EUROGRAPHICS 2012 / C. Andujar, E. Puppo *Short Paper*

Solid-state Culled Discrete Element Granular Systems

S. R. Holladay[1] and P. Egbert[1]

[1]Brigham Young University

Abstract

Dry granular materials are common in computer animations, such as when a character drags a finger through sand or grabs a handful of wheat. However, simulating hundreds of thousands to millions of interacting particulates can take several minutes per frame or longer. More efficient granular material algorithms compromise visual accuracy due to granules' complex behavior and detail. To shorten simulation times without sacrificing visual fidelity, we simulate individual granules close to the material's surface and use implicit surfaces and constructive solid geometry boolean functions to cull away solid-state granules beneath the surface and only replace them where the simulation requires. This surface dynamically updates in response to collisions or exposure from the surface granules. Our technique selectively removes the majority of granules for significant speedups, while staying flexible enough to cull out and repopulate granules to respond to multiple external forces.

Categories and Subject Descriptors (according to ACM CCS): I.6.8 [Simulation and Modeling]: Types of Simulation—Discrete event

---

1. Introduction

Visual effects artists often need to simulate dry granular materials such as sand, wheat, dirt, or gravel, for computer-generated animations. However, visually accurate granular simulations are time and memory intensive. Even relatively small piles can contain millions of particles. Computing the interactions and frictional behavior between all granules can take several hours or even days. This high computation time, combined with large memory consumption, makes iteration of detailed simulations impractical.

Granular simulations are difficult to optimize while retaining the unique properties and behaviors of granular materials [BCPE94]. Granular materials can behave like a solid, a liquid, and a gas [FP08]. The solid state occurs when granules settle so that their irregular shapes "interlock" and cannot flow freely. The liquid state occurs when the granules separate enough to flow. When individual granules move independant of all others, they are in their gas state.

Current optimizations are efficient but lose granular behaviors or detail. When granular detail is needed, such as when sand is kicked up or a stick is dragged through dirt, a separate particle simulation has to be set up, run, and meticulously integrated with the optimized system to look like one material. Often, due to time crunches, those details are either left out or do not produce results that meet expectations.

We propose an automated system that simulates individual particulates for surface detail but is supported by a simpler simulation model below. Also, just as a granular system transfers between physical states, we represent the solid area as a simple triangular mesh, then automate the transfer to and from individual granules. We propose what those models are and how we can solve for the transfers.

1.1. Previous work

Discrete elements methods (DEM) [CS79] simulate granular materials by simulating every individual granule, but they are computationally expensive. Attempts to speed up DEM replace each granule with multiple connected spheres [BYM05] or clump granules into coarser granules in areas of less import [ATO09] , but the number of granules is still often too high for desirable simulation times or behavior.

Alternatively, fluid simulations model granular flow naturally and can be faster than DEM [ZB05]. [NGL10] extends [ZB05] and [AO11] modifies SPH fluids for more sand-like, non-Newtonian behavior. However, continuum models settle viscously into a solid state. Fluid methods can also lose

DOI: 10.2312/conf/EG2012/short/065-068

granular-level detail and behavior when they are in a solid or gas state, though [ZB05] isolates rigid portions of the simulation and [LD09] clusters rigid particles. We will apply the idea of isolating the rigid areas to DEM.

Height fields simulate buildup and erosion of granular surfaces [SOH99]. However, this technique is constrained to a 2D surface and does not have individual granular detail.

Recent work combines above methods into a hybrid model. For surface granular detail, [ZY10] replaces interior granules with the more efficient height field representation and dynamically covers it with DEM granules on the surface. However, it does not take into account external collisions and is constrained to the height field. We will expand on this idea of covering a vacant volume with individual granules.

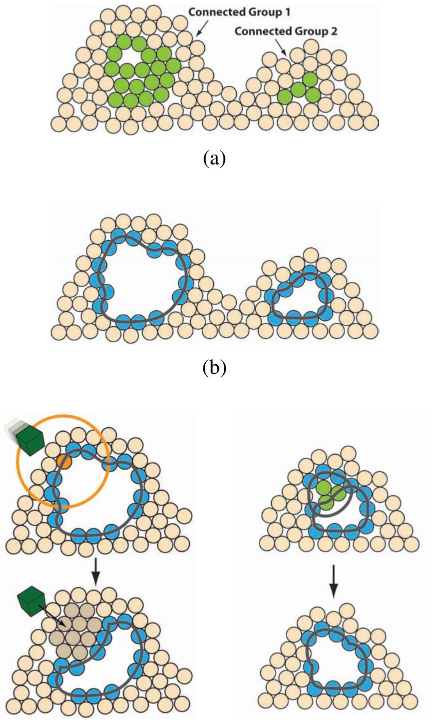
2. Methods

We automate the transition between individual granules and a culled solid state. Our method uses DEM on the surface but culls away interior granules that are in the solid state, deleting the buried granules and bounding each culled space with a rigid, fully-enclosed mesh. The DEM granules form a surface layer thick enough to retain the granular detail and motion we desire. The mesh constantly grows, shrinks, and moves in response to the surface granules. As a pile grows, so does the mesh. When the mesh is exposed or impacted, it carves away the local mesh and restores discrete granules to the carved area. We use constructive solid geometry (CSG) operations [LTH86] and implicit geometry to update the mesh. A layer of granules constrained to that mesh prevents exterior granules from falling through.

Our simulation algorithm has four parts per time step.

1. First, we run a timestep of the DEM simulation on the *exterior granules*, granules that are free to move independently, and interface granules.
2. Next, we delete *interior granules*, buried granules settled into a solid state (Figure 1a).
3. Third, we form *interface meshes*, rigid meshes around each hollow pocket from the deleted granules. These hollow meshes represent granular solid masses. *Interface granules*, granules neighboring the deleted granules, are constrained to their respective mesh (Figure 1b) to prevent DEM granules from falling into the vacant pockets.
4. Finally, where interface granules become exposed or impacted by a force exceeding a defined threshold, we carve into the local portion of the mesh and repopulate that space with exterior granules (Figure 1c).

Our process culls out the majority of granules on average by converting between the solid and DEM states, a solid-DEM coupling. This reduces the majority of time and memory spent on buried granules. Our method also responds to external forces by dynamically replacing granules where the action requires. Finally, we introduce a novel interface mesh between the DEM granules and empty pockets. It interacts and transforms with the outer granular mass without any constraint to height fields. The following sections give details for each substep.



(a)

(b)

(c) (d)

Figure 1: Illustrated cutout of our 3D simulation. a) Interior granules (green) are grouped based on contact. b) Each group of interior granules forms an interface mesh (gray) and constrains bounding granules (blue) to it. Interior granules are deleted. The hollow area represents a solid mass of granules. c) An interface granule (orange) detects an impact. The local mesh is carved away by CSG subtraction (orange sphere) and new exterior granules (darker tan) are repopulated into the carved space before the next simulation step. Previously existing exterior granules (tan) hide new granules from view. d) Where granules build up, interface meshes merge with a CSG union and constrain their interface granules to the mesh.

2.1. Culling interior granules

In our granular simulation, we cull out interior granules by *completely deleting* them. For a granule to be tagged as interior, it must have a similar velocity to its neighbors (rigidly bunched-together granules in the *solid state*) and be deeper than a threshold distance from the surface. Exterior granules are never frozen, only deleted if they become interior. This not only saves on time but also memory.

To compute a granule's distance to the surface, we generate an implicit mesh hugging the frozen granules (Figure 2a) by replacing each frozen granule with an implicit sphere, or metaball. Clumps of metaballs join into a single implicit mesh while granules separated from the mass have their own surface. Then we run a union operation over all the meshes to merge inner meshes with their outer mesh (Figure 2b). Once the meshes are formed, we extrude inward by a depth of at least two granule diameters, deep enough that the granule deletion will not be visible. Any granules inside these shrunken meshes are marked interior and deleted.

Our use of implicit surfaces gives us the ability to continually compare a granule's depth from any angle (not just in the y-direction, as with height maps), and we can cull granules from inside any shape of granular clumps.

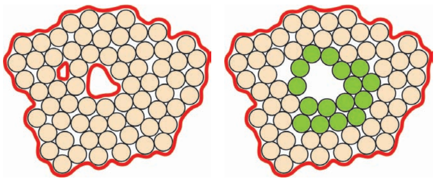

(a) (b)

Figure 2: a) Create an implicit mesh (red) out of frozen granules (tan). b) The union of surfaces retains the outer surface. Granules farther than a threshold distance are interior granules (green).

2.2. Creating or extending interface surfaces

After deleting interior granules, we must replace this deleted "solid" mass with a hollow interface surface to prevent exterior granules from caving in. Before deleting the interior granules, we group them into connected groups based on contact (Figure 1a). We replace each group of interior granules with metaballs to form an implicit *interface mesh*, making one mesh per connected group (Figure 1b). Next, we take each granule that was bounding, or colliding against, a deleted interior granule and lock its relative position to its respective mesh (Figures 1b and 3), resulting in a rigid shell of granules. Each implicit mesh combined with the granules constrained to it makes up our *interface surface*, each surface animating as one mass independent of the others. The interface surface contains granules for interacting with exterior granules and has an implicit mesh for calculating the repopulation step (Section 2.4).

If an interface granule becomes an interior granule, its group of interior granules form a new implicit mesh that merges with that granule's existing interface mesh by applying a CSG union (Figure 1d). In this way, the interface granules act as "sensors" to determine if a new interface surface needs to merge with an existing one.

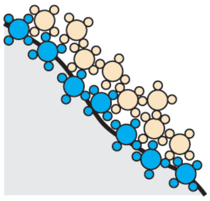


Figure 3: Free-moving exterior granules (tan) are naturally supported by locked interface granules (blue).

2.3. Rigid-body simulation

We treat individual granules as discrete, multi-sphere rigid bodies [BYM05] simulated with an iterative LCP solver. Since we completely delete static granules, memory is kept within reasonable limits. All granules, whether exterior or interface, simulate together in a single time step per frame. Even with two state representations, the simulation runs as a single rigid body solver without a need for special cases.

2.4. Repopulating granules

Interface granules use impact information to determine when and where to make the state transition to DEM. When an object impacts an interface granule hard enough to penetrate culled granules, we carve away the local portion of the mesh and repopulate the carved volume with exterior granules (Figure 4). To make this transition, the interface granule chips off without affecting the momentum of the interface surface and signals the surface's mesh to carve away a local area by CSG subtracting a sphere whose radius is proportional to the impact intensity. Interface granules inside that sphere are also deleted and new ones scattered onto the carved mesh. To repopulate granules into the carved-out volume, we fill that volume with a 3D grid of exterior granules, giving us the discrete granule behavior we desire. The new granules will be buried under a layer of exterior granules when created, so no visual or popping artifacts are seen in the transition (Figure 1c).

We also repopulate exterior granules if the interface surface is close to being exposed. There should always be a layer of exterior granules covering the interface surface to retain the granular detail we are aiming for.

3. Results

We simulate a falling mass of granules (Figure 5) and compare our results against a brute-force DEM to demonstrate the increased efficiency of our algorithm (Table 1). In the brute-force method we simulate every single grain without any culling.

At the start of the simulation, with the granular pile suspended in the air, our system was significantly faster than brute force because the interior of the entire mass is culled

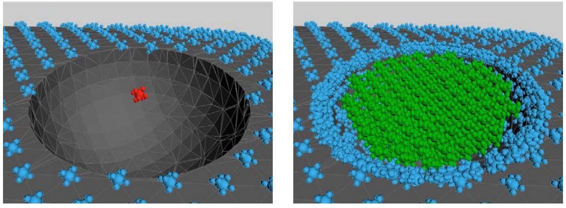

(a) (b)

Figure 4: Beneath a layer of exterior granules, a) an impacted interface granule (red) triggers interface mesh to be carved and b) exterior granules to be repopulated within the volume of the CSG subtraction.

	Full DEM Sim	Granular Culling
Total time	7 hr. 11 min.	31.8 min.
Min frame time	6.3 min.	0.12 min.
Avg frame time	6.8 min.	0.74 min.
Min granule count	15,600	1,442

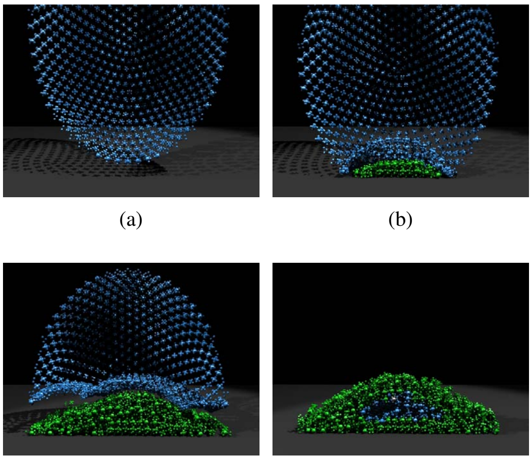
Table 1: Comparison of simulation times between a full volume of DEM granules versus our culling algorithm. We used the same falling oval shape for both.

away. The granular mesh had 1442 granules and averaged 7.51 seconds per frame. In comparison, the full volume of granules added up to about 15,600 granules and took 6.28 minutes per frame. Our method does lose volume in the transitions, but we retain the visual behavior of DEM.

4. Conclusions and Future Work

Granular systems need an efficient, automated method to include discrete granules. We have a novel solution for transitioning to and from discrete granules for significant performance improvement of DEM. Our solution is novel in the way it builds an interface mesh that adaptively transitions between granules and solid meshes in response to buildup, exposure, and collisions. It dynamically culls out and replaces granules in both static and moving masses of granules.

We have introduced a multi-state model where granule representations are simplified based on the current state of the granules. Granules in a solid state are culled where possible. Granules in the liquid and gaseous states are simulated using DEM techniques. In the future, fluid simulations could be inserted for the liquid state. Our work leads to artist-friendly DEM granular simulations. Simulation time is dramatically reduced, thereby giving the artist more time and flexibility in the animations he or she can run.


(a) (b)

(c) (d)

Figure 5: A cutout of falling granules in various time steps. a) Interface granules (beneath unshown exterior granules). b) Exterior granules repopulate local to impact. c) Granules continue to repopulate. d) Granules begin to cull out of the center where they settle again. The layer of external granules is thick enough that the surface can still flow. The granular mesh inbetween has minimal thickness of one granule.

References

[AO11] ALDUÁN I., OTADUY M. A.: SPH granular flow with friction and cohesion. In *Eurographics/ACM SIGGRAPH Symposium on Computer Animation* (2011). 1

[ATO09] ALDUÁN I., TENA A., OTADUY M. A.: Simulation of high-resolution granular media. In *Proceedings of Congreso Español de Informática Gráfica* (2009). 1

[BCPE94] BOUCHAUD J. P., CATES M. E., PRAKASH J. R., EDWARDS S. F.: A model for the dynamics of sandpile surfaces. *J. de Physique* (1994), 1383Ű1410. 1

[BYM05] BELL N., YU Y., MUCHA P. J.: Particle-based simulation of granular materials. In *ACM SIGGRAPH/Eurographics Symposium on Computer Animation* (2005), pp. 77–86. 1, 3

[CS79] CUNDALL P. A., STRACK O. D. L.: A discrete numerical model for granular assemblies. *Geotechnique 29*, 1 (1979). 1

[FP08] FORTERRE Y., POULIQUEN O.: Flows of dense granular media. *Annual Review of Fluid Mechanics 40*, 1 (2008), 1–24. 1

[LD09] LENAERTS T., DUTRÉ P.: Mixing fluids and granular materials. *Computer Graphics Forum 28*, 2 (2009), 213–218. 2

[LTH86] LAIDLAW D. H., TRUMBORE W. B., HUGHES J. F.: Constructive solid geometry for polyhedral objects. In *Computer Graphics (Proceedings of SIGGRAPH 86)* (1986), ACM, pp. 161–170. 2

[NGL10] NARAIN R., GOLAS A., LIN M. C.: Free-flowing granular materials with two-way solid coupling. In *ACM Transactions on Graphics* (2010), ACM SIGGRAPH Asia. 1

[SOH99] SUMNER R. W., O'BRIEN J. F., HODGINS J. K.: Animating sand, mud, and snow. *Computer Graphics Forum 18* (1999), 17–26. 2

[ZB05] ZHU Y., BRIDSON R.: Animating sand as a fluid. *ACM Transactions on Graphics 24* (2005), 965–972. 1, 2

[ZY10] ZHU B., YANG X.: Animating sand as a surface flow. In *Proceedings of Eurographics 2010 (Short Paper)* (2010). 2