


Fig. 9. Three kinematic objects with different shapes interact with the sand and different trails can be noticed.

typically set the sampling distance to be ΔL if not otherwise specified.

- 3 **Intersection test.** For each sampling point \mathbf{s}_k , an infinitesimal value η_k is calculated according to whether \mathbf{s}_i resides inside the kinematic object or not

$$\eta_k = \Delta L \begin{cases} \mathbf{V} + \boldsymbol{\omega} \times (\mathbf{s}_k - \mathbf{c}), & \text{inside} \\ (u, v, 0)^T, & \text{outside} \end{cases} \quad (25)$$

To check whether \mathbf{s}_i resides inside the kinematic object, we discretize the kinematic object into a signed distance field at the beginning of simulation.

During the interaction, we transform s_k back to the local frame of the kinematic object as shown in Figure 7(d). Now, we can easily perform the intersection test by checking whether s_{k0} is inside the kinematic object. Therefore, there is no need to recompute the signed distance as the kinematic object rotated.

- 4 **Integration.** We accumulate all sampling points together to compute \tilde{u} , \tilde{v} and \tilde{w} as follows

$$\begin{pmatrix} \tilde{u} \\ \tilde{v} \\ \tilde{w} \end{pmatrix} = \frac{1}{N\Delta L} \sum_k \eta_k, \quad (26)$$

where N is the total number of sampling points.

Figure 9 demonstrates an object interaction example where three rigid bodies slide on the surface of a sand bed. We can notice different trails due to the different shapes of the rigid bodies.

6 VISUAL REPRESENTATION

We currently use two methods for visualization. The first one is by reconstructing the surface mesh from the height field h , as shown in Figure 1. The other one is by applying particles to track the granular media. In this section, we mainly discuss how to use particles to track the granular media, which consists of three steps. We first introduce a three-dimensional uniform grid $J_{s,t,d} := (s\Delta l, t\Delta l, d\Delta l)$ with Δl representing the grid spacing. Each grid cell stores at most one particle whose position is denoted as $\mathbf{x}_p = (x_p, y_p, z_p)$. For the convenience of discussion, we introduce the following bilinear interpolation function for the grid I to calculate an interpolated value of \mathcal{F} at \mathbf{x}_p

$$\mathcal{F}_I(\mathbf{x}_p) = \sum_{i,j} \mathcal{F}_{i,j} w_{i,j}(\mathbf{x}_p), \quad (27)$$

where $w_{i,j}(\mathbf{x}_p)$ is the weighting function defined as

$$w_{i,j}(\mathbf{x}_p) = N \left(\frac{x_p}{\Delta L} - i \right) N \left(\frac{y_p}{\Delta L} - j \right) \quad (28)$$

with

$$N(x) = \begin{cases} 1 - |x|, & 0 \leq |x| < 1 \\ 0, & \text{otherwise} \end{cases}. \quad (29)$$

For example, $u_I(\mathbf{x}_p)$ and $v_I(\mathbf{x}_p)$ represents the x and y components of the horizontal velocity at \mathbf{x}_p , respectively.

6.1 Particle generation

The guiding principle of generating particles is to keep the particle distribution dynamically regular as well as dense enough to cover all interested areas. An ideal sampling algorithm is to use the Poisson disk sampling method [46]. However, generating a large number of particles with the Poisson disk sampling method is time-consuming. Therefore, we instead assign each component of \mathbf{x}_p with a random-number generator as follows

$$\mathbf{x}_p = \Delta l \begin{pmatrix} s + \mathcal{R}(0, 1) \\ t + \mathcal{R}(0, 1) \\ d + \mathcal{R}(0, 1) \end{pmatrix} \quad (30)$$

where $\mathcal{R}(0, 1)$ is a function that generates a scalar value between 0 and 1.

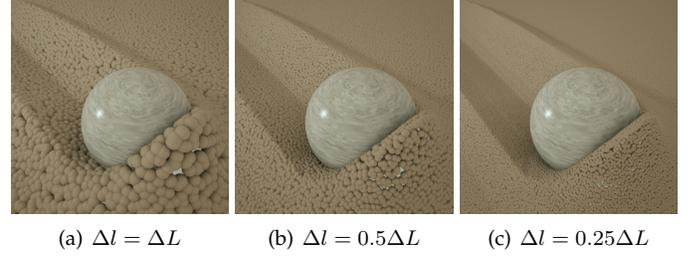


Fig. 10. Particle-based visualization for the granular media under three different resolutions.

6.2 Particle advection

We assume all particles can only move horizontally. In other words, if a particle \mathbf{x}_p resides in the $d - th$ layer of grid J at the beginning of advection, it will still reside in the $d - th$ layer after advection. During the advection, we first calculate the horizontal velocity $u_I(\mathbf{x}_p)$ and $v_I(\mathbf{x}_p)$ at \mathbf{x}_p . Then, we move the particle forward to a new position \mathbf{x}'_p according to the following formulation

$$\mathbf{x}'_p = \mathbf{x}_p + \Delta t \begin{pmatrix} u_I(\mathbf{x}_p) \\ v_I(\mathbf{x}_p) \\ 0 \end{pmatrix}. \quad (31)$$

The new grid cell indexes can then be identified as

$$s' = \text{floor}\left(\frac{x'_p}{\Delta l}\right), \quad t' = \text{floor}\left(\frac{y'_p}{\Delta l}\right), \quad d' = d, \quad (32)$$

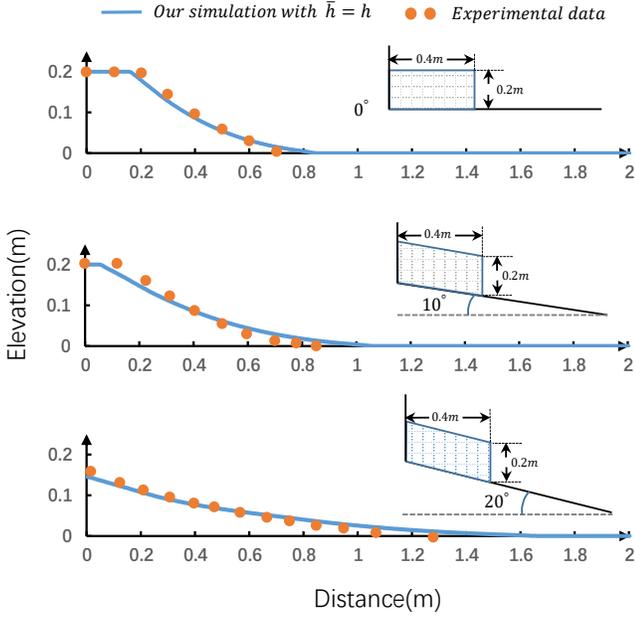
where $\text{floor}(x)$ is a function returning the largest integer number that is not greater than x . Finally, we take a conflict check to ensure that each new grid cell $J_{s',t',d'}$ still stores at most one particle. In case more than one particles enter the same grid cell, we only retain the last entered particle. For empty cells, we generate new particles as discussed in Section 6.1.

6.3 Transforming particles

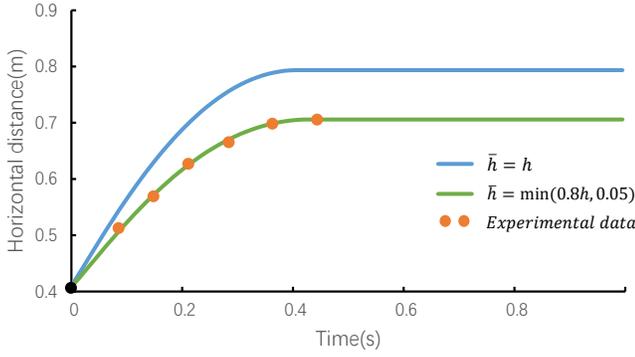
The final step is to transform all particles' coordinates according to the height of the sand bed. This step is only used for visualization and will not change the real positions of particles. Considering a particle at \mathbf{x}_p , we first compute the surface height $s_I(\mathbf{x}_p)$. The transformed position is then expressed as

$$\mathbf{x}_p^h = \begin{pmatrix} x_p \\ y_p \\ s_I(\mathbf{x}_p) - z_p \end{pmatrix}. \quad (33)$$

After transformation, if a particle locates at a grid cell $I_{i,j}$ whose depth is smaller than a predefined threshold or the vertical coordinate of the particle is lower than the ground, we will disable the display of this particle. Figure 10 shows a comparison of our particle-based enrichment with three different resolutions of particles. We can notice improved visual effects as the resolution increases. Nevertheless, we can still notice a visual artifact that the motion of the particles seems to be going in a counter-intuitive direction. This is because the velocity originated from the dilute layer is averaged across the whole volume of the sand. Therefore, the particles actually move slower than expected if $\bar{h} < h$. One possible solution is to multiply the advection velocity



(a) Comparison of the final deposit profiles with experiments



(b) Comparison of front progress curves with experiments

Fig. 11. A Dambreak example, which is 0.4 meters long and 0.2 meters deep, is set up to evaluate the accuracy of our method. The grid spacing is set to 0.01m.

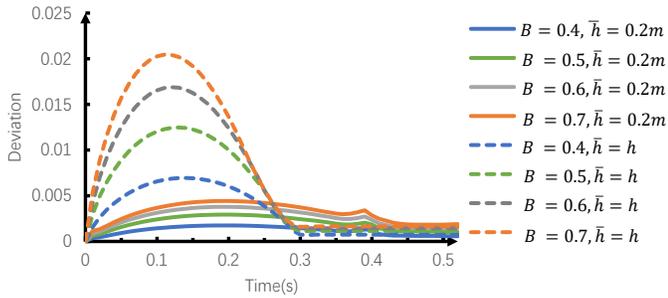


Fig. 12. A second dambreak case to verify the technical soundness of using \bar{h} . The deviation is calculated with $\|h_B - h_{0.3}\|$.

$u_I(\mathbf{x}_p)$ and $v_I(\mathbf{x}_p)$ with a scaling factor, e.g., h/\bar{h} . Please refer to the video for such a comparison. However, for large-scale simulations where the motion of individual particles is not our concern, we can also apply a level set for rendering, as shown in Figure 13 and 17.

7 RESULTS AND DISCUSSIONS

We implemented our method in CUDA and ran all simulations on an NVIDIA GTX 1060. To avoid write conflicts during the particle advection, we achieve mutual exclusion by applying CUDA atomic operations. The timestep size should be chosen as the minimum of Equation 14 and the following CFL condition

$$\Delta t \leq C \min_{i,j} \left\{ \frac{\Delta x}{u_{i,j}}, \frac{\Delta y}{v_{i,j}} \right\}. \quad (34)$$

where C represents the Courant number. However, for GPU-based real-time applications, a fixed-size timestep is sometimes preferable. Therefore, in certain cases, we also estimate an appropriate timestep from Equation 14 and 34 and use it for the whole simulation. For the boundary wall condition, we use ghost cells to realize wall boundary conditions. We mirror the value of ghost cells to the nearest boundary cell. Therefore, the flux through the boundary is zero, which can be verified from Equation 11. Table 1 summarizes the statistics and timings for all examples.

7.1 Evaluation

To evaluate the accuracy of our method, we set up a dam-break, which is 0.4 meters long and 0.2 meters deep, on a flume. The internal and basal friction angles are both set to 30.9° . The real experimental results for this example can also be found in [47].

In our simulation, three tests are carried out with the inclination angle of the flume being set to 0° , 10° and 20° , respectively. Note that the setup of the last two tests differ slightly from the real experiment because our shallow sand equations are derived with respect to the horizontal plane. We first assume the granular media is shallow, which means \bar{h} was just set to h during the simulation. The simulation results are plotted at Figure 11, together with the experimental data. The comparison in Figure 11(a) shows that our simulation with the shallowness assumption leads to a slight overestimation of runout in all three cases. For the example with an inclination angle of 0° , we further demonstrate a comparison of the front (defined as $h = 0.05m$) progress curve with experimental data. From above comparisons, it can be concluded that a simple shallowness assumption is not enough to generate simulation results that exactly match the real data, which explain the reasonableness of using a two-layer model to describe dry granular flows. By setting $\bar{h} = \min(0.8h, 0.05)$, Figure 11(b) shows a simulation result that converges exactly to the experimental data.

For further verification, we setup another dambreak example whose initial profile is expressed as follows

$$h_B(x, t = 0) = \begin{cases} 0.4 + B, & 0 \leq x < 1, \\ B, & 1 \leq x \leq 2, \end{cases} \quad (35)$$

where B can be any positive number. According to our intuitive understanding, the flowing pattern for this example should remain approximately the same for different values of B . However, if we apply the shallowness assumption and set $\bar{h} = h$, we can notice significant differences of the simulation results with different B , as shown in Figure 12. Nevertheless, this deviation can be greatly minimized by



Fig. 13. A psychological sand table.

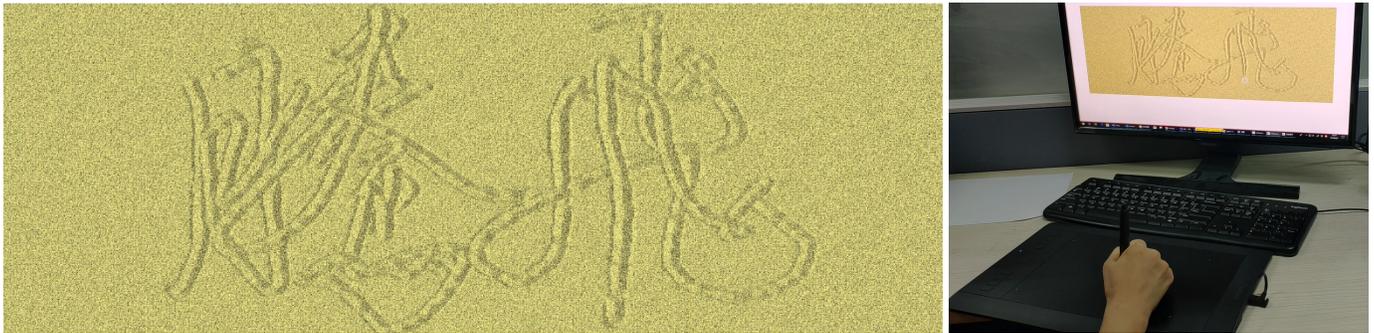


Fig. 14. Sand painting. We develop a simple sand painting system which is used to write Chinese words (The meaning is flying).

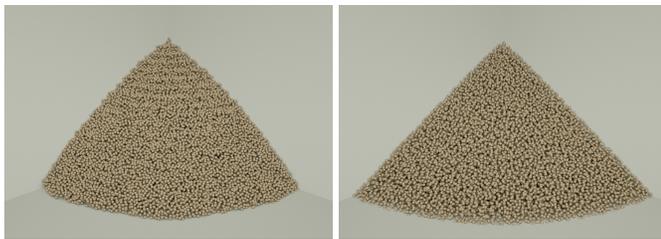


Fig. 15. A comparison between the simulation results produced by our method (right) and Narain et al. [8]'s method(left).



Fig. 16. Rotating fan blades interact with the dry granular media.

setting \bar{h} to a constant value (e.g., $\bar{h} = 0.2m$) according to our two-layer model.

Generally speaking, there is no uniform standard for us to choose the right value of \bar{h} because the dynamic behaviors of granular flows depend on a lot of factors, such as the particle mass, particle size, particle shape, etc. However, for a specific material, the above two evaluations provide us some insights on how to choose the right value. Most of time, we can just set \bar{h} to a constant value if the simulation accuracy is not our priority. Otherwise, we can improve the accuracy by adjusting the value of \bar{h} , either by multiplying a scaling factor or with a numerical truncation.

Comparison to [8] We also evaluate the flexibility and efficiency by comparing our method to a three-dimensional solver proposed by Narain et al. [8]. The three-dimensional solver is executed on a $64 \times 64 \times 64$ uniform grid while ours on a 64×64 height field. The friction angles for both methods are set to 30° . To match the visual appearance, we use 50 layers of particles for visual enrichment for our method. From Figure 15, it can be noted that the simulation result produced by our solver is quite similar to that produced by the Narain et al [8]. However, our method gains nearly two orders of magnitude speedup over their method in terms of

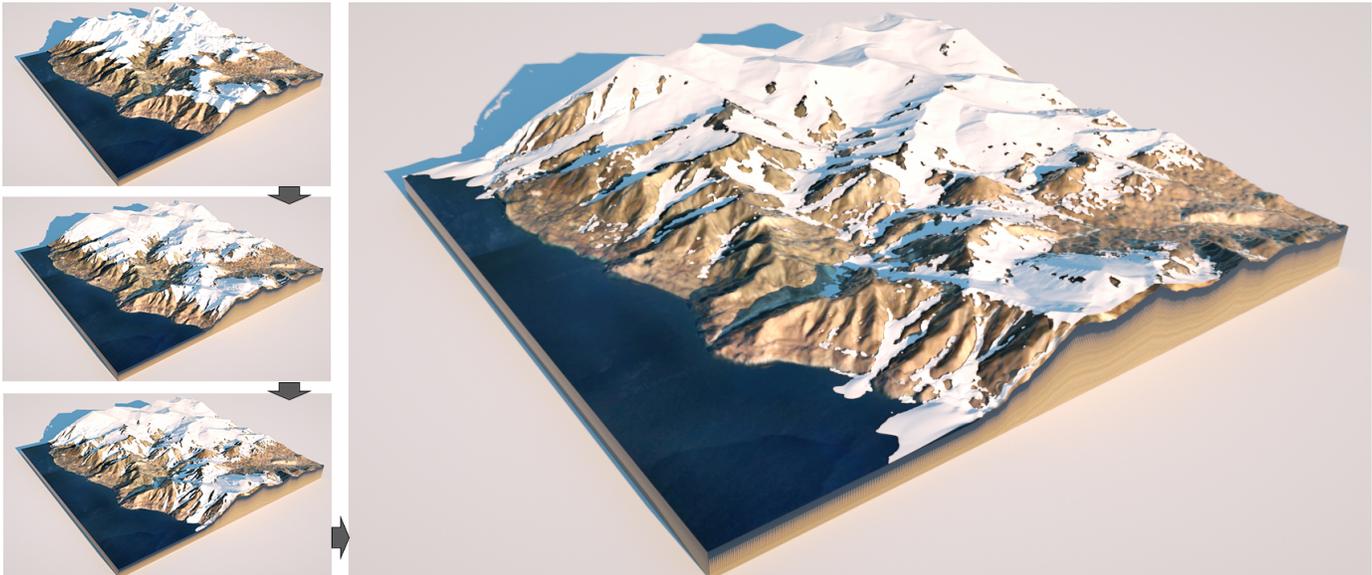


Fig. 17. A snow avalanche. Our method is able to simulate large-scale dry granular media in real-time.

Name	I	J	θ	$h(\Delta L)$	$\Delta t(s)$	$t_d(ms)$	$t_i(ms)$	$t_p(ms)$
Figure 1	512×512	-	20	h	$C = 0.1$	20	-	-
Figure 1	256×256	$1024 \times 1024 \times 1$	30	10	$C = 0.1$	4.8	4.1	70
Figure 5	64×64	-	15~60	5	0.01	< 1	-	-
Figure 6	256×256	-	30	10~40	0.01	1.6	-	-
Figure 9	256×256	$1024 \times 1024 \times 1$	30	10	$C = 0.1$	4.8	3.8	11
Figure 13	512×256	-	30	10	$C = 0.1$	10.3	5.8	-
Figure 14	512×256	$1024 \times 512 \times 1$	30	10	0.002	5.1	4.1	9
Figure 15(b)	64×64	$128 \times 128 \times 50$	30	10	$C = 0.1$	< 1	-	65
Figure 16	128×128	$512 \times 512 \times 1$	30	10	$C = 0.1$	1.6	1.2	17

TABLE 1

Statistics and timings per frame of our examples. t_d represents the average computational cost for solving the shallow sand equations, t_i represents the average computational cost for fluid-solid interactions, t_p represents the average computational cost for particle tracking. “-” means the corresponding item is not considered.

the computational time for solving the dynamics. Although the performance of their method is currently tested on CPU, we believe our method should still run much faster than a three-dimensional solver if both are implemented on GPU.

7.2 More examples

In Figure 1, we simulate a snow avalanche whose friction angle is set to 20° . We reconstruct the snow surface with a 2D marching cubes algorithm for an iso-value of $h = 0.1\Delta L$. Figure 1 shows the drawing of a butterfly with a wooden stick, which is then flattened with a long wooden block. Figure 13 shows a psychological sand table with two moving cars. Figure 16 shows the interaction between rotating fan blades and the dry granular media. In Figure 14, we simulate a sand canvas with our method. The painting stroke is interpolated with the latest two input points and the stroke width is initialized to 5 times of the grid spacing. In this example, we simply consider the interaction as a cylinder and do the intersection test analytically.

8 CONCLUSION

We have proposed shallow sand equations for real-time simulation of dry granular flows. Our method is able to capture complex behaviors of dry granular flows, e.g., sand

piling with different friction angles. Besides, the flowing pattern can be manually controlled by changing the depth of the dilute layer. We also introduce an efficient ray casting algorithm for one-way solid-fluid coupling, allowing us to use rigid bodies with different shapes as the interaction tool. To improve the visual effects, we propose to use particles to track the granular media. Experiments show that our method is efficient and flexible to simulate both shallow and relatively deep granular flows.

Limitations We have applied two heuristics to divide the simulation domain into two regions. According to our evaluation, this assumption works quite well for granular flows whose vertical length scale is smaller than the horizontal length scale. However, if the vertical length is much larger, certain kind of mismatch could arise during the simulation, as shown in Figure 15. Although this can always be improved by adjusting the value of h , choosing an appropriate value of \bar{h} for an arbitrary complex scenario is usually not an easy task. For a specific granular media, whether there exists a standard rule that helps us match the simulation result exactly to real experimental data is not clear yet. Besides, since our method only uses the height field to solve the dynamics of dry granular flows, its ability in modeling scenarios involving sand splashes is also limited. One possible solution could be to integrate a Lagrangian method (e.g., SPH) into our method to model

free-flowing particles. Finally, the object interaction method is not accurate enough yet, which means our method cannot strictly prevent from particle entering rigid bodies. However, for bulky bodies, this artifact is usually not obvious.

Future work We will investigate the possibility of extending our method to model granular flows with cohesion. We would also like to implement more object interaction methods, such a fully solid-fluid coupling. Finally, it would be interesting to integrate our method into a real-time sand painting system to achieve realistic sand arts.

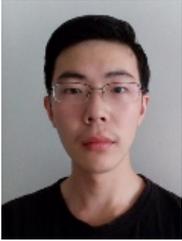
REFERENCES

- [1] P.-G. de Gennes, "Granular matter: a tentative view," *Reviews of modern physics*, vol. 71, no. 2, p. S374, 1999.
- [2] S. P. Pudasaini and K. Hutter, *Avalanche dynamics: dynamics of rapid flows of dense granular avalanches*. Springer Science & Business Media, 2007.
- [3] B. E. Feldman and J. F. O'Brien, "Modeling the accumulation of wind-driven snow," in *ACM SIGGRAPH 2002 conference abstracts and applications*. ACM, 2002, pp. 218–218.
- [4] A. Stomakhin, C. Schroeder, L. Chai, J. Teran, and A. Selle, "A material point method for snow simulation," *ACM Trans. Graph.* (SIGGRAPH), vol. 32, no. 4, p. 102, 2013.
- [5] M. Pauly, R. Keiser, B. Adams, P. Dutré, M. Gross, and L. J. Guibas, "Meshless animation of fracturing solids," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 957–964, Jul. 2005.
- [6] Z. Bao, J. m. Hong, J. Teran, and R. Fedkiw, "Fracturing rigid materials," *IEEE Transactions on Visualization and Computer Graphics*, vol. 13, no. 2, pp. 370–378, March 2007.
- [7] D. Hahn and C. Wojtan, "High-resolution brittle fracture simulation with boundary elements," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 4, p. 151, 2015.
- [8] R. Narain, A. Golas, and M. C. Lin, "Free-flowing granular materials with two-way solid coupling," *ACM Transactions on Graphics (TOG)*, vol. 29, no. 6, p. 173, 2010.
- [9] G. Klár, T. Gast, A. Pradhana, C. Fu, C. Schroeder, C. Jiang, and J. Teran, "Drucker-prager elastoplasticity for sand animation," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 103:1–103:12, Jul. 2016.
- [10] X. He, H. Wang, and E. Wu, "Projective peridynamics for modeling versatile elastoplastic materials," *IEEE transactions on visualization and computer graphics*, 2017.
- [11] O. Pouliquen and F. Chevoir, "Dense flows of dry granular material," *Comptes Rendus Physique*, vol. 3, no. 2, pp. 163–175, 2002.
- [12] S. B. Savage and K. Hutter, "The motion of a finite mass of granular material down a rough incline," *Journal of fluid mechanics*, vol. 199, pp. 177–215, 1989.
- [13] A. Luciani, A. Habibi, and E. Manzotti, "A multi-scale physical model of granular materials," in *Graphics interface'95*, 1995, pp. 136–146.
- [14] N. Bell, Y. Yu, and P. J. Mucha, "Particle-based simulation of granular materials," in *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*. ACM, 2005, pp. 77–86.
- [15] W. Rungtirananon, Z. Szego, Y. Kanamori, and T. Nishita, "Real-time animation of sand-water interaction," *Computer Graphics Forum*, vol. 27, no. 7, pp. 1887–1893, 2008.
- [16] T. Lenaerts and P. Dutré, "Mixing fluids and granular materials," *Computer Graphics Forum*, vol. 28, no. 2, pp. 213–218, 2009.
- [17] I. Alduán and M. A. Otaduy, "Sph granular flow with friction and cohesion," in *Proceedings of the 2011 ACM SIGGRAPH/Eurographics symposium on computer animation*. ACM, 2011, pp. 25–32.
- [18] I. Alduán, A. Tena, and M. A. Otaduy, "Simulation of high-resolution granular media." in *CEIG*, 2009, pp. 11–18.
- [19] Y. Chang, K. Bao, J. Zhu, and E. Wu, "A particle-based method for granular flow simulation," *Science China Information Sciences*, vol. 55, no. 5, pp. 1062–1072, 2012.
- [20] J.-P. Longmore, P. Marais, and M. M. Kuttel, "Towards realistic and interactive sand simulation: A gpu-based framework," *Powder technology*, vol. 235, pp. 983–1000, 2013.
- [21] M. Ihmsen, A. Wahl, and M. Teschner, "A lagrangian framework for simulating granular material with high detail," *Computers & Graphics*, vol. 37, no. 7, pp. 800–808, 2013.
- [22] X. Yan, Y.-T. Jiang, C.-F. Li, R. R. Martin, and S.-M. Hu, "Multiphase sph simulation for interactive fluids and solids," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 79:1–79:11, Jul. 2016. [Online]. Available: <http://doi.acm.org/10.1145/2897824.2925897>
- [23] Y. Zhu and R. Bridson, "Animating sand as a fluid," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 965–972, Jul. 2005. [Online]. Available: <http://doi.acm.org/10.1145/1073204.1073298>
- [24] D. Nkulikiyimfura, J.-I. Kim, and H. Kim, "A real-time sand simulation using a gpu," in *2012 8th International Conference on Computing Technology and Information Management (ICCM)*, vol. 1. IEEE, 2012, pp. 495–498.
- [25] G. Daviet and F. Bertails-Descoubes, "A semi-implicit material point method for the continuum simulation of granular materials," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 102, 2016.
- [26] M. Gao*, X. Wang*, K. Wu*, A. Pradhana, E. Sifakis, C. Yuksel, and C. Jiang, "Gpu optimization of material point methods," *ACM Transactions on Graphics (Proceedings of SIGGRAPH ASIA 2018)*, vol. 37, no. 6, 2018.
- [27] Y. Yue, B. Smith, P. Y. Chen, M. Chantharayukhonthorn, K. Kamrin, and E. Grinspun, "Hybrid grains: Adaptive coupling of discrete and continuum simulations of granular media," *ACM Trans. Graph.*, vol. 37, no. 6, pp. 283:1–283:19, Nov. 2018.
- [28] J.-b. Fei, Y.-x. Jie, B.-y. Zhang, and X.-d. Fu, "A shallow constitutive law-based granular flow model for avalanches," *Computers and Geotechnics*, vol. 68, pp. 109–116, 2015.
- [29] F. Bouchut, A. Mangeney-Castelnaud, B. Perthame, and J.-P. Vilotte, "A new model of saint venant and savage-hutter type for gravity driven shallow water flows," *Comptes rendus mathématique*, vol. 336, no. 6, pp. 531–536, 2003.
- [30] E. D. Fernández-Nieto, F. Bouchut, D. Bresch, M. C. Diaz, and A. Mangeney, "A new savage-hutter type model for submarine avalanches and generated tsunami," *Journal of Computational Physics*, vol. 227, no. 16, pp. 7720–7754, 2008.
- [31] S. McDougall and O. Hungr, "A model for the analysis of rapid landslide motion across three-dimensional terrain," *Canadian Geotechnical Journal*, vol. 41, no. 6, pp. 1084–1097, 2004.
- [32] O. Hungr, "A model for the runout analysis of rapid flow slides, debris flows, and avalanches," *Canadian Geotechnical Journal*, vol. 32, no. 4, pp. 610–623, 1995.
- [33] X. Li and J. M. Moshell, "Modeling soil: Realtime dynamic models for soil slippage and manipulation," in *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '93. New York, NY, USA: ACM, 1993, pp. 361–368. [Online]. Available: <http://doi.acm.org/10.1145/166117.166162>
- [34] R. W. Sumner, J. F. O'Brien, and J. K. Hodgins, "Animating sand, mud, and snow," *Computer Graphics Forum*, vol. 18, no. 1, pp. 17–26, 1999.
- [35] K. Onoue and T. Nishita, "An interactive deformation system for granular material," *Computer Graphics Forum*, vol. 24, no. 1, pp. 51–60, 2005.
- [36] B. Zhu and X. Yang, "Animating sand as a surface flow." in *Eurographics (Short Papers)*, 2010, pp. 9–12.
- [37] R. H. Kazi, K. C. Chua, S. Zhao, R. Davis, and K.-L. Low, "Sand-canvas: a multi-touch art medium inspired by sand animation," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2011, pp. 1283–1292.
- [38] P.-Y. Chen and S.-K. Wong, "Real-time auto stylized sand art drawing," in *2013 International Conference on Computer-Aided Design and Computer Graphics (CAD/Graphics)*. IEEE, 2013, pp. 439–440.
- [39] K.-M. Chen and S.-K. Wong, "Interactive sand art drawing using kinect," in *Proceedings of the 7th International Symposium on Visual Information Communication and Interaction*. ACM, 2014, p. 78.
- [40] Z. Chen, B. Kim, D. Ito, and H. Wang, "Wetbrush: Gpu-based 3d painting simulation at the bristle level," *ACM Transactions on Graphics (TOG)*, vol. 34, no. 6, p. 200, 2015.
- [41] A. Kurganov, G. Petrova et al., "A second-order well-balanced positivity preserving central-upwind scheme for the saint-venant system," *Communications in Mathematical Sciences*, vol. 5, no. 1, pp. 133–160, 2007.
- [42] J. C. Simo and T. J. Hughes, *Computational inelasticity*. Springer Science & Business Media, 2006, vol. 7.
- [43] N. Thurey, M. Muller-Fischer, S. Schirm, and M. Gross, "Real-time breaking waves for shallow water simulations," in *15th Pacific Conference on Computer Graphics and Applications (PG'07)*. IEEE, 2007, pp. 39–46.

- [44] B. Solenthaler, P. Bucher, N. Chentanez, M. Mller, and M. Gross, "SPH Based Shallow Water Simulation," in *Workshop in Virtual Reality Interactions and Physical Simulation "VRIPHYS" (2011)*, J. Bender, K. Erleben, and E. Galin, Eds. The Eurographics Association, 2011.
- [45] J. Kruger and R. Westermann, "Acceleration techniques for gpu-based volume rendering," in *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*. IEEE Computer Society, 2003, p. 38.
- [46] L.-Y. Wei, "Parallel poisson disk sampling," *ACM Transactions on Graphics (TOG)*, vol. 27, no. 3, p. 20, 2008.
- [47] O. Hungr, "Simplified models of spreading flow of dry granular material," *Canadian Geotechnical Journal*, vol. 45, no. 8, pp. 1156–1168, 2008.



Guoping Wang is currently a professor in Peking University, where he is also the director of Graphics & Interactive Technology Laboratory. He got B.S. and M.S. degree from Dept. of Mathematics, Harbin Institute of Technology in 1987 and 1990 respectively, and got Ph.D from Institute of Mathematics, Fudan University in 1997. He achieved the National Science Fund for Distinguished Young Scholars in 2009. His research interests include Virtual Reality, Computer Graphics, Human-Computer Interaction, and Multimedia. He is member of ISO/IEC JTC1/SC24 International Organization for Standardization.



Kuxin Zhu is with the School of Electronic and Computer Engineering, Peking University Shenzhen Graduate School, Peking University. His research interests include computer graphics and physical-based simulation.



Xiaowei He is currently a research associate professor at the HCI Lab of Institute of Software, Chinese Academy of Science(CAS). He received his B.S. and M.S. degrees from Peking University in 2008, 2011, respectively and Ph.D in 2017 from the Institute of Software, CAS. His research interests include computer graphics and physical-based simulation.



Sheng Li received both B.S. and M.S. in computer science from Shandong University in 1997 and 2000 respectively, and received his Ph.D. in 2005 from the Institute of Software, Chinese Academy of Science (CAS). Currently, he is an Associate Professor of the School of Electronics Engineering and Computer Sciences, Peking University. He works as a member of the Graphics and Interaction Lab. and publishes over 30 refereed papers in prestigious international conferences and journals. His research interests include virtual reality, computer graphics, physical simulation and animation, artificial intelligence. He is a member of ACM and IEEE.



Hongan Wang received the M.S. degree from Institute of Computing Technology Chinese Academy of Sciences, P.R. China. He received the Ph.D. degree from Institute of Software Chinese Academy of Sciences, P.R. China. He is currently a Research Professor in Institute of Software Chinese Academy of Sciences, P.R. China where he is the director in the Intelligence Engineering Lab. His research interests include real-time intelligence, human-computer interaction and user interface. He has published over 100 papers in RTAI and HCI fields, including IEEE RTSS, ACM CHI, JHCS, ACM IUI, ACM TIST, etc.