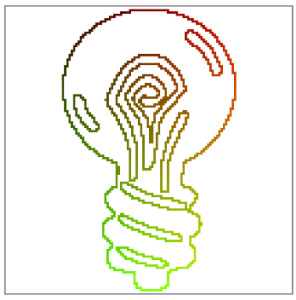
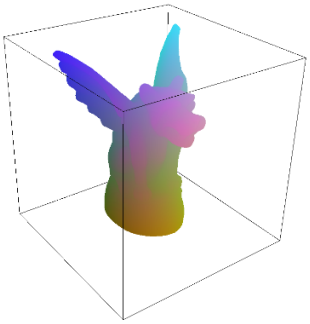
Spatial Hashing on the GPU



Sparse 2D data
n=1381 pixels in 128^2 image

Hash table H
m=38^2 (=1444)

Offset table Φ
r=18^2



Sparse 3D data
n=41,127 voxels in 128^3 volume

Hash table H
m=35^3 (=42,875)

Offset table Φ
r=19^3

Description :

Hashing is widely used in high-performance computing to store sparse tables. Rather than allocating memory for a whole potentially huge table a small array of lists is created and keys for accessing the original table are hashed into indices of the smaller array. But typical hashing mechanisms have been designed for sequential CPUs rather than highly parallel processors like GPUs.
Perfect Spatial Hashing [1] is a very powerful method for GPU-friendly hasing of data in 2D or 3D grids which features many applications in computer graphics like efficient vector graphics rendering or voxel model coloring and collisions detection. This project is an opportunity to learn about the specifics of GPU progamming by carefuly implementing and testing spatial hashing.

Nombre d'élèves : 3

Contraintes dures de développement :
- C++ / OpenGL

Difficulté :
- mathématique : low/medium
- implémentation : high

Encadrement :
Élie Michel, elie.michel@telecom-paristech.fr
Merci de lire l'article de référence avant de prendre contact.

[1] Perfect Spatial Hashing, Lefebvre and Hoppe, 2006