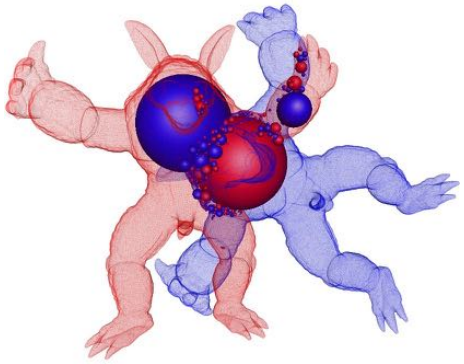
Bremen

CG VR

Parallel Collision Detection in Constant Time



Rene Weller, Udo Frese and Gabriel Zachmann

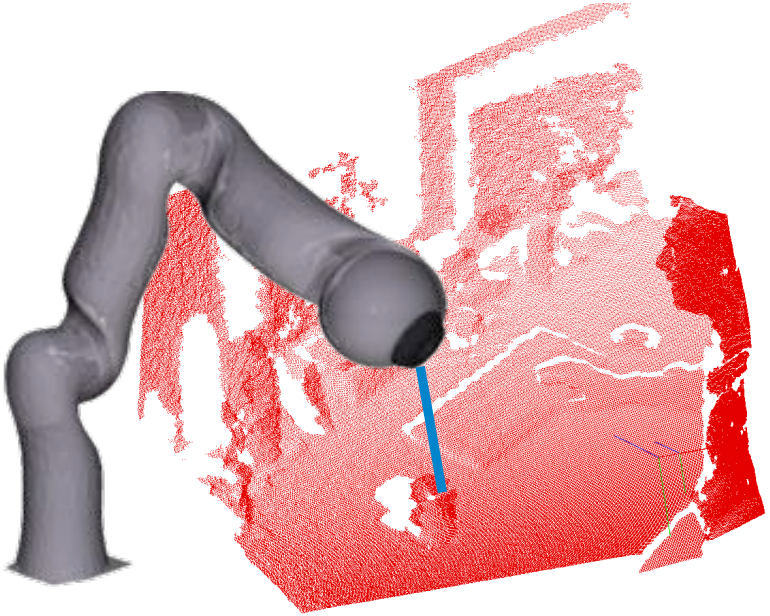
University of Bremen

cgvr.cs.uni-bremen.de

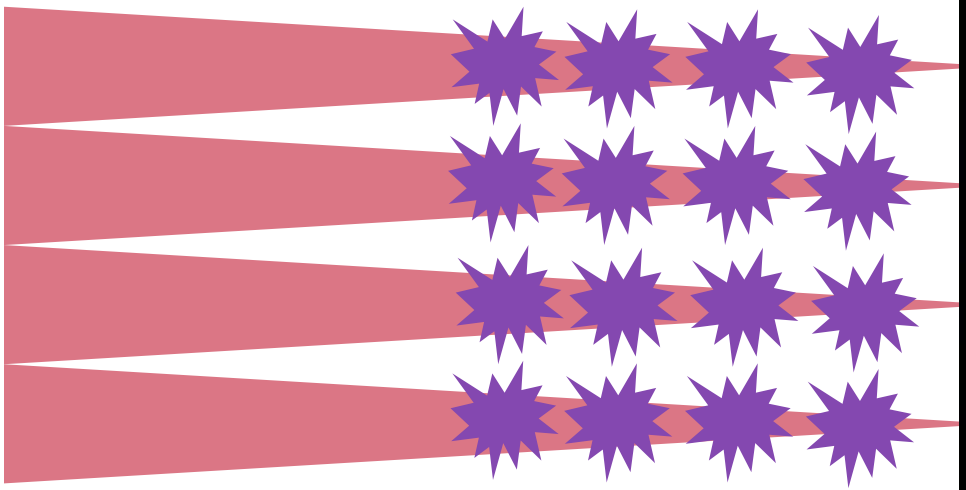
Motivation

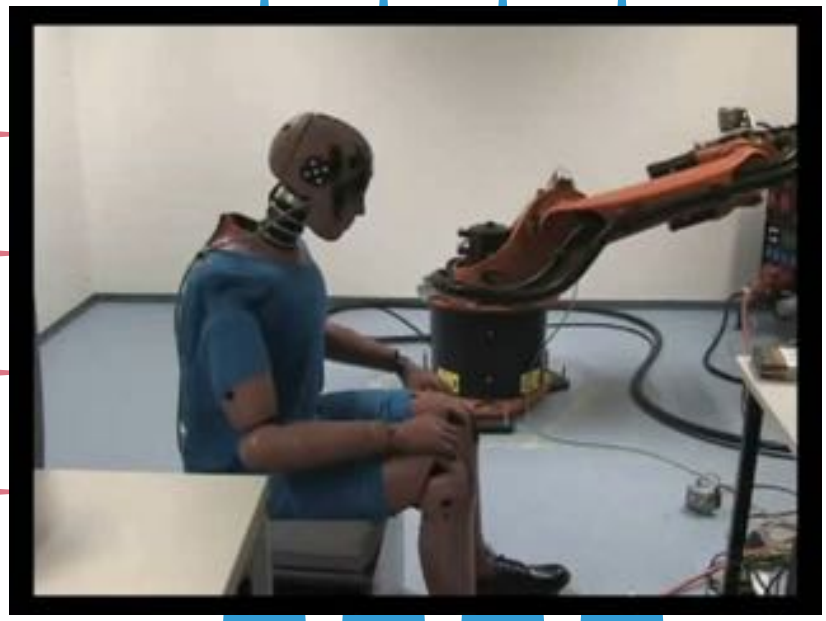
[The Elder Scrolls V, 2011]

[NBA 2k13, 2012]



Motivation





[Haddadin, 2009]

$$O(n^2)$$

Previous Works

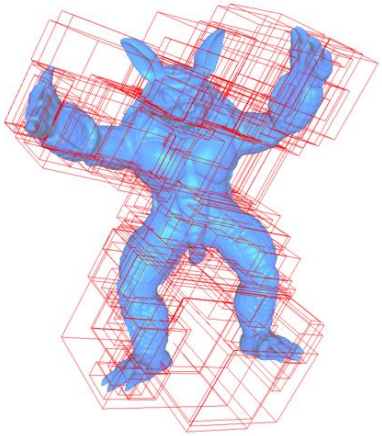
- Distance of convex polytopes [Lin & Canny, 1991]:

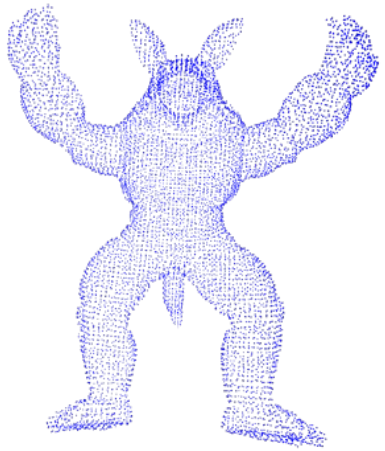
  Worst case: $\mathcal{O}(\sqrt{n})$, n = # faces

- All intersections of *n* convex polytopes [Suri et al., 1998]:

  $\mathcal{O}((n+k)\log\uparrow 2\ n)$, k = # intersecting pairs

- Voxel based algorithms [McNeely et al., 1999]
  - Worst case: $\mathcal{O}(n)$ n=# points in pointshell
  - Fast, independent of object's triangle count
  - Memory consuming, aliasing artifacts

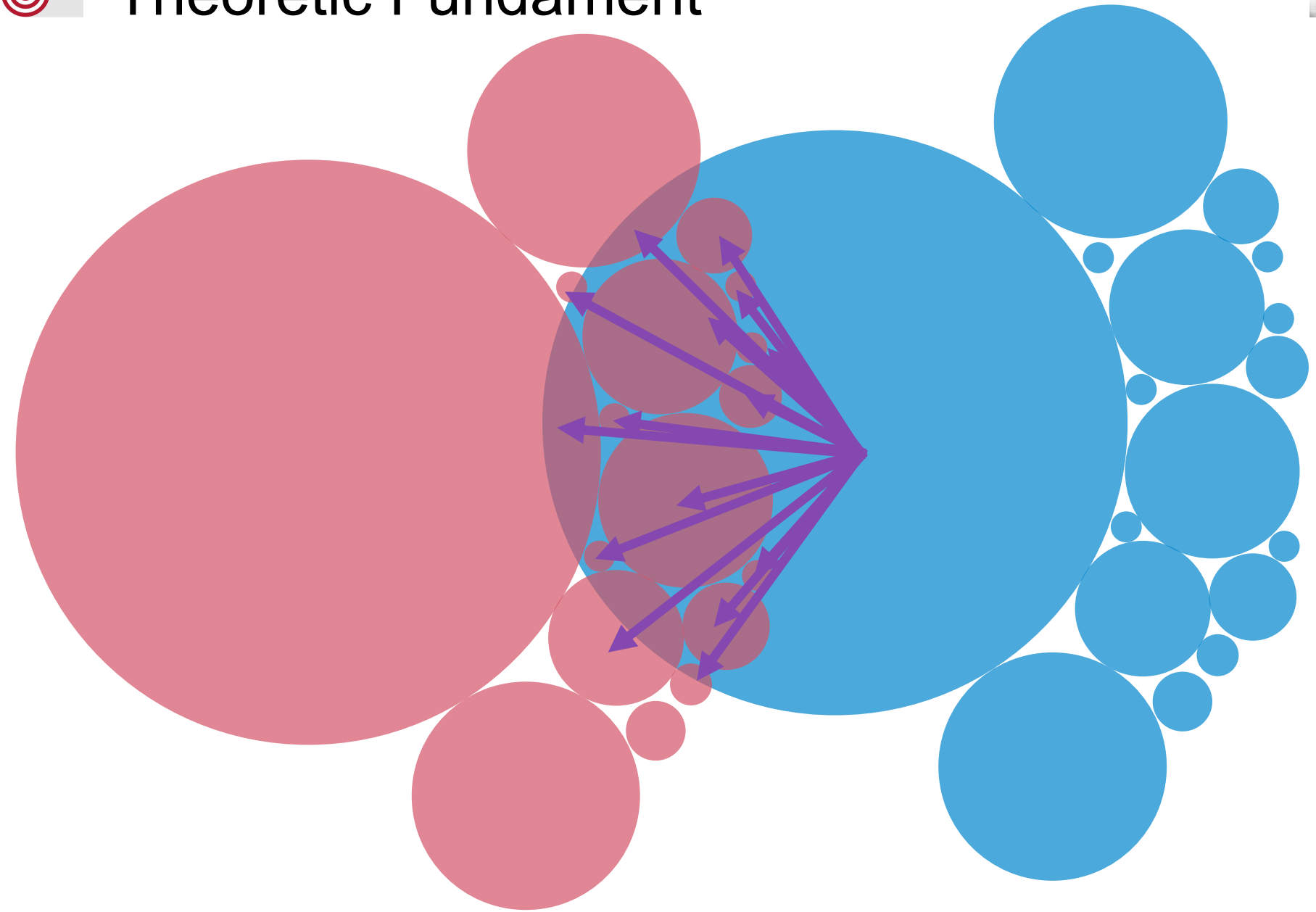




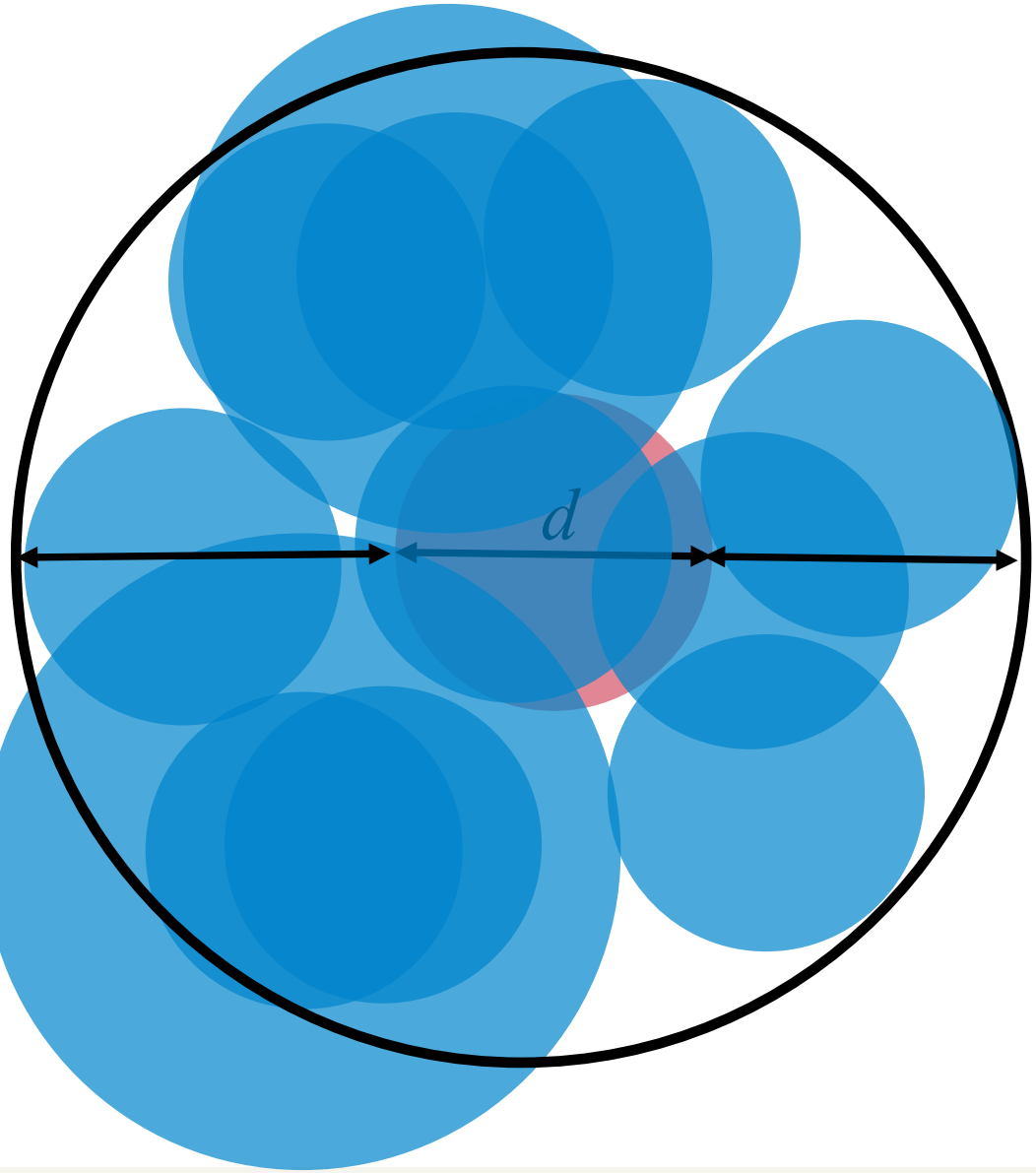
Preliminary Considerations

- Fill object
  - from the *inside*
  - with *non-overlapping*
  - *polydispere* spheres
- Independent of object representation
- Penetration volume
  - “the most complicate yet accurate method” [Fisher and Lin, 2001]
  - Continuous forces and torques
- Complexity: ?

Theoretic Fundament



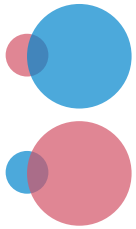
Theoretic Fundament



Theoretic Fundament

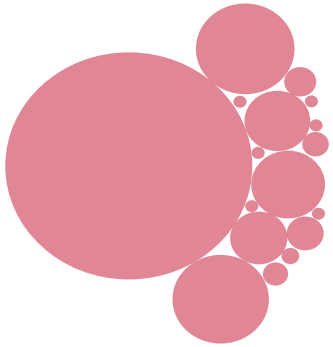
- Lemma: A single sphere s intersects a constant number of disjoint spheres A with at least the same radius.
- Theorem: The maximum number of intersecting pairs of spheres of two polydisperse sphere packings A and B with n spheres is in $O(n)$.

  - Check each sphere $s \downarrow i \in A$ against larger spheres in B
  - Check each sphere $s \downarrow j \in B$ against larger spheres in A



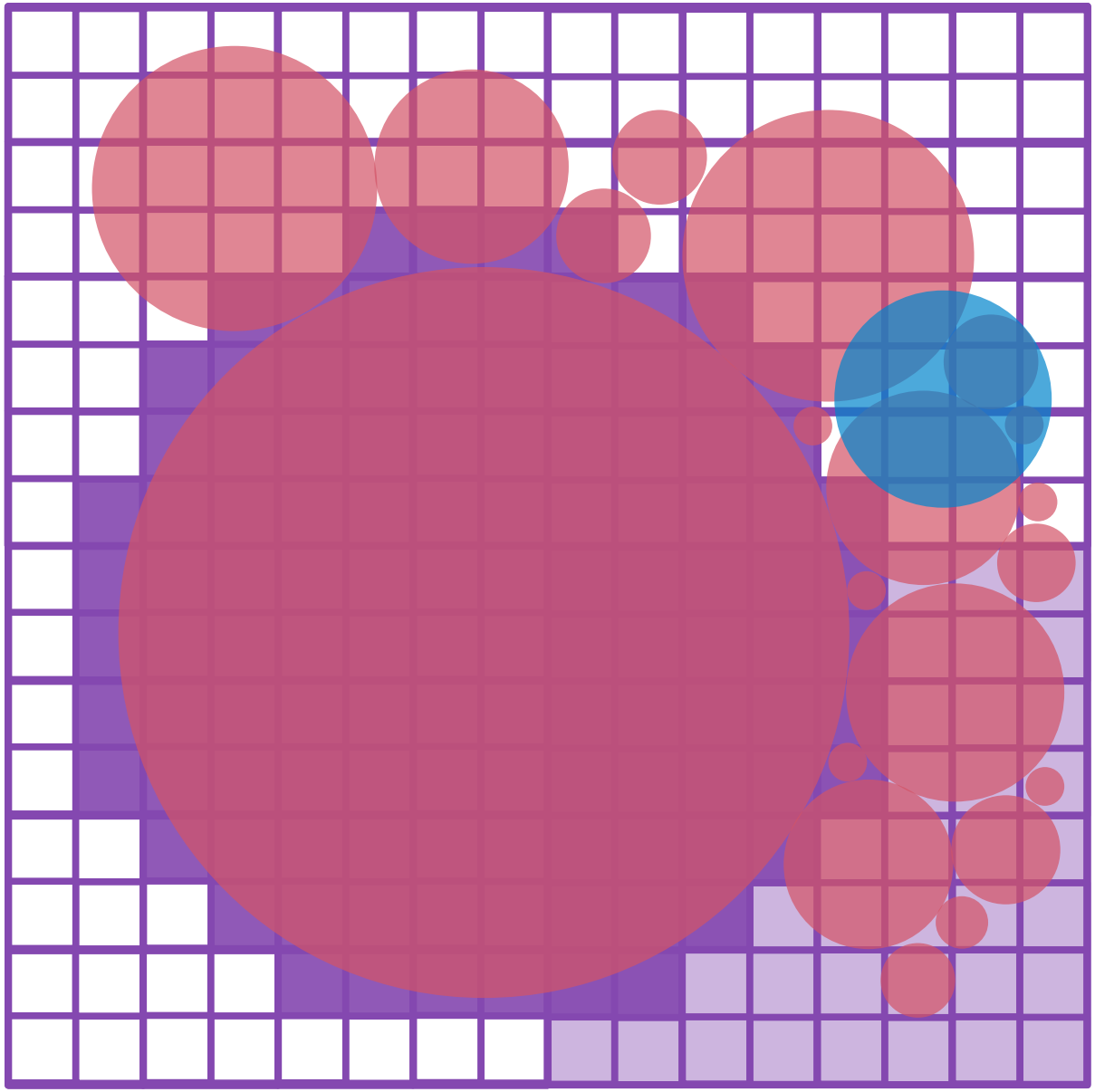
$\Rightarrow O(1)$ for each $s \downarrow i \in A$ and $s \downarrow j \in B$

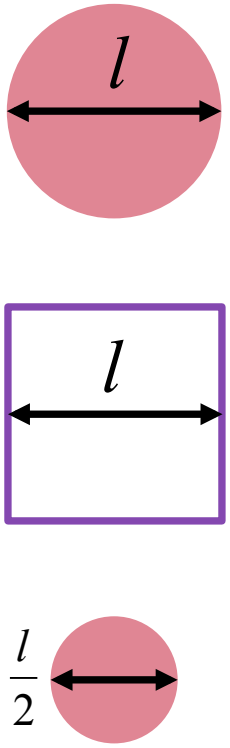
$\Rightarrow O(n)$ in total



Our Algorithm

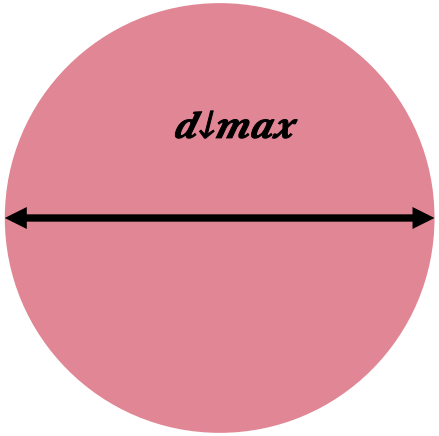
Level 4

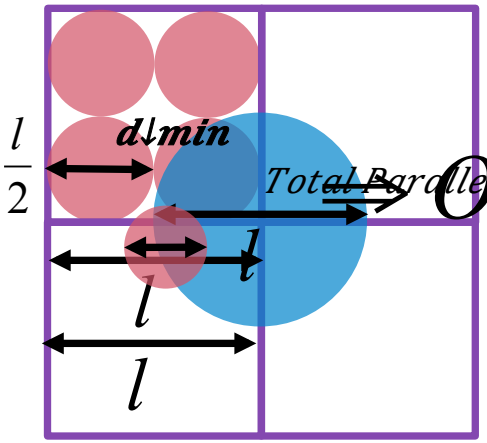




Analysis

```
For all spheres s∈B:                                   ⇒O(n)
    Compute hierarchy level l
    For all levels l≤l↓i ≤l↓max:                       ⇒O(1)
        For all cells c↓j in level l↓i overlapped by s ⇒O(1)
            For all spheres s↓k ∈c↓j                   ⇒O(1)
                Compute overlap volume for s and s↓k   ⇒O(1)
```



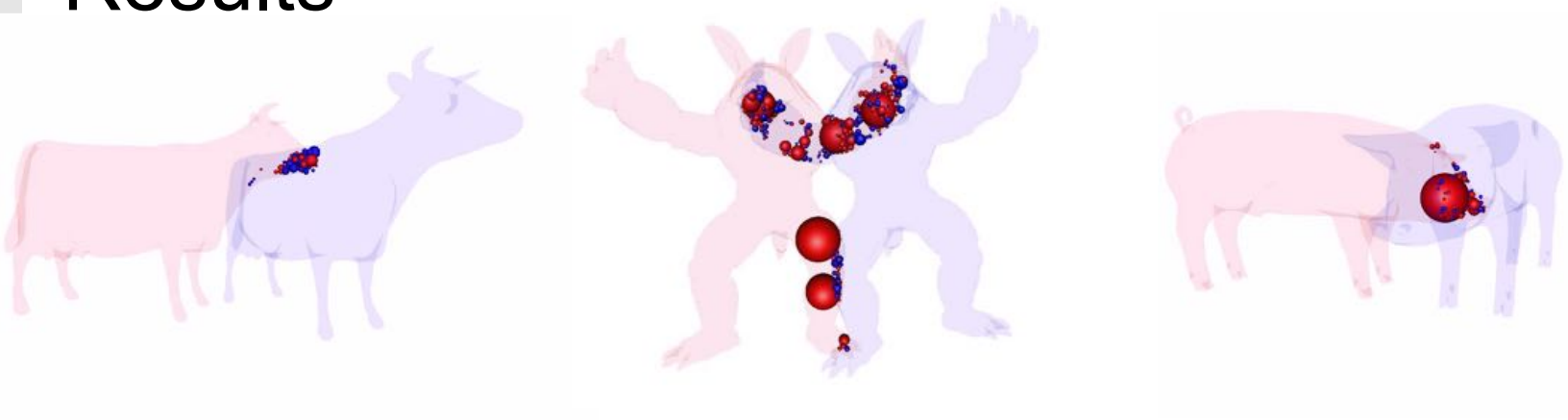


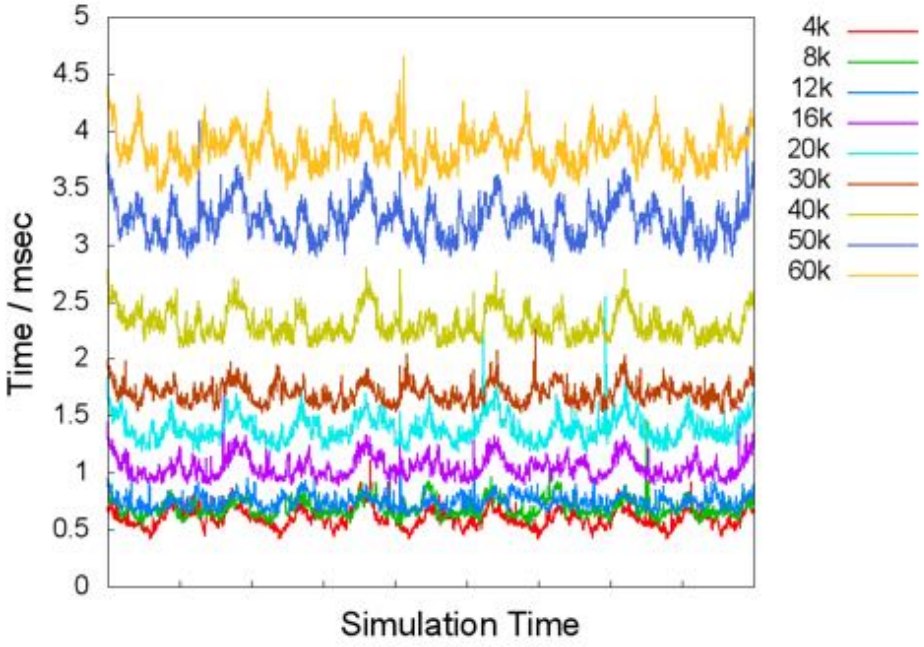
Total Time: $O(n)$

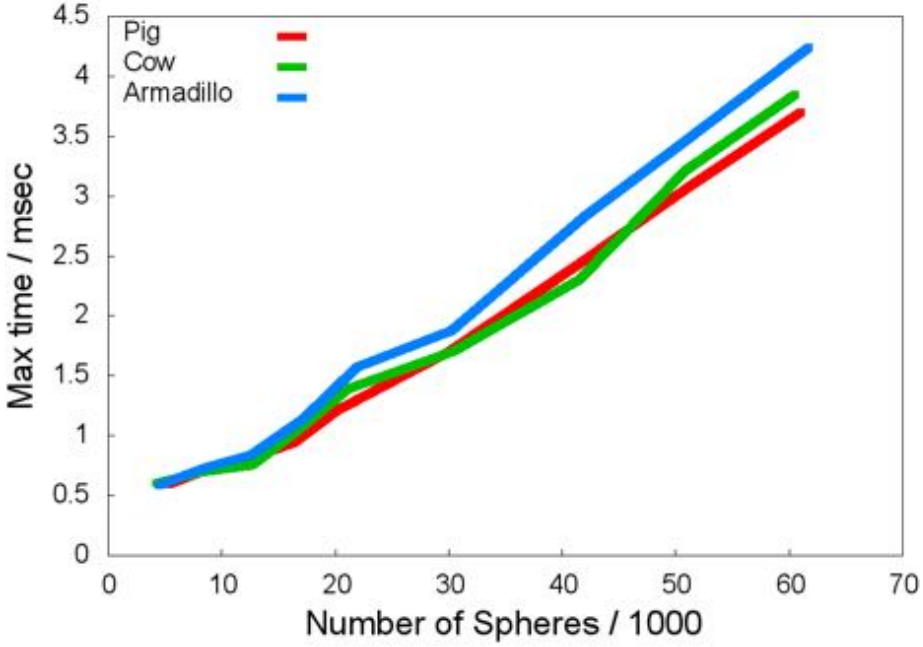
Total Parallel Time: $O(1)$

$$\Rightarrow O\left(\log \frac{d_{\max}}{d_{\min}}\right)$$

Results

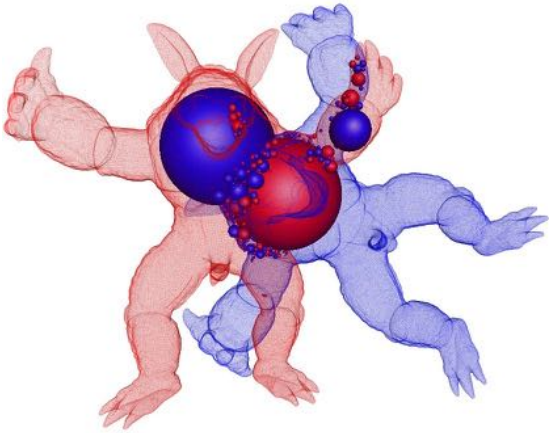


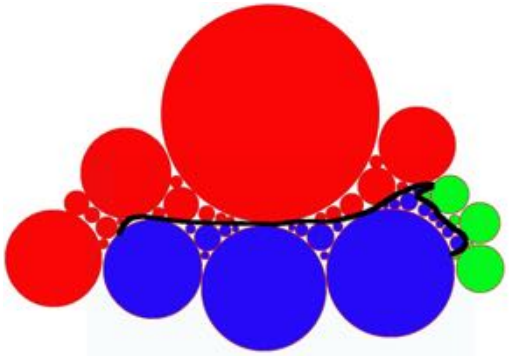

Time / msec
Simulation Time
4k
8k
12k
16k
20k
30k
40k
50k
60k

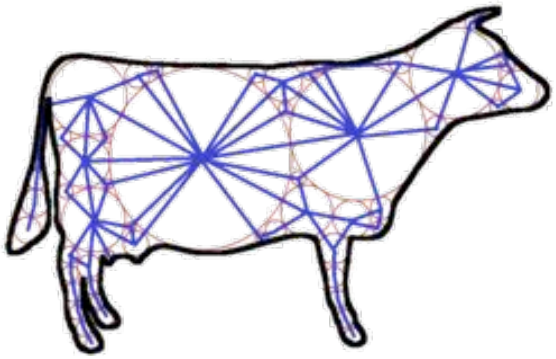

Max time / msec
Number of Spheres / 1000
Pig
Cow
Armadillo

Conclusions and Future Work

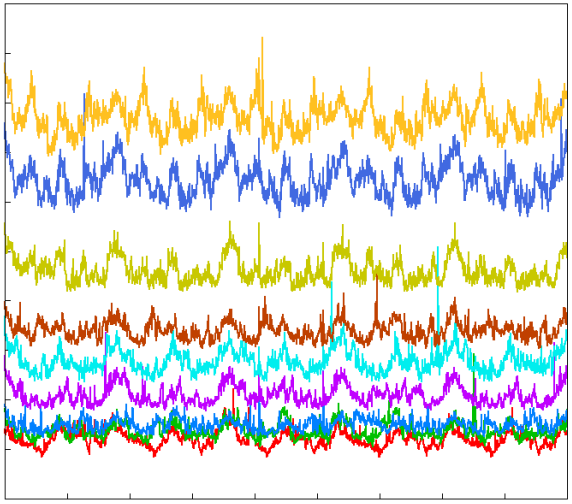
- Proven $\mathcal{O}(n)$ complexity for the overlap of two sphere packings
- New collision detection algorithm
  - Worst case sequential time: $\mathcal{O}(n)$
  - Worst case parallel time: $\mathcal{O}(1)$
- Geforce GTX680: < 1 msec for 30k spheres
- Extension to deformable objects
  - Sphere-Spring-Systems
- Support of thin sheets





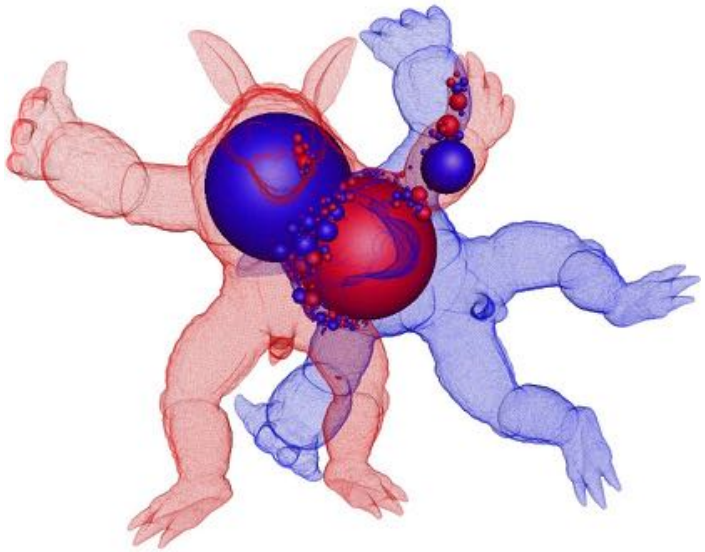


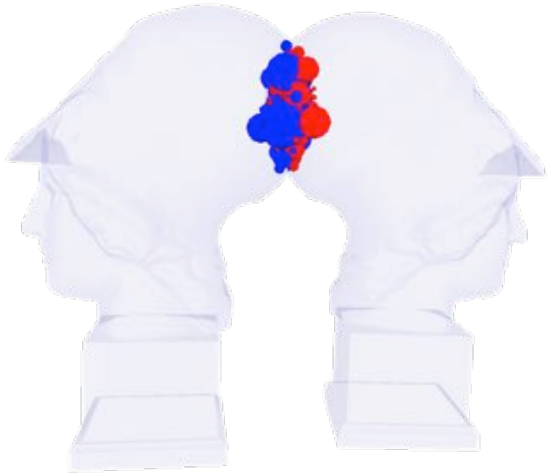
Thank You!



$O(1)$

$O(n)$





- DFG Grant SFB/TR 8 I08 [DextrousSpace], GZ TRR 8/3-2013