Fast Sphere Packing with Adaptive Grids on the GPU

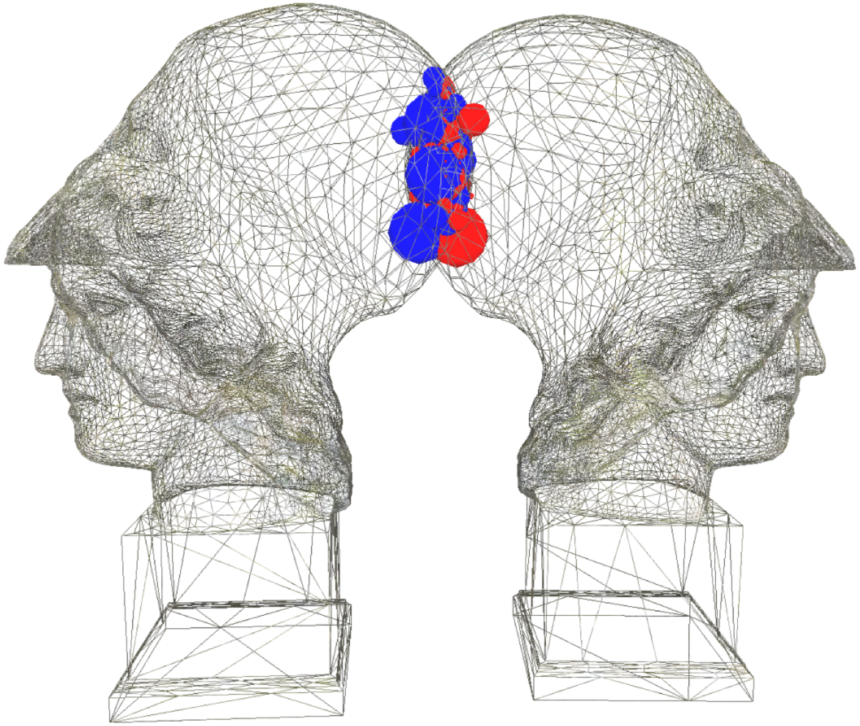
Jörn Teuber, René Weller, Gabriel Zachmann, Stefan Guthe

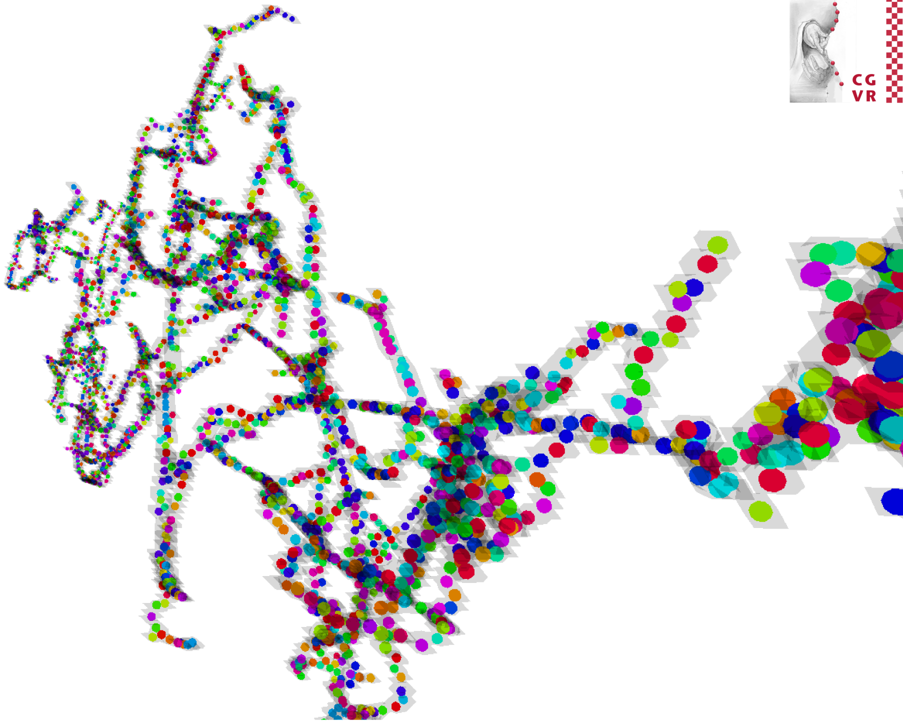
Clausthal University, Germany

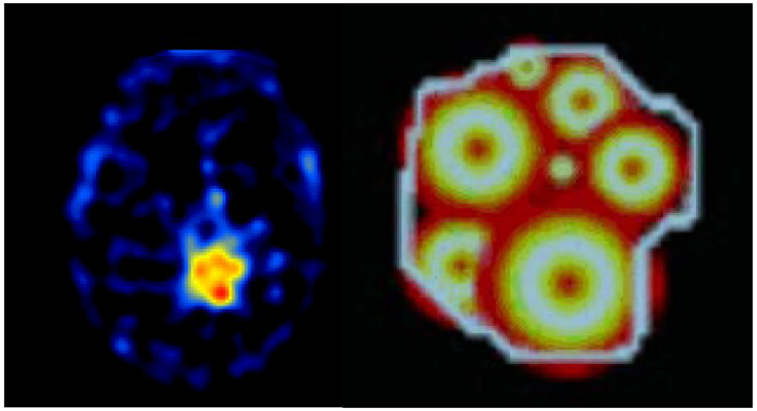
University of Bremen, Germany

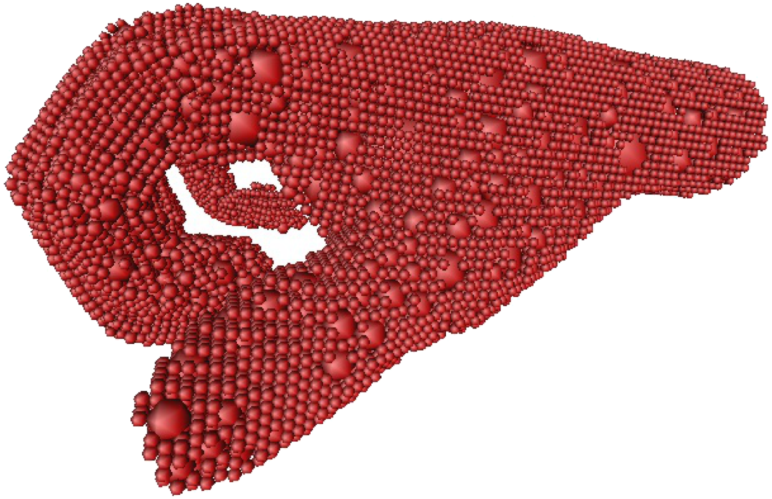
GI VR/AR ’13, Sep 2013, Würzburg, Germany

Motivation

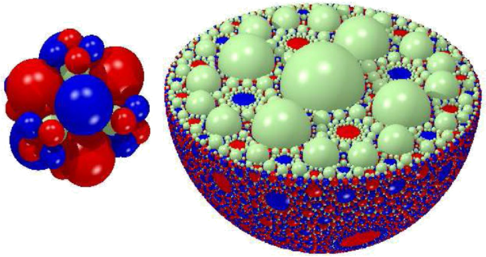




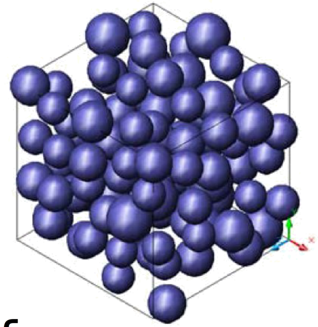




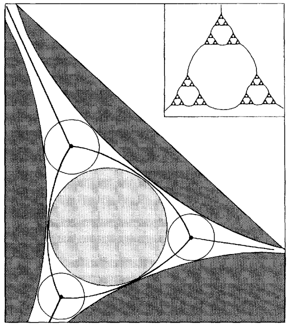
Previous Work



- Dense packings
  - H.J. Herrmann, R. Makmoodi Baram and M. Wackenhut, 2006



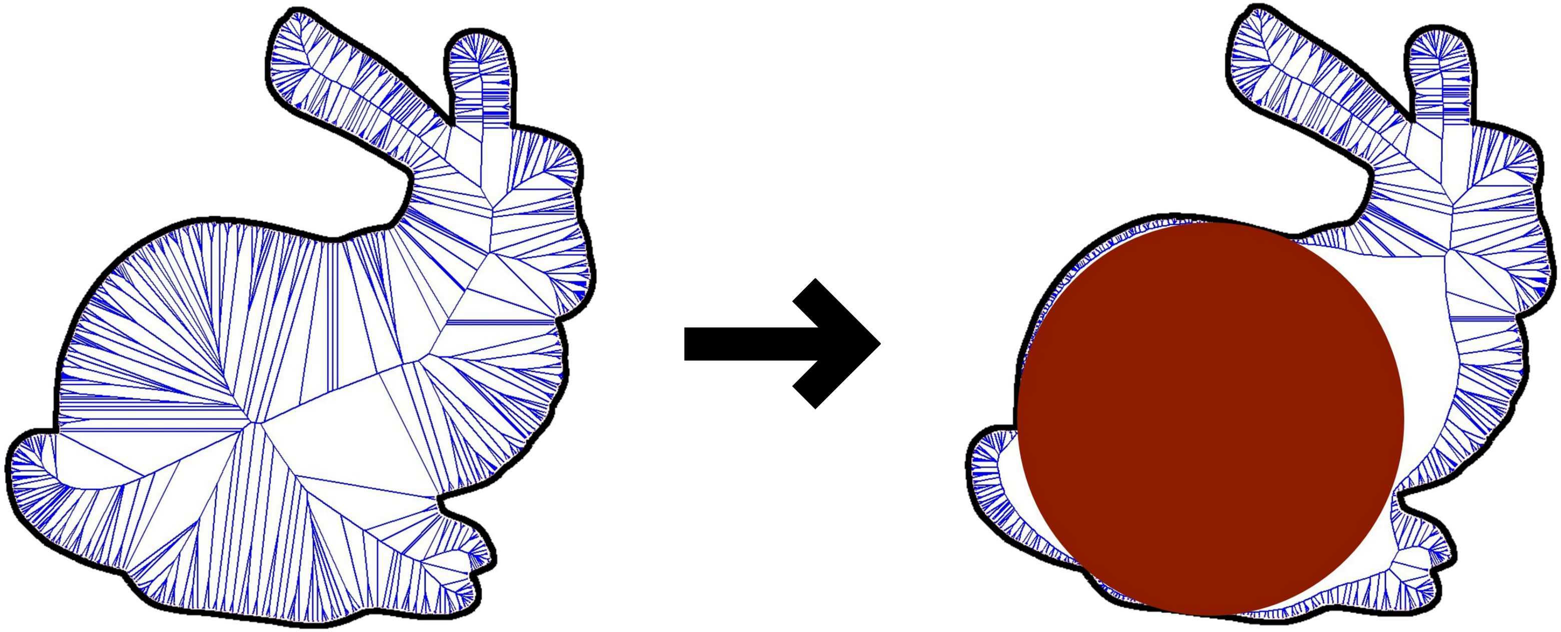
- Fractal Properties of Dense Packing of Spherical Particles
  - Adil Amirjanov and Konstantin Sobolev, 2006
- Three-Dimensional Apollonian Packing as a Model for Dense Granular Systems
  - V. Anishchik and N. N. Medvedev, 1995



- Protosphere: A GPU-Assisted Prototype Guided Sphere Packing Algorithm for Arbitrary Objects
  - R. Weller and G. Zachmann, 2010

Protosphere Recap

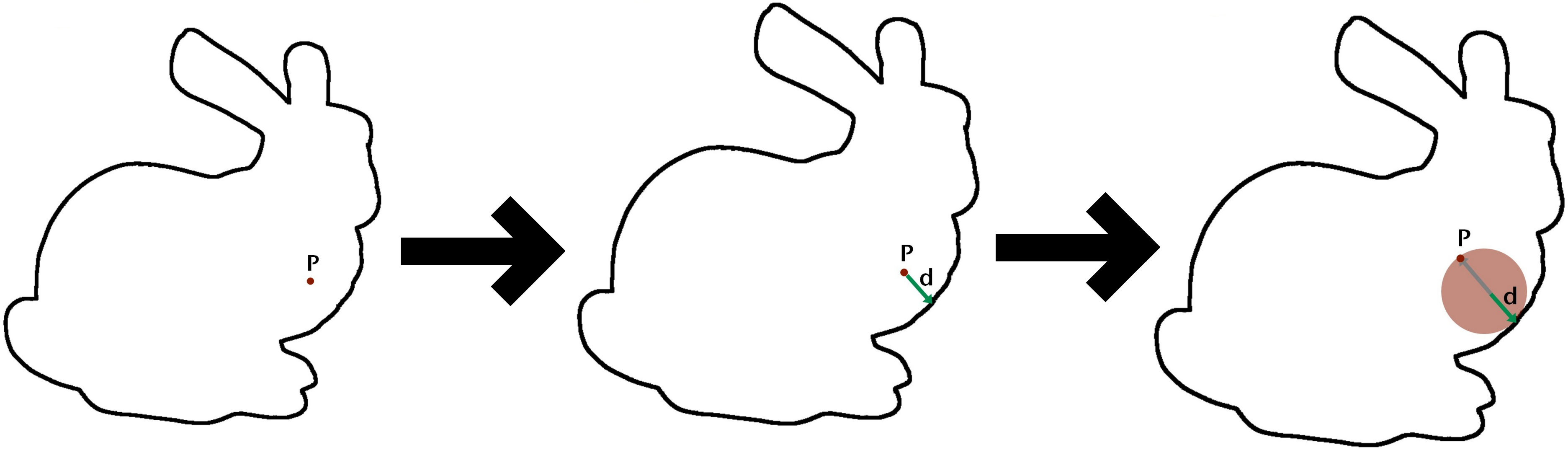
- A greedy approximation of Voronoi nodes



Protosphere Recap

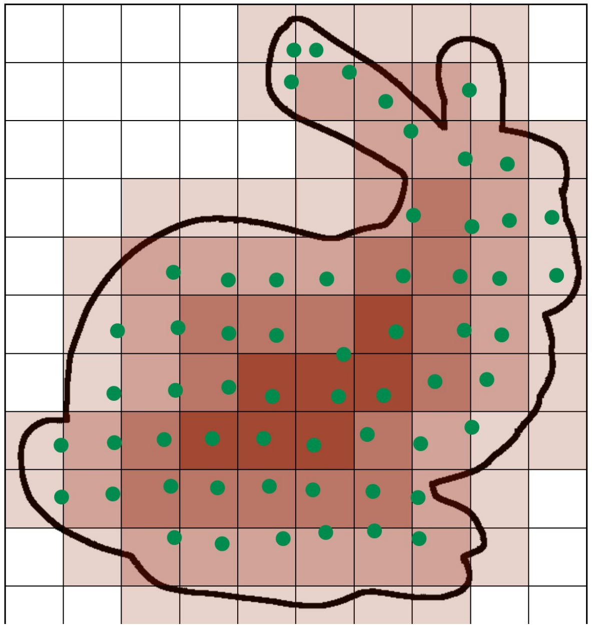
- Method:

  1) Place Prototype inside object
  2) Find closest point on objects hull
  3) Move prototype away from closest point



Protosphere Recap

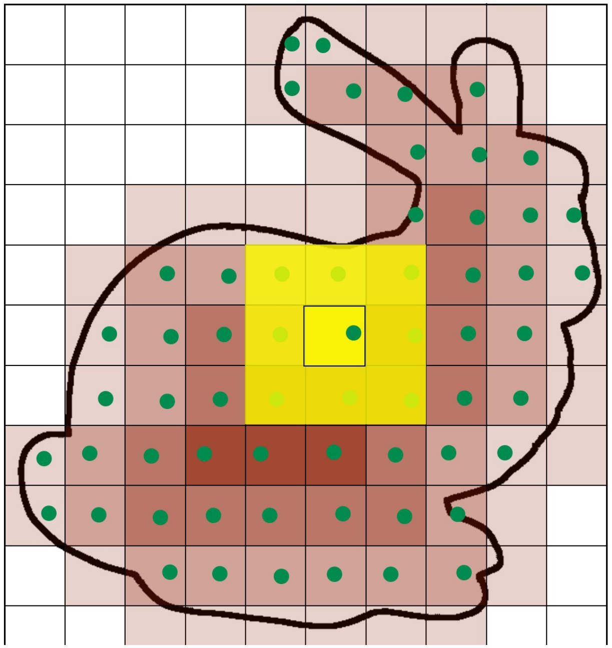
- Parallisation: Uniform Grid



- Every cell stores:
  - References to all triangles intersecting it
  - References to all spheres intersecting it
  - Discrete distance to the next cell with triangles or spheres

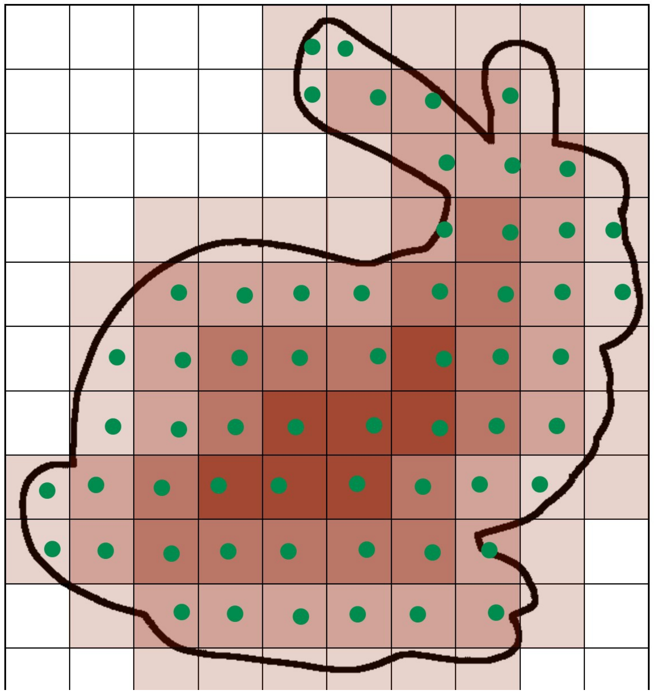
Protosphere Recap

- Naiv parallelisation: one core for every prototype
- Actual parallelisation: one core for every cell in the discrete distance to each prototype

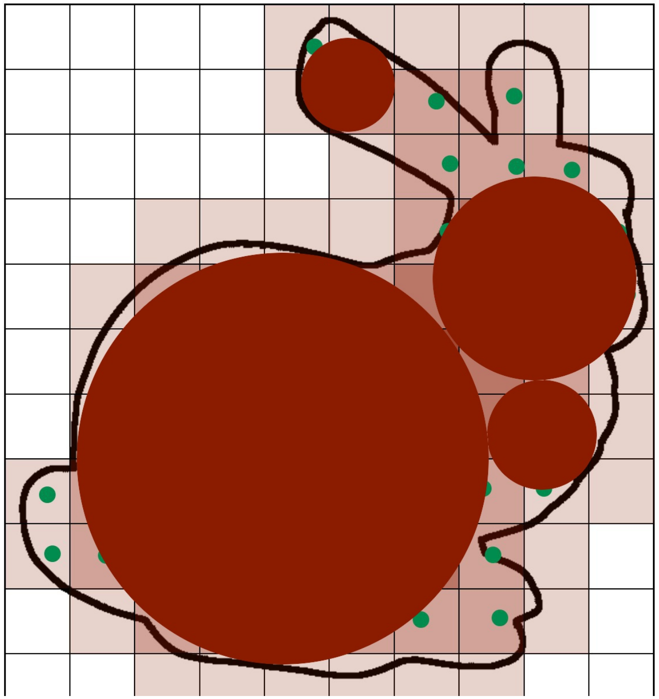


Protosphere Recap

Starting efficiency low (many Prototypes but few new spheres)

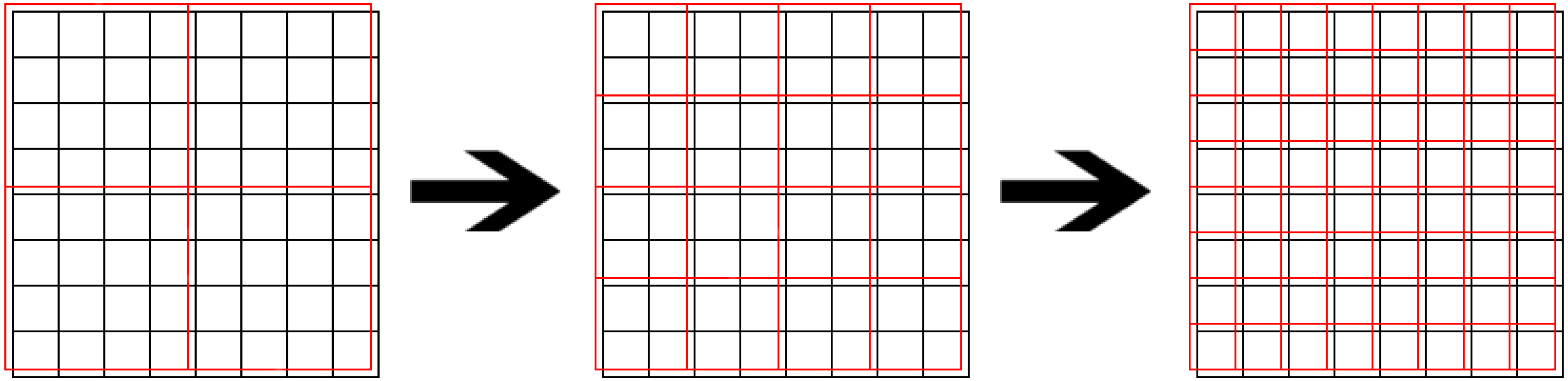


Fill rate at the end depends on predefined resolution



Implicit Grid Refinement

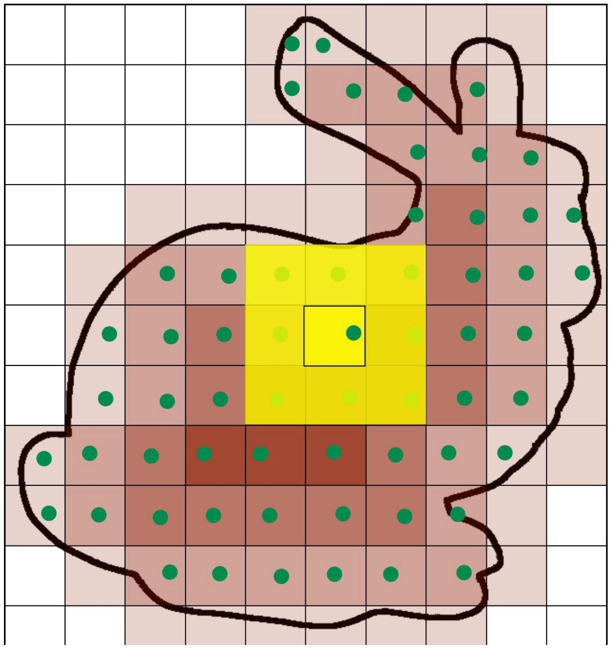
- We need: coarse grid at the start, fine grid at the end
- Idea:
  - Split every cell periodicly into 2^n new cells
  - decouple the prototypes from the triangle data



Implicit Grid Refinement

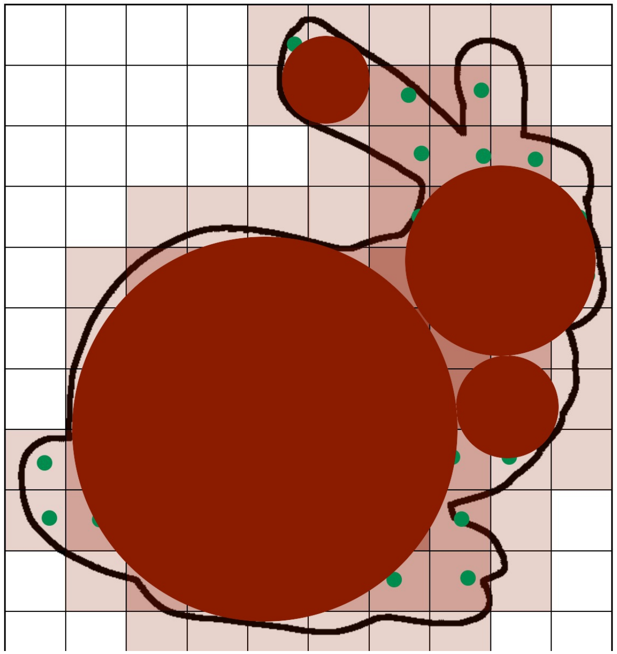
+

Higher degree of parallelisation



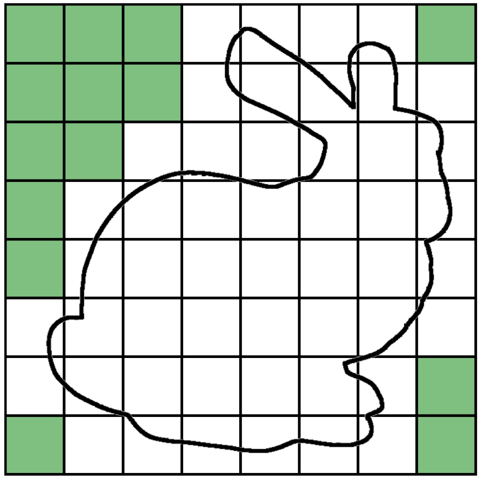
−

Predefined maximal grid resolution

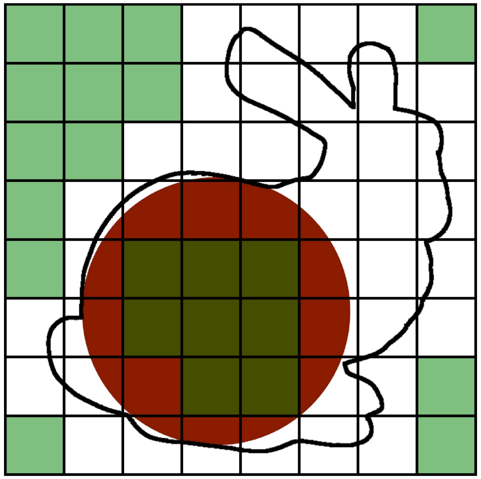


Adaptive Grid Storage

- End performance depends on fine grids
- Fine grids need a lot of memory
- Idea: save only cells inside or on the border of the object

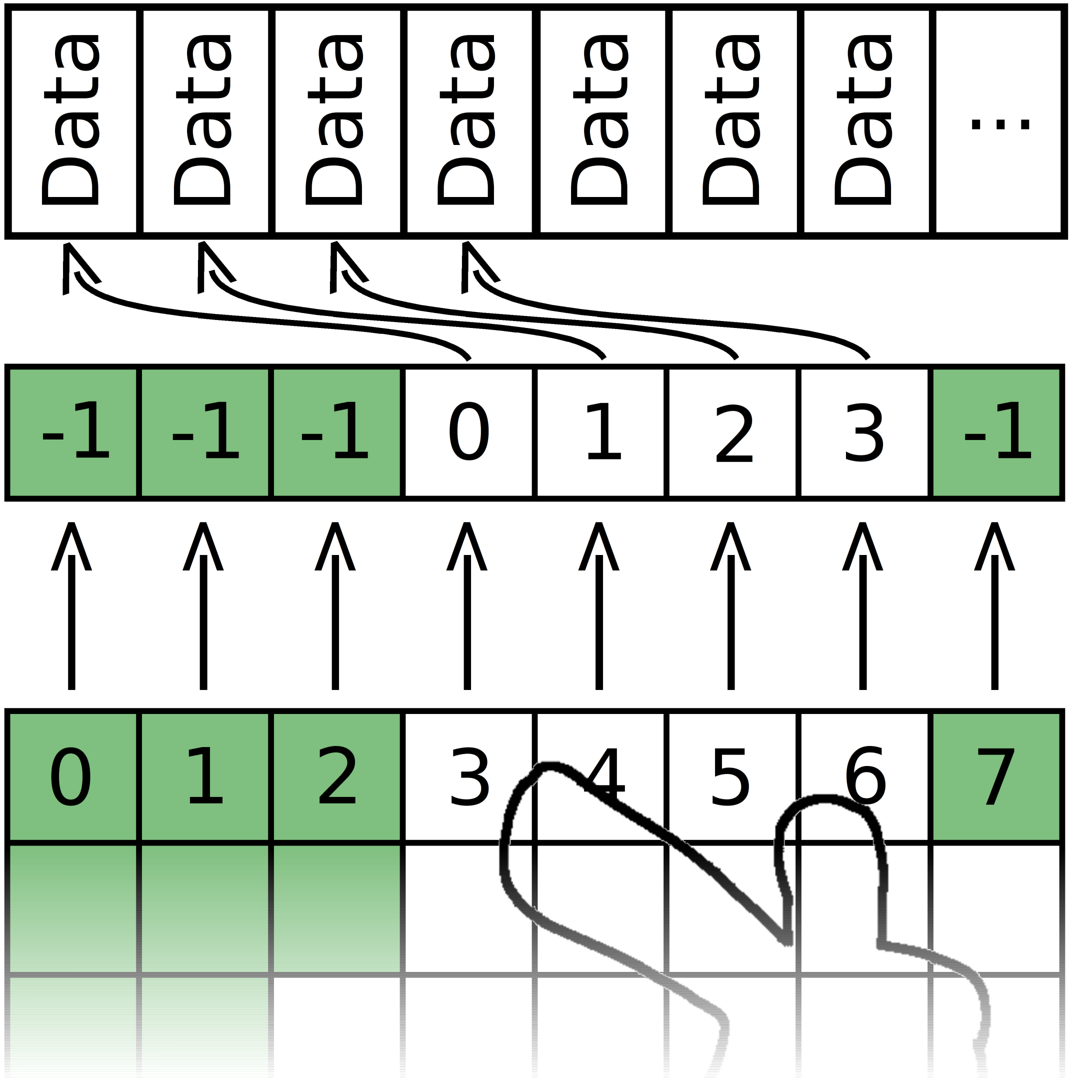


~22% outside



~34% outside

Adaptive Grid Storage



- Working Array:

Data	Data	Data	Data	Data	Data	Data	...

- Adress Array:

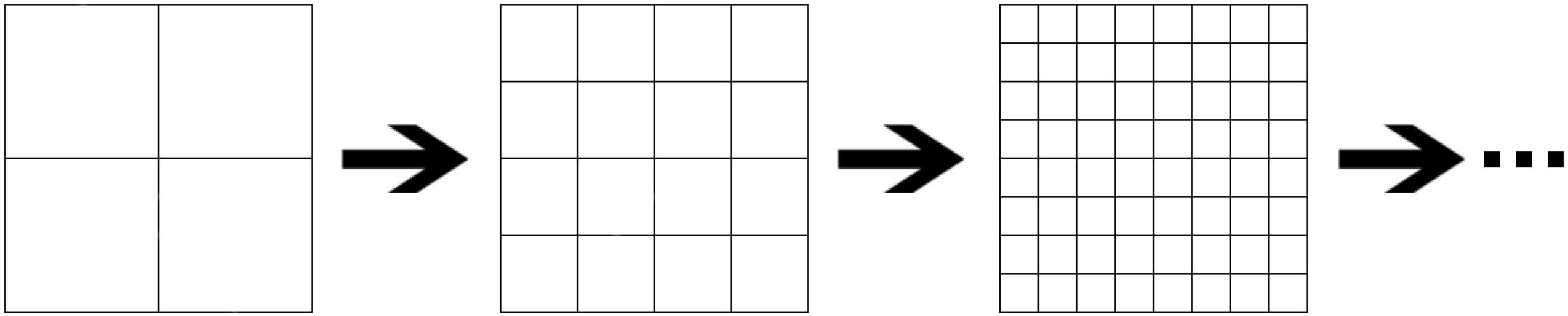
-1	-1	-1	0	1	2	3	-1

- Index in the grid:

0	1	2	3	4	5	6	7

Explicit Grid Refinement

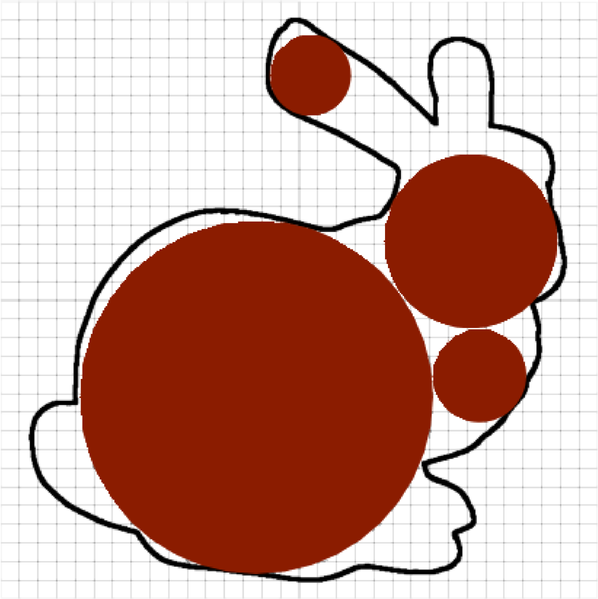
- We need: refine the grid while packing the object
- Idea: Split every cell in the data grid periodicly into 2^n new cells



Explicit Grid Refinement

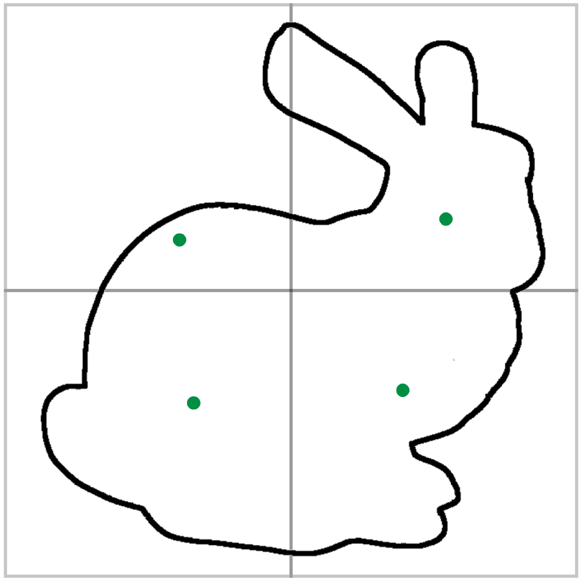
+

Good performance when splitting to very fine grids at the end



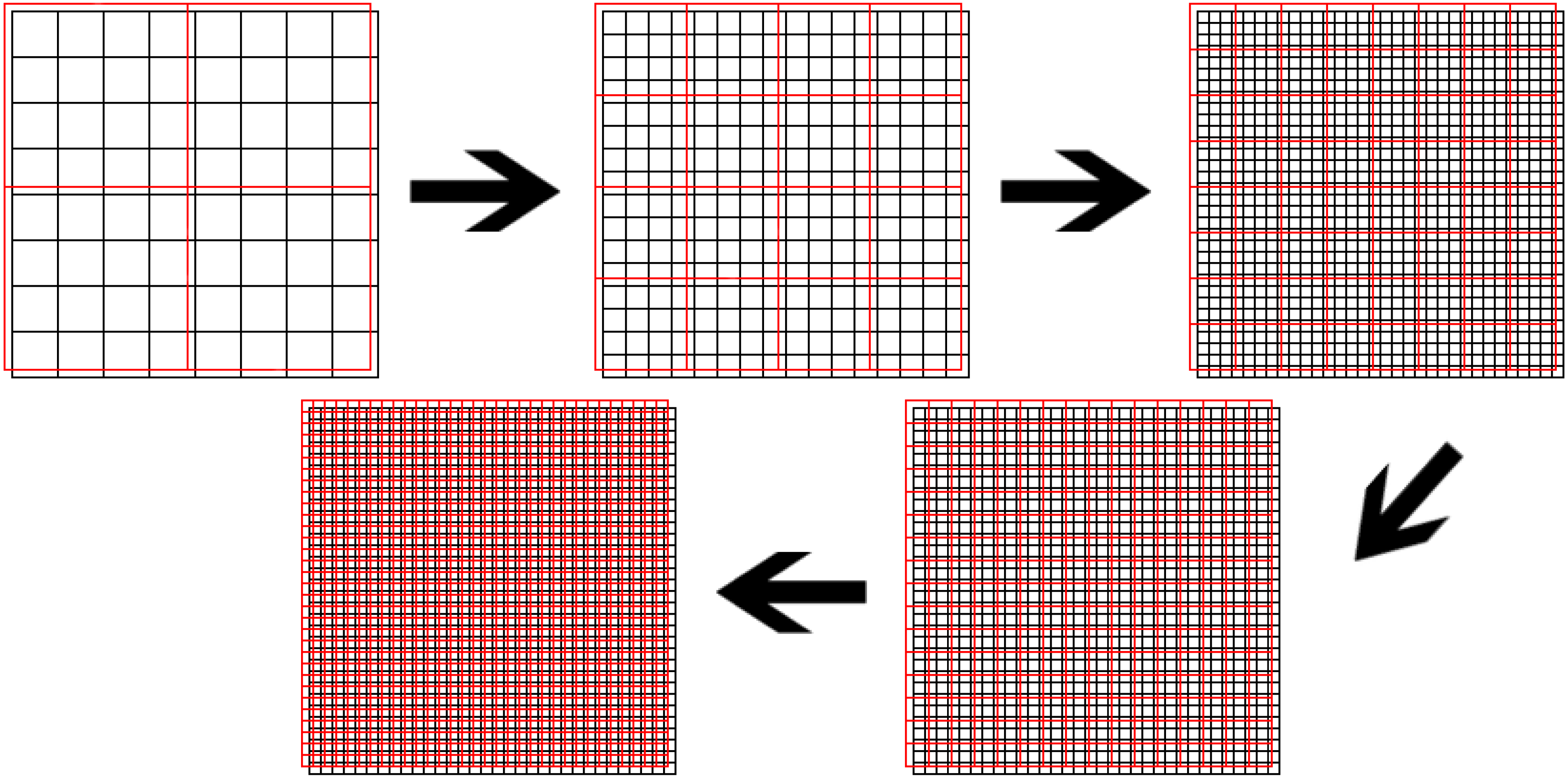
−

Many triangle-to-point-tests for one prototype at the start

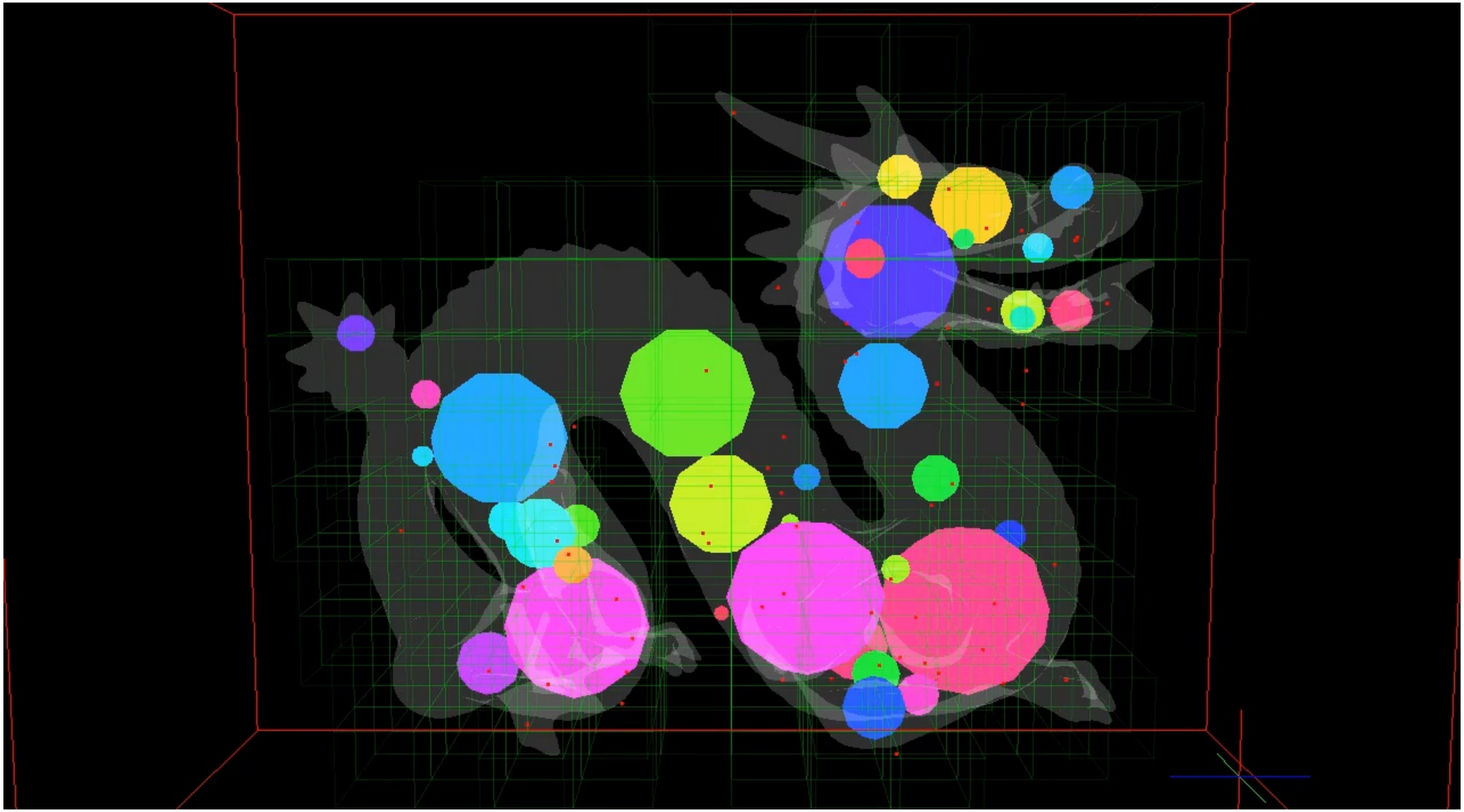


Hybrid Grid

- Combines the explicit and implicit grid refinements



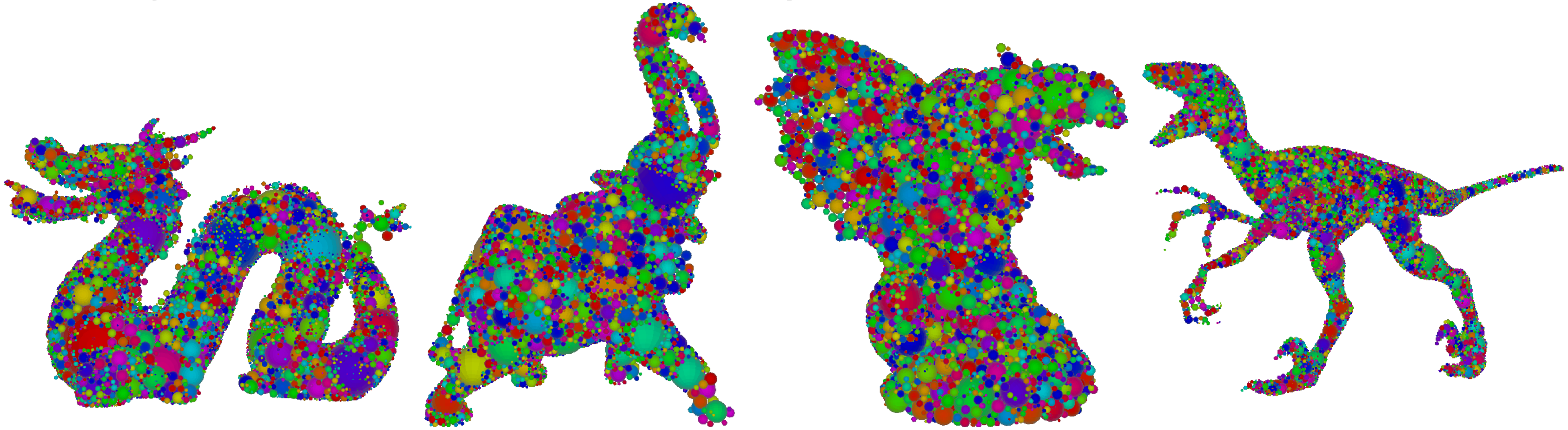
Demonstration



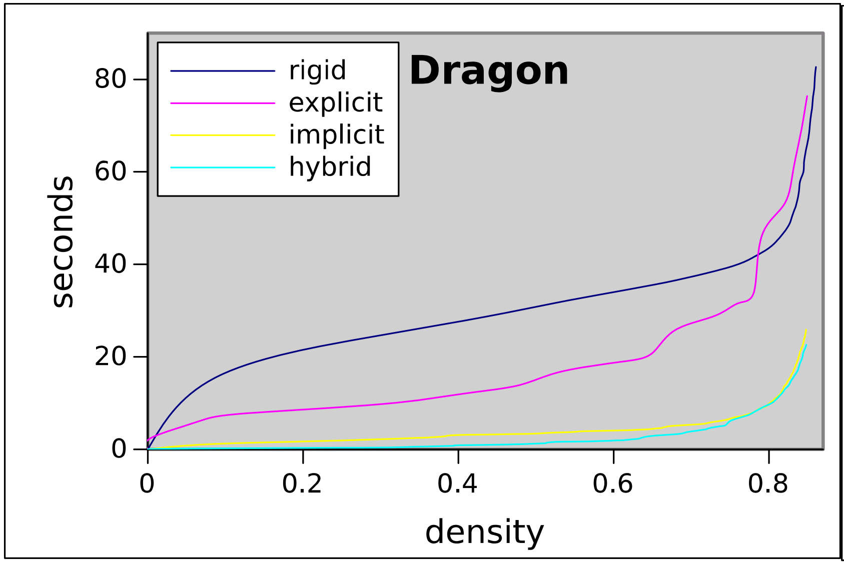
Results

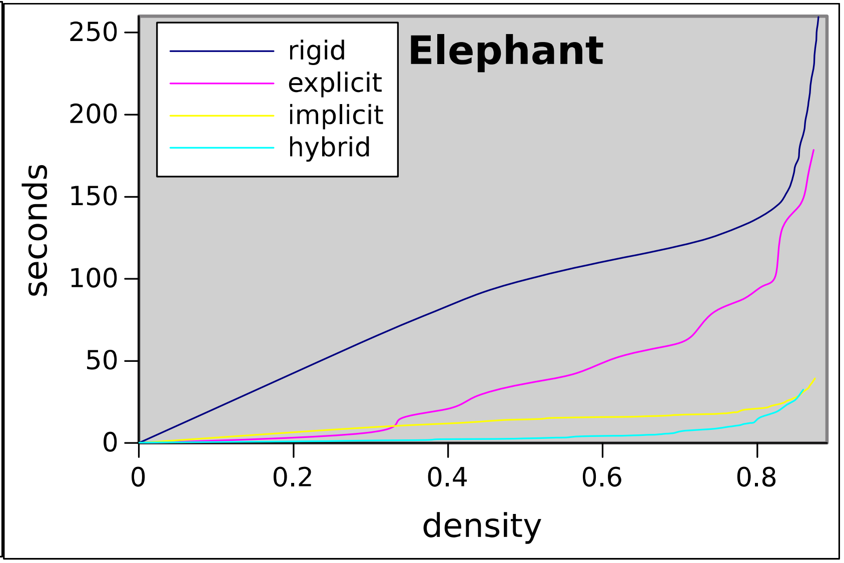
- Objects used: (up to 800k triangles)

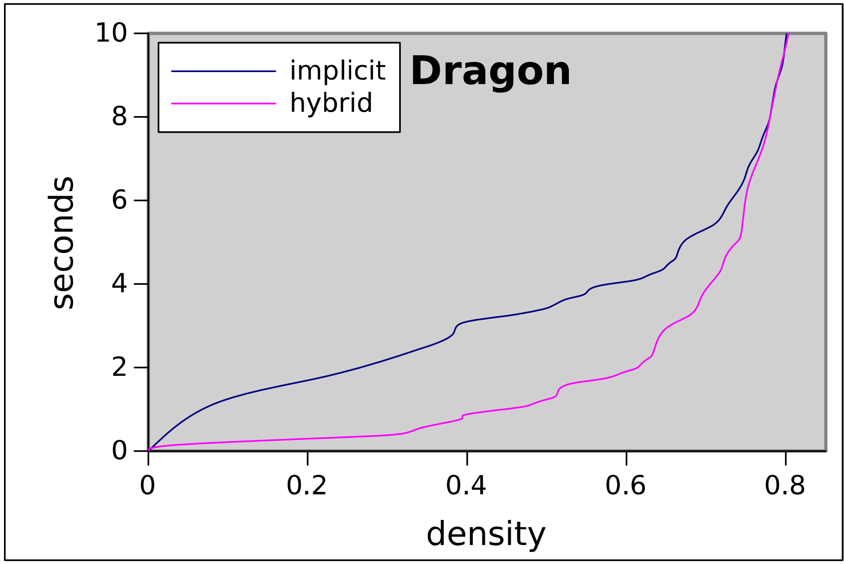
- Objects filled: (about 200k spheres)

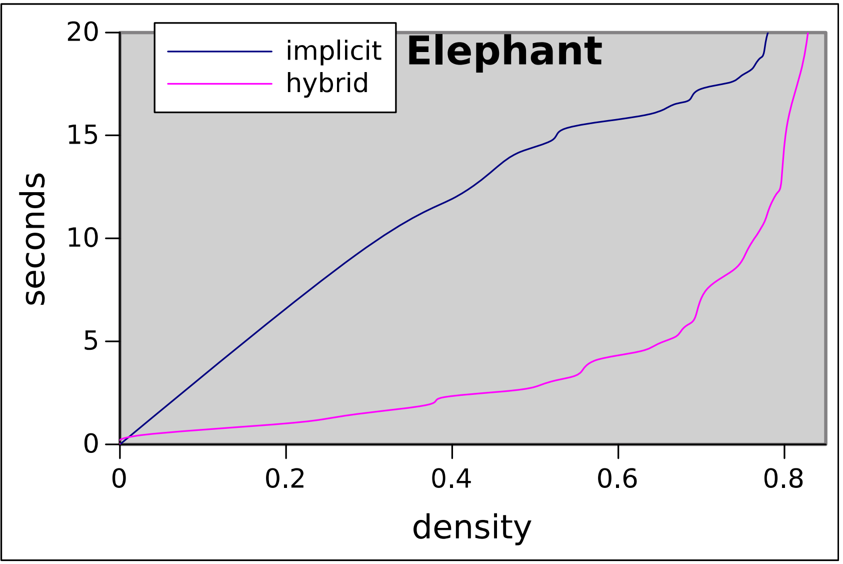


Results


Dragon
rigid
explicit
implicit
hybrid
seconds
0
20
40
60
80
density
0
0.2
0.4
0.6
0.8


Elephant
rigid
explicit
implicit
hybrid
seconds
0
50
100
150
200
250
density
0
0.2
0.4
0.6
0.8


Dragon
implicit
hybrid
seconds
0
2
4
6
8
10
density
0
0.2
0.4
0.6
0.8


Elephant
implicit
hybrid
seconds
0
5
10
15
20
density
0
0.2
0.4
0.6
0.8

Conclusion and Future Works

- Hybrid Grid is 100 times faster than Protosphere
- 200,000 spheres and 95% coverage in ~30 seconds
- 80% coverage in less than 10 seconds
- First iteration in near real-time (~0.05 seconds)

- Optimal parameter guessing based on object
- More object representations
- Testing more grid-like data structures