Multi-Scale Modeling and Rendering of Granular Materials

Pramook Khungurn

October 28, 2015

This document is created as I read the paper "Multi-Scale Modeling and Rendering of Granular Materials" [Meng et al., 2015].

1 Overview

- The parameter presents a stochastic model for grains.
- It renders grains with three separate algorithms:
  - Explicit path tracing (EPT) for the finest scale.
  - Volume path tracing a homogeneous medium (VPT).
  - Diffusion approximnation (DA).

  The paper describes a criteria for switching between them.
- VPT and DA requires suitable parameters. The paper describes an algorithm for deriving these parameters without performing a per-scene precomputation.

1.1 Stochastic Granular Model

- The input for the model are:
  - A surface of the boundary of the aggregate object (i.e., the macrogeometry).
  - A model for the geometry and material properties of individual grains.
  - The size and packing rate of the grains.
  - Mixing weights if there are multiple types of grains.
- Assumptions:
  - Grains are randomly positioned and oriented.
  - Grains are non-overlapping.
- Grain arrangement via tiled sphere packing.
  - They say each grain is contained in a non-overlapping sphere.
  - To get the actual arrangement of grains, they procedurally pack the spheres instead of the grains.
    - This limits the packing rate for non-spherical grains, but they say this simplifies things.
  - The building block of sphere packing is a *tiled*, which is a cuboid of non-overlapping spheres.
  - The tiles are generated by the algorithm of Skoge et al. [Skoge et al., 2006] and are repeated throughout the volume.

- For each sphere,
  - a grain type is chosen randomly according to the mixing weights specified by the user, and
  - the grain's orientation is randomized.

  To ensure deterministic appearance, the paper sets the random seed for each grain to be based on the tile ID and the sphere ID.

1.2 Explicit Path Tracing

- At the finest level of detail, the paper traces paths through the geometry directly.
- The boundary mesh is voxelized.
  - Each voxel corresponds to a tile described in the last section.
  - Each voxel is either "fully outside", "partially inside", or "fully inside."
- When a ray hits a voxel that is at least partially inside, the path tracer intersects the ray with the bounding spheres inside the voxel.
- When intersecting a bounding sphere in a voxel that is partially inside, the path tracer needs to determine whether the sphere is inside or outside the shape. It does so by tracing the ray from the center of the sphere and see if the ray intersects the bounding surface on the inside or not. If so, it is in; otherwise, it is not.
- After the hitting bounding sphere is identified, the grain inside is instantialized. The ray is then transformed to the grain's local coordinate system, and then intersect with the ray.

1.3 Volumetric Path Tracing

- This is basically the standard volume path tracing in a homogeneous medium.
- The parameters needed for this are:
  - the extinction coefficient σ_t,
  - the scattering albedo α_s, and
  - the phase function Φ (isotropic).
- Unlike DPT, VPT can do emitter sampling, which significantly reduce variance.

1.4 Diffusion Approximation

- VPT can be costly for long paths.
- To make the computation even cheaper, they transition to a diffusion approximation.
- The paper switches to DA by sampling a location on the boundary mesh and estimating the diffusion transport.
- Components:
  - A technique based on [Li et al., 2005].
  - d'Eon et al.'s improved diffusion model [D'Eon and Irving, 2011].
  - A Monte Carlo integration scheme [Habel et al., 2013].
  - Multipole expansion to account for finite thickness [Donner and Jensen, 2005].

  - An original virtual source placement procedure.

- The parameters for this step is the reduced medium parameter (σ_t' and α_s'), which are obtained from the parameters of VPT using Jensen's et al.'s method in the dipole paper [Jensen et al., 2001].

2 RTE Parameters

We need to derive RTE parameters that are consistent with the behavior of EPT. This is non-trivial because the scattering elements are not points. The paper introduces a light transport model called the *teleportation transport* (TT), in order to account for non-point scatterings. Then, it uses the model to derive RTE parameters.

2.1 Teleportation Transport Model

- The TT model is created to characterize paths generated by EPT in granular material.
- The TT model consists of two steps occur in alternation.
  1. **Inter-grain transportation** decides how far along a ray to move before the next interaction with a grain bounding sphere.
  2. **Intra-grain transportation** deals with interaction between the ray and the grain inside and "transporting" the ray to the point where it exits the bounding sphere.

2.1.1 Inter-Grain Transport

- We have to determine the free-flight distances from one sphere to the next.
- An idea that we might use: just tabulate this distribution like Moon et al. did [Moon et al., 2007].
- To avoid precomputation, they use a model by Dixmier [Dixmier, 1978]:

$$p_b(z) = \sigma_b e^{-\sigma_b z}, \text{ with } \sigma_b = \frac{3}{4R}\frac{f}{1-f}.$$

  Here, z is the free-flight distance, R is the radius of the sphere, and f is the packing rate of the sphere.

2.1.2 Intra-Grain Transport

- The paper defines the *teleportation scattering distribution function* (TSDF) $S(\mathbf{x}_i, \omega_i \to \mathbf{x}_o, \omega_o)$ where $\mathbf{x}_i$ and $\mathbf{x}_o$ are points on the sphere and ω_i and ω_i are directions.
- This function can be computed by tracing rays through the sphere and having the ray interact with the grain.
- The paper avoids using this function and tries to derive the RTE parameters from it instead.

2.2 RTE & Diffusion Parameters

- **Phase function and albedo**
  - Due to symmetry of the random grain rotations, the phase function will depend solely on the cosine of the deflecting angle $\cos\theta = \omega_i \cdot \omega_o$.
  - The paper tabulate this 1D function.
  - The integral of the 1D distribution over the outgoing directions of the unit hemisphere is the effective albedo σ_s.

- **Combined free flight distribution**
  - We need to extend the spherical free-flight distance $\lambda_b = 1/\sigma_f$ with what can happen inside the sphere.
  - There are two cases:
    1. The ray that goes into the sphere might not hit the grain inside and just straight through the sphere.
    2. The ray interacts hits the grain and leaves the sphere.
  - Let β be the probability that the ray that goes into the sphere does not hit anything. This can be estimated by just tracing rays. This is called the *hit probability*.
  - Let the λ_δ be the mean length of the portion of unscattered rays inside the sphere. This can be computed by simulation as well.
  - Here's what can happens after a ray leaves a sphere before it hits something inside the sphere.
    * It travels a distance of λ_b and then hits something inside.
      The contribution of this case is $\beta\lambda_b$.
    * It travels a distance of λ_b, hits nothing inside so travels distance λ_δ, and then travels a distance of λ_b again, before hitting something inside.
      The contribution of this case is $\beta(1-\beta)(2\lambda_b + \lambda_\delta)$.
    * It travels, not hit, travels, not hit, travels, and then hits.
      The contribution of this case is $\beta(1-\beta)^2(3\lambda_b + 2\lambda_\delta)$.
  - So, the distance before the ray hits something is:

$$\lambda_\beta = \beta \sum_{i=1}^{n} (1-\beta)^i [i(\lambda_b + \lambda_\delta) + \lambda_b] = (\lambda_b + \lambda_\delta)\frac{1-\beta}{\beta} + \lambda_b.$$

  - Once the ray hits something inside, it has probability α_s of coming out. With simulation, we can estimate the mean teleport vector $\mathbf{x}_o - \mathbf{x}_i$. The length of which we denote λ_s.
  - The mean free-flight length is then given by:

$$\lambda_t = \lambda_\beta + \alpha_s \lambda_s.$$

  - The probability distribution of the free-flight distance is then given by:

$$p_t(z) = \sigma_t e^{-\sigma_t z}$$

    where $\sigma_t = 1/\lambda_t$.

- The diffusion prameters are given by:

$$\begin{aligned}
\sigma_s' &= (1-g)\alpha_s\sigma_t \\
\sigma_t' &= \sigma_s' + (1-\alpha_s)\sigma_t \\
\alpha_s' &= \sigma_s'/\sigma_t',
\end{aligned}$$

  where g is the mean scattering cosine computed from the phase function.

3 Switching between Rendering Techniques

- Rays are assume to start outside the granular material volume.
- At first, renderings are performed with EPT.

- **EPT → VPT**
  - Spawn a bundle of $N = 16$ rays per pixel. These rays are traced in lock-step.
  - At each bounce k, suppose there are N_k rays left. (Ray can be terminated by Russian roulette and other reasons.) Compute the standard deviation σ_k of N_k of the vertex positions of the N_k rays.
  - The algorithm switches to VPT when:

    $$\sigma_k > \tau \frac{N_k}{N}$$

    where τ is a user-specified multiple of the maximum grain radius r.
  - The paper uses $\tau = 4$.
- **VPT → DA**
  - Measure the minimum distance between the vertex position $\mathbf{x}_k$ at the kth bounce to the boundary surface.
  - The paper switches to DA once the distance above is greater than $\min(1/\sigma_t', 0.5/\sigma_{tr})$. (See the definition of this in the Stam paper [**?**].)
  - The criteria above allows the algorithm to switch to DA more quickly in material with lower albedo.
- **Russian roulette**
  - EPT converges much more slower than VPT and DA. The latter two algorithms converges very fast, and not so many samples are needed.
  - So, we can cut time spending on VPT and DA by introducing Russian roulette when we switch.
  - The paper introduces the probably of continuing with VPT and DA, which they call P_a.
  - The optimal P_a is scene dependent, but the paper introduces a way to compute it with some precomputation.
  - The paper renders an image 1% of the original size with the same number of pixels as the original image.
  - In this rendering, for each pixel (x, y) in the small image:
    - the sample variance of the low order (EPT) contributions $V_L(x, y)$,
    - the sample variance of the high order (VPT and DA) contributions $V_H(x, y)$,
    - the CPU time for computing the low order contributions $t_L(x, y)$, and
    - the CPU time for computing the high order contributions $t_H(x, y)$.
  - The variance of the combined image as a function of the number of samples n and the acceptance rate P_a is:

    $$V \approx \frac{1}{n}\left(V_L + \frac{V_H}{P_a}\right)$$

    where V_H and V_L are averages of $V_H(x, y)$ and $V_L(x, y)$ across the image.
  - The total time needed to render is:

    $$t = n(t_L + P_a t_H)$$

    where t_L and t_H are averages across the image as well.

– Solving for P_a that miniminzes t, we have :

$$P_a = \sqrt{\frac{V_H t_L}{V_L t_H}}.$$

(It is unclear what constraint they optimize against. Do they set V to be a specific value and use the constraint $V \leq (V_L + V_H/P_a)/n$?)

References

[D'Eon and Irving, 2011] D'Eon, E. and Irving, G. (2011). A quantized-diffusion model for rendering translucent materials. In *ACM SIGGRAPH 2011 Papers*, SIGGRAPH '11, pages 56:1–56:14, New York, NY, USA. ACM.

[Dixmier, 1978] Dixmier, M. (1978). Une nouvelle description des empilements alatoires et des fluides denses. *Le Journal de Physique*, 39:873–895.

[Donner and Jensen, 2005] Donner, C. and Jensen, H. W. (2005). Light diffusion in multi-layered translucent materials. In *ACM SIGGRAPH 2005 Papers*, SIGGRAPH '05, pages 1032–1039, New York, NY, USA. ACM.

[Habel et al., 2013] Habel, R., Christensen, P. H., and Jarosz, W. (2013). Photon beam diffusion: A hybrid monte carlo method for subsurface scattering. In *Proceedings of the Eurographics Symposium on Rendering*, EGSR '13, pages 27–37, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

[Jensen et al., 2001] Jensen, H. W., Marschner, S. R., Levoy, M., and Hanrahan, P. (2001). A practical model for subsurface light transport. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 511–518, New York, NY, USA. ACM.

[Li et al., 2005] Li, H., Pellacini, F., and Torrance, K. E. (2005). A hybrid monte carlo method for accurate and efficient subsurface scattering. In *Proceedings of the Sixteenth Eurographics Conference on Rendering Techniques*, EGSR '05, pages 283–290, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

[Meng et al., 2015] Meng, J., Papas, M., Habel, R., Dachsbacher, C., Marschner, S., Gross, M., and Jarosz, W. (2015). Multi-scale modeling and rendering of granular materials. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 34(4).

[Moon et al., 2007] Moon, J. T., Walter, B., and Marschner, S. R. (2007). Rendering discrete random media using precomputed scattering solutions. In *Proceedings of the 18th Eurographics Conference on Rendering Techniques*, EGSR'07, pages 231–242, Aire-la-Ville, Switzerland, Switzerland. Eurographics Association.

[Skoge et al., 2006] Skoge, M., Donev, A., Stillinger, F. H., and Torquato, S. (2006). Packing hyperspheres in high-dimensional euclidean spaces. *Physical Review E*, 74(4):041127.