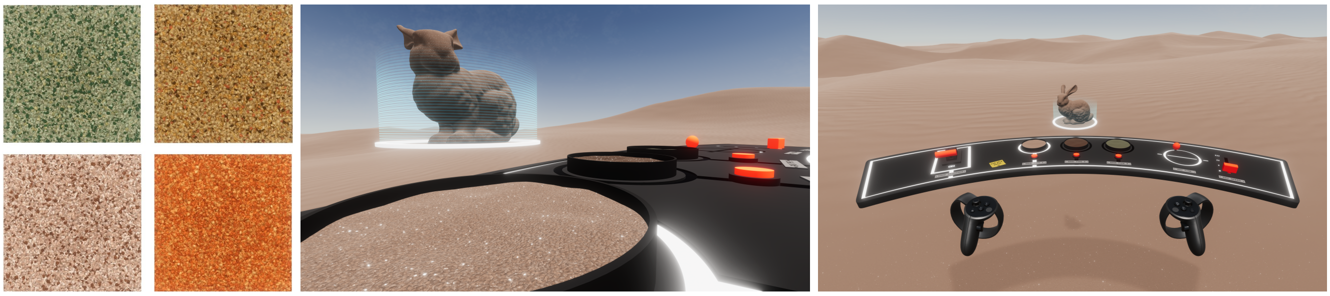
International Conference on Artificial Reality and Telexistence
Eurographics Symposium on Virtual Environments (2022)
R. Kondo and T. Teo (Editors)

Demo

Rendering Glinty Granular Materials in Virtual Reality

Nynne Kajs[1], Mikkel Gjøl[2], Jakob Gath[2], Henrik Philippi[3,1], Jeppe Revall Frisvad[1], and J. Andreas Bærentzen[1]

[1]Technical University of Denmark
[2]Playdead, Denmark
[3]Luxion, Denmark



(a) (b) (c)

Figure 1: *We present a method for rendering of glinty granular materials at multiple scales in virtual reality (VR). To showcase our method, we render different types of sand under different skies in VR. (a) We enable the user to select parameters and procedurally generate different sand material textures. (b) A closeup of the sand as it appears in our demo. Our method renders explicit sand grains and glints at close distances. (c) Overview of the VR demo that sets the user in a desert environment with a workbench enabling control of relevant parts of the scene. At greater distances, we transition to shading using a BRDF rather than explicit microscale geometry.*

Abstract

Highly realistic rendering of grainy materials like sand is achievable given significant computational resources and a lot of time for the rendering of each frame. In an interactive virtual environment, we cannot afford such luxuries. Frame rates must be kept high and precomputation should be kept at a level that does not limit the interactivity. We propose a system for editable procedural generation of sand appearance and demonstrate interactive virtual reality (VR) inspection of the generated sand under different skies. Our method enables stable real-time rendering of the glinty appearance that granular materials exhibit as a function of observer distance. This enables simultaneous nearby and distant inspection of the material.

CCS Concepts

• ***Computing methodologies*** → ***Virtual reality;***

1. Introduction

Real-time rendering of glinty granular materials is a challenge because their appearances vary greatly depending on the distance to the observer. At a close distance, individual granules and their optical properties are discernible while the material exhibits an almost uniform appearance when observed from afar. Glinty granular materials consist of specular granules that produce characteristic highlights, namely glints, observable only at certain distances. Often, glints are replicated for real-time applications using a bidirectional reflectance distribution function (BRDF) [CSDD20]. While an approach like this provides stable and convincing glints, it cannot be used to visually represent the underlying granular structure of the material. This is a shortcoming especially for virtual reality (VR) applications, as the user is free to inspect the materials even at extremely close distances. Other approaches achieve real-time multi-scale rendering of granular materials by transitioning from mesh-based to point-based rendering of granules [MB20]. While rendering explicit geometry at close scale avoids the shortcomings of a BRDF, it is unknown if such a method can be used to produce anti-aliased glints in VR for granular materials. In this demonstrator, our objective is multi-scale VR rendering of visually convincing granular materials with editable appearance. We aim at stable glint behavior and no aliasing issues.

2. Method

We procedurally generate sand textures and use multisampling, mipmapping, and bloom effects to render glinty sand materials in VR (Figure 1). Our demo enables users to generate sand textures

DOI: 10.2312/egve.20221304

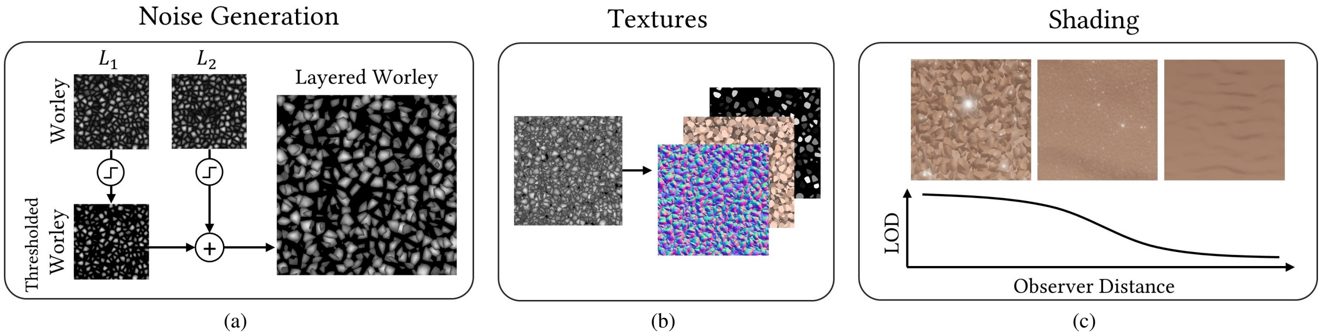


Figure 2: *Overview of our method. (a) We generate a texture of grains in layers ($L_1, L_2, \ldots, L_N$) from top (L_1) to bottom (L_N) using thresholded Worley [Wor96] noise. The threshold leaves space between the grains for the lower layers to fill. (b) When generating the layered noise, we save information about sand grain normals, colors, and degree of specularity in textures. (c) As the distance to the observer increases, we transition from Phong [Pho75] shading based on the generated textures to shading using a BRDF [Kaj22] and macroscopic surface normals.*

to their liking using tileable, layered Worley [Wor96] noise (Figure 2a). Each layer uses its own random seed and is thresholded to generate space between the sand grains. We draw layers from top to bottom using a write mask so that lower layers only draw into the gaps of higher layers. The result is a texture resembling stacked sand grains. We get sand grain normals from the gradients of the noise. During noise generation, colors are assigned to each sand grain using a pseudo-random process, and the brightness of the colors decreases for each layer we move down to model that less light reaches the lower layers. Finally, each sand grain in the upper layer is assigned a random intensity value used to define its degree of specularity. Information about normals, colors, and specularity are saved in textures (Figure 2b). The user controls colors, random seed, threshold level (sand grain roundness), and the number of layers N. Since the threshold affects the horizontal spacing between grains, more layers are required for a higher threshold.

At short observer distances, we use the sand grain information saved in textures in a simple Phong [Pho75] shading. The highlights of this shading produce the glints, and the textures are multisampled to avoid noisy highlights. Mipmapping is used to ensure that textures are sampled appropriately for midrange observer distances. Without mipmapping, glints become smaller than a pixel and aliasing occurs. However, the mipmapping process excessively blurs the glints to a point where they are barely visible. To combat this issue, highlights are multiplied by a large value dependent on the observer distance and HDR and bloom effects are used to enhance the glints. At greater observer distances, glints are no longer visible for real sand which appears uniform. To replicate this observation, we transition to using a BRDF and the macroscopic surface normal at greater distances using a simple sigmoid function (Figure 2c). We selected the sigmoid function parameters (shift and scale), so that the change in sand appearance would resemble the transition observed for real sand. We add macroscale details such as sand ripples using normal textures applied to the macro normal. This method does not ensure cohesion across scales. However, it is assumed that from a perceptual point of view, physically accurate cohesion is not essential for convincingly rendering the sand.

We rely on previous work [Kaj22] to include the microscale shading properties of sand grains when shading at macroscale. In this previous work, a suitable local macroscale shading model was found for sand through comparison with path tracing while considering the importance of diffuse, specular, and transmissive shading properties as well as direct and indirect lighting.

The VR demonstration of our method is made in Unity (https://unity.com/) for Oculus Rift S (https://www.oculus.com/rift-s/). The demo sets the user in a desert environment, and enables the user to control aspects of the sand rendering in the environment, such as choosing between three different sand materials and comparing the approach to simple Lambertian shading. The user can also control the time and the speed of the day to observe the sand material under varying lighting conditions. Additionally, we present a Unity editor demo that lets the user edit and procedurally generate sand textures to use for the sand material. The two demos are available from our project repository https://github.com/NynneKajs/glinty-granular-materials-vr.

Acknowledgements. This project received funding from Danmarks Erhvervsfremmebestyrelse (Danish Board of Business Development) through Vision Denmark and from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 956585 (PRIME). The involved companies, Playdead and Luxion, acted in a supervisory role. Thus, this project does not reflect any current work of the companies and can be shared openly to the benefit of the research community and other businesses.

References

[CSDD20] Chermain X., Sauvage B., Dischler J., Dachsbacher C.: Procedural physically based BRDF for real-time rendering of glints. *Computer Graphics Forum 39* (October 2020), 243–253. 1

[Kaj22] Kajs N.: *Real-Time Rendering of Granular Materials*. Master's thesis, Department of Applied Mathematics and Computer Science, Technical University of Denmark, 2022. 2

[MB20] Michel É., Boubekeur T.: Real time multiscale rendering of dense dynamic stackings. *Computer Graphics Forum 39*, 7 (October 2020), 169–179. 1

[Pho75] Phong B. T.: Illumination for computer generated pictures. *Communications of the ACM 18*, 6 (June 1975), 311–317. 2

[Wor96] Worley S.: A cellular texture basis function. In *Proceedings of SIGGRAPH '96* (August 1996), ACM, p. 291–294. 2