Massively Parallel Batch Neural Gas for BVH Construction

R. Weller[1], D. Mainzer[2], **A. Srinivas[1]**, M. Teschner[3] and G. Zachmann[1]

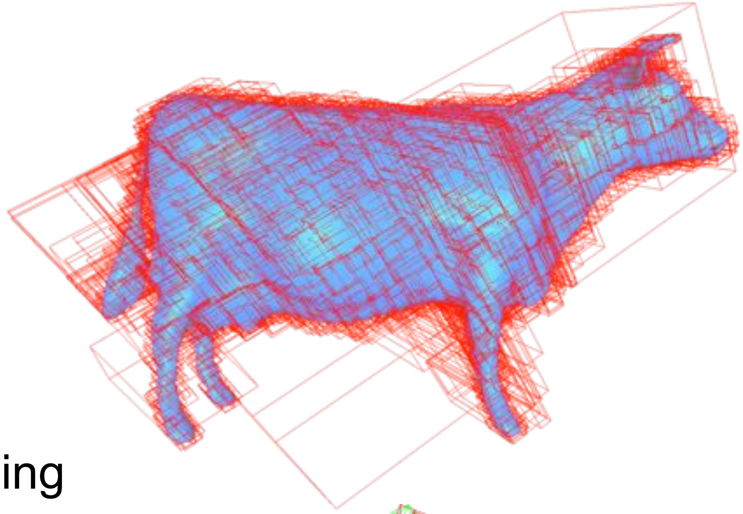
[1] University of Bremen, Germany

[2] Clausthal University, Germany

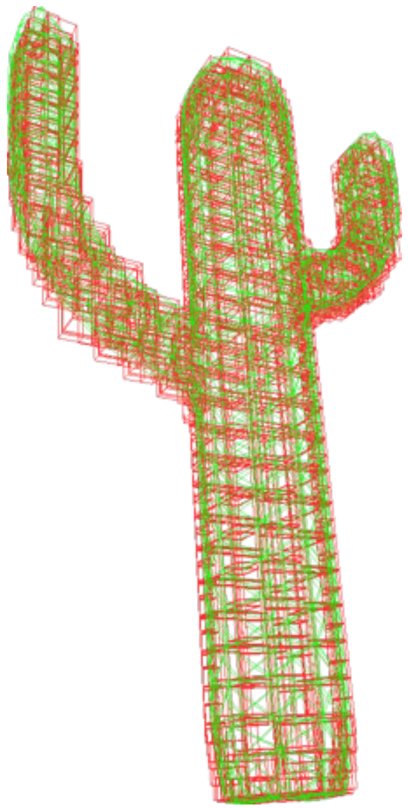
[3] University of Freiburg, Germany

VRIPHYS Sept 2014, Bremen

Motivation for BVHs



- Commonly used in wide variety of graphics problems
  - Collision detection, Ray tracing, Culling
- Bounding Volumes:
  - AABB, OBB, Sphere, Convex Hull, ...
- Branching factor and splitting criteria



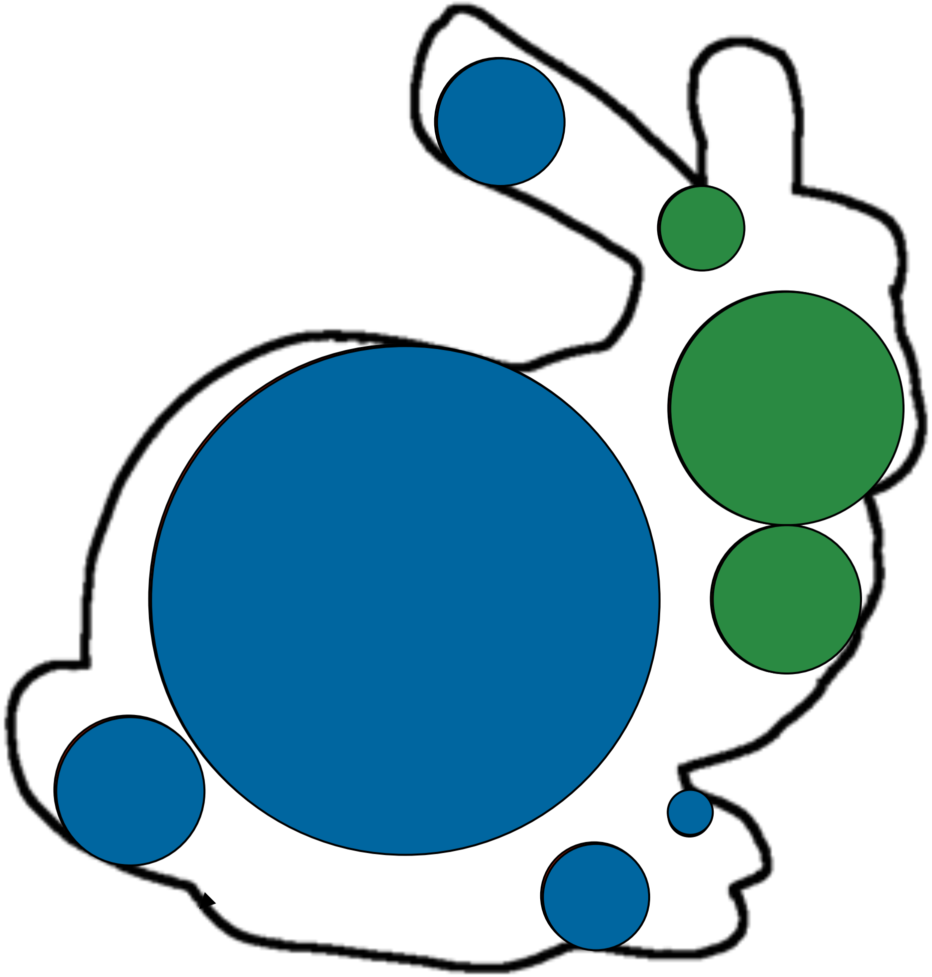
Previous Work

- Binary axis aligned bounding box tree [Bergen97]
  - Splitting AABB along longest side
- Surface area heuristic BVHs on GPU [LGS*09]
  - Slower build time than CPU version
  - Faster traversal of BVH
- Oriented bounding box-tree construction [GLM96]
  - Presented a new construction method
  - Optimal Solution in $O(n^3)$ and hard to implement [O'R85]

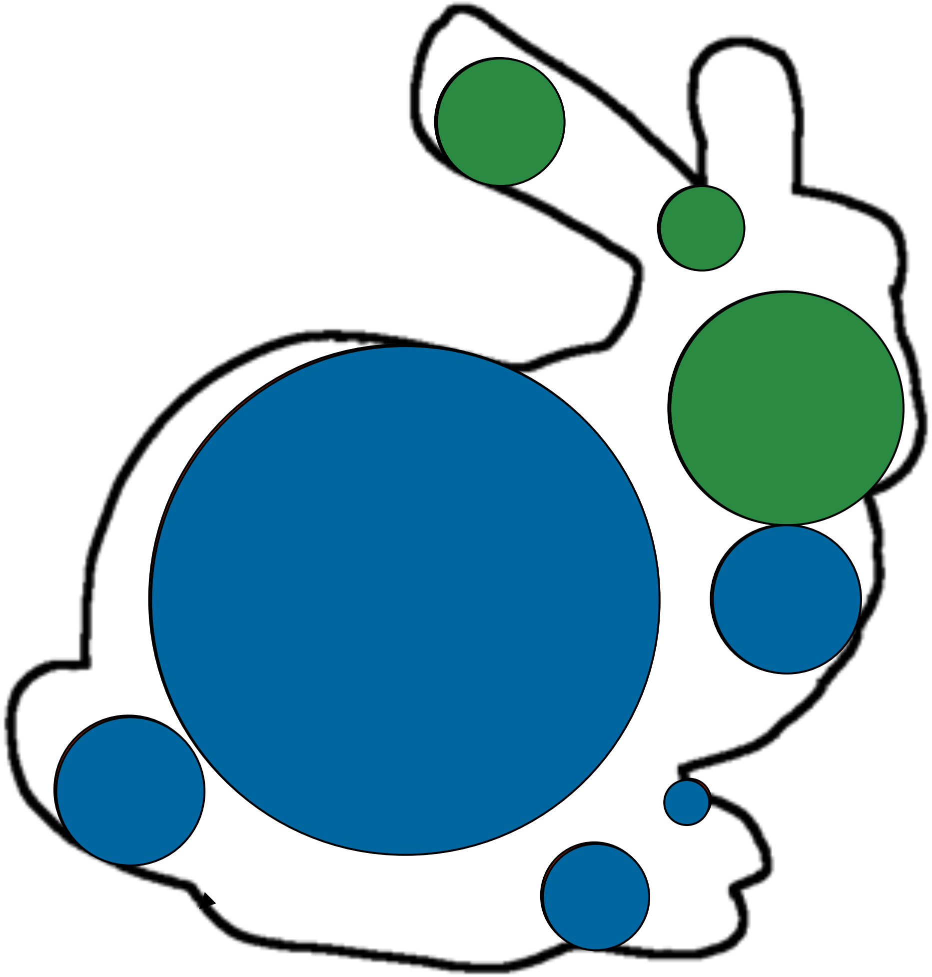
Volume-based approaches

- Volumetric method: volumetric poly-disperse sphere packing
  - All sphere are *inside* and *do not overlap* each other
- Need different approach for constructing hierarchy tree
- Investigate on other factors such as
  - Splitting criteria
  - Branching factor

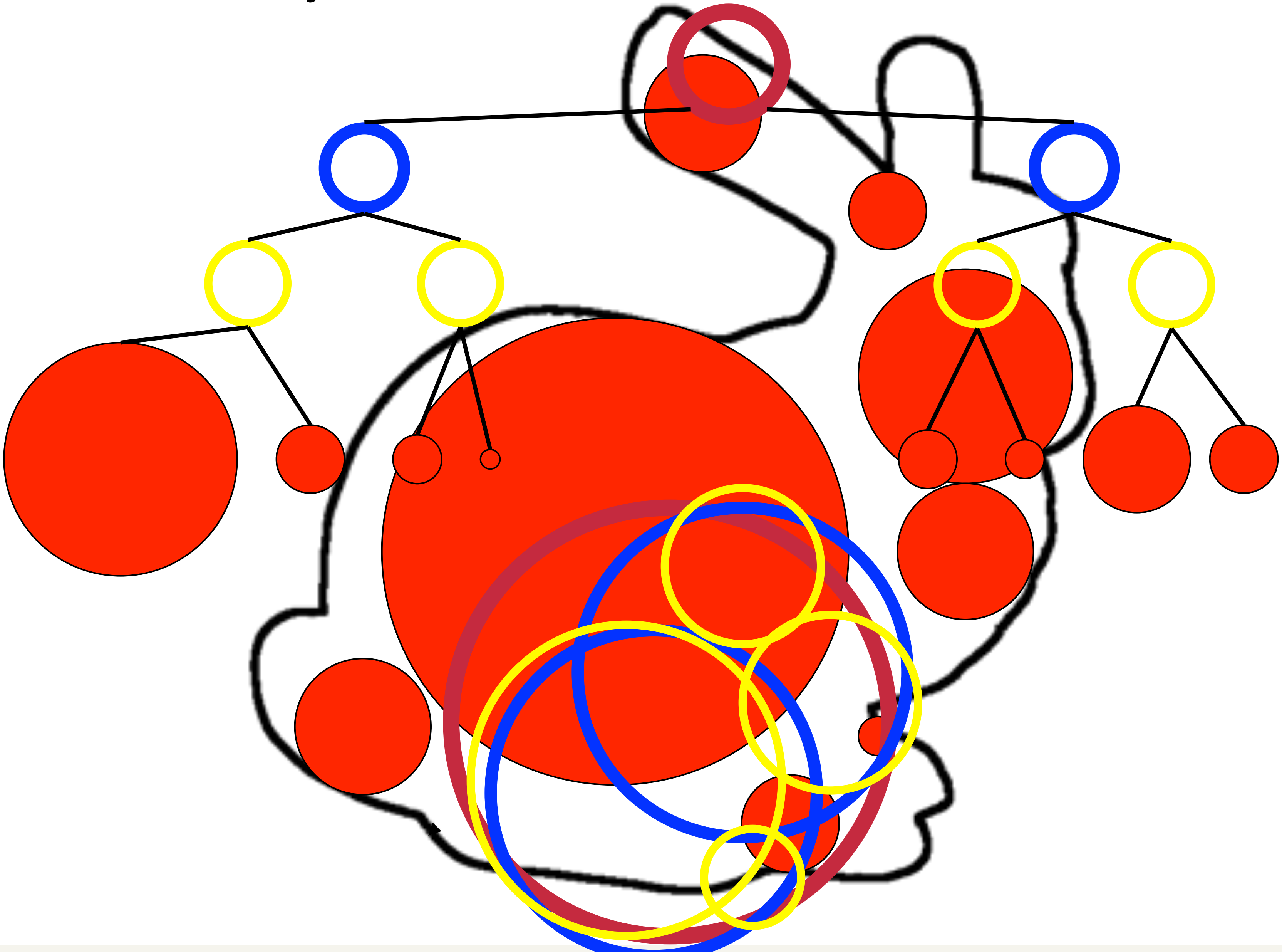
Biggest Sphere Splitting Criterion



Outer Sphere Splitting Criterion



Hierachy Creation



Batch Neural Gas

- Cost function minimize mean squared Euclidean distance for each data point to its nearest center point
- Very robust – independent of initialization of center points
- Extendable to define *importance* of a data point
- Rank for prototypes (with $n$prototypes):

$$k_{ij} := |\{w_k : d(x_j, w_k) < d(x_j, w_i)\}| \in \{0, \ldots, n\}$$

- Position for prototypes:

$$w_i := \frac{\sum_{j=0}^{m} h_\lambda(k_{ij}) x_j}{\sum_{j=0}^{m} h_\lambda(k_{ij})}$$

- Convergence rate controlled by monotonically decreasing function $h_\lambda(\ldots)$

- BNG only utilizes location of the center of the spheres
- Ignores the extent of the spheres
  - Prototypes avoid regions covered with a very large sphere
  - Regions treated as outlier → Non-Optimal IST
- Extended BNG Version
  - Magnification controlled BNG [HHV06]

$$w_i := \frac{\sum_{j=0}^{m} h_\lambda(k_{ij}) v(x_j) x_j}{\sum_{j=0}^{m} h_\lambda(k_{ij}) v(x_j)}$$

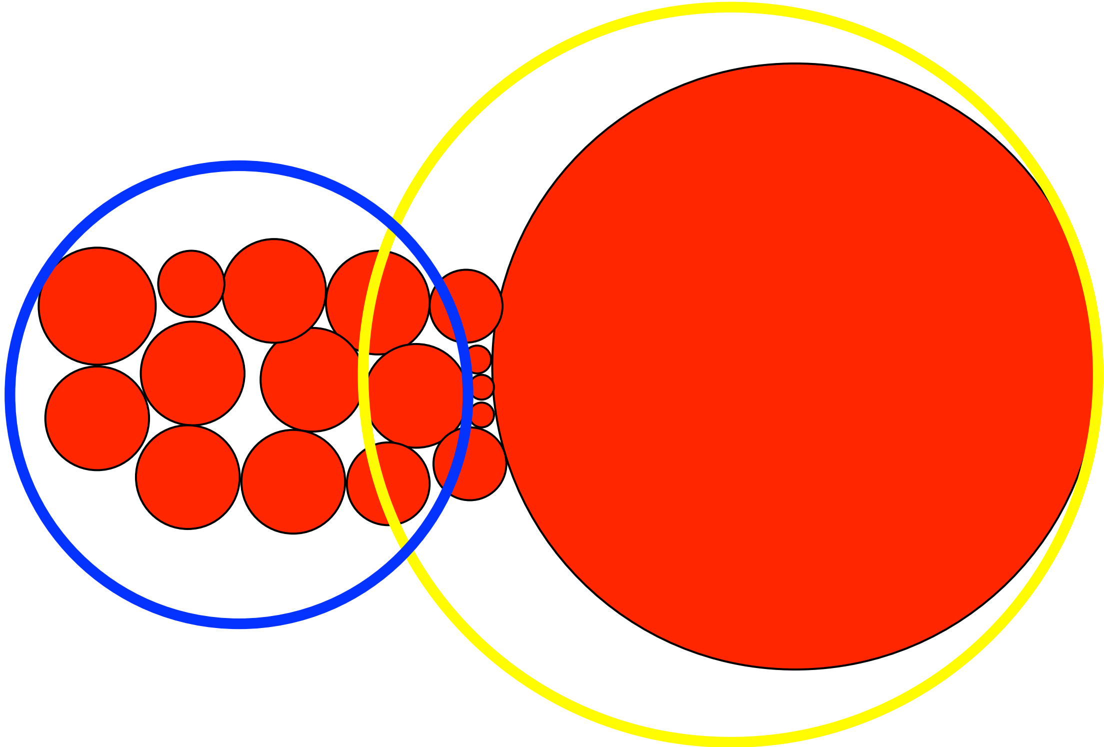
  - Use volume of the sphere: $v(x_j) = \frac{4}{3}\pi r^3$
  - Runtime: $\mathcal{O}(n \log n)$

- BNG only utilizes location of the center of the spheres
- Ignores the extent of the spheres
  - Prototypes avoid regions covered with a very large sphere
  - Regions treated as outlier → Non-optimal bounding spheres
- Extended BNG Version
  - Magnification controlled BNG [HHV06]

$$w_i := \frac{\sum_{j=0}^{m} h_\lambda(k_{ij}) v(x_j) x_j}{\sum_{j=0}^{m} h_\lambda(k_{ij}) v(x_j)}$$

  - Use volume of the sphere: $v(x_j) = \frac{4}{3}\pi r^3$
  - Runtime: $\mathcal{O}(n \log n)$

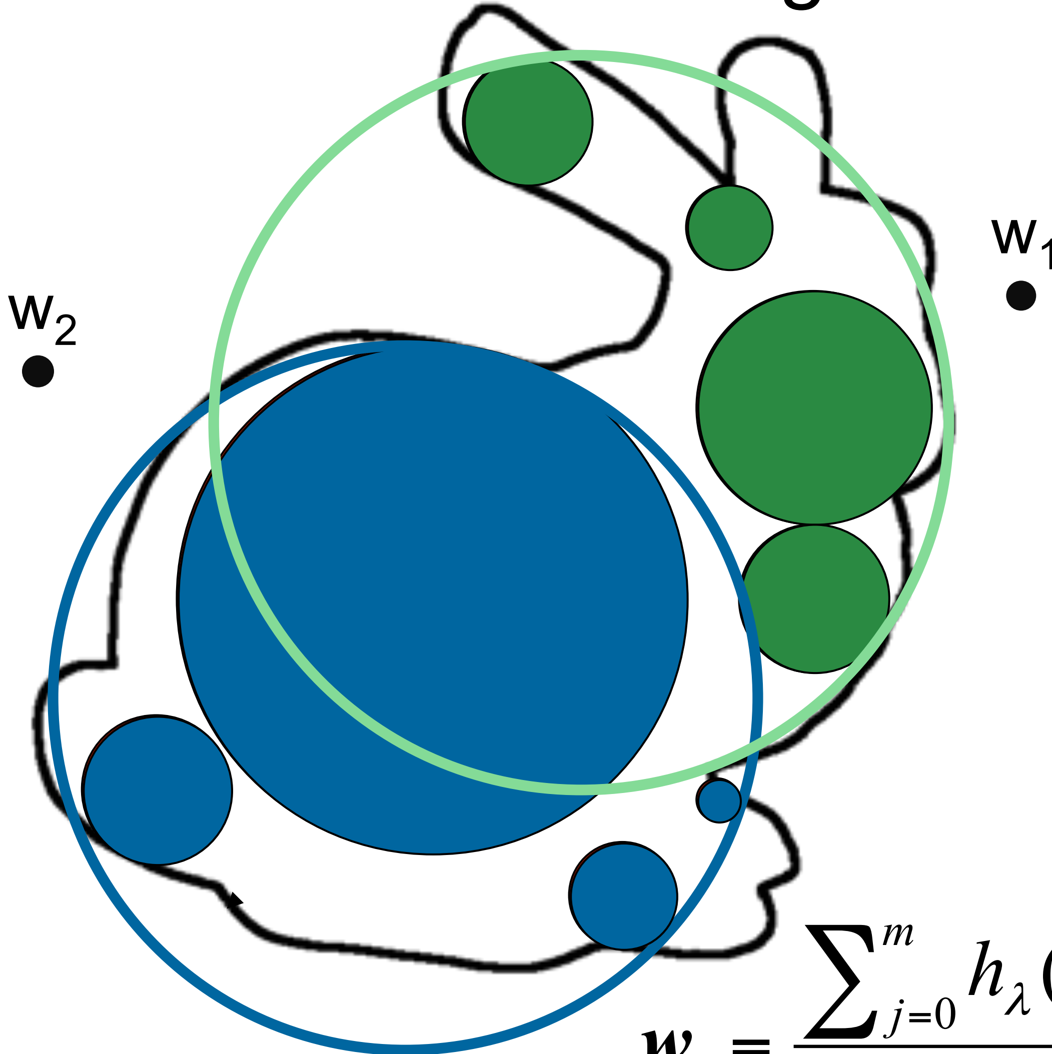
Non-Optimal Subdivision



Batch Neural Gas

- Need an algorithm which considers volume during IST construction
- BNG clustering with a modified cost function which considers volume
- Very robust – independent of initialization of center points.

Batch Neural Gas Clustering

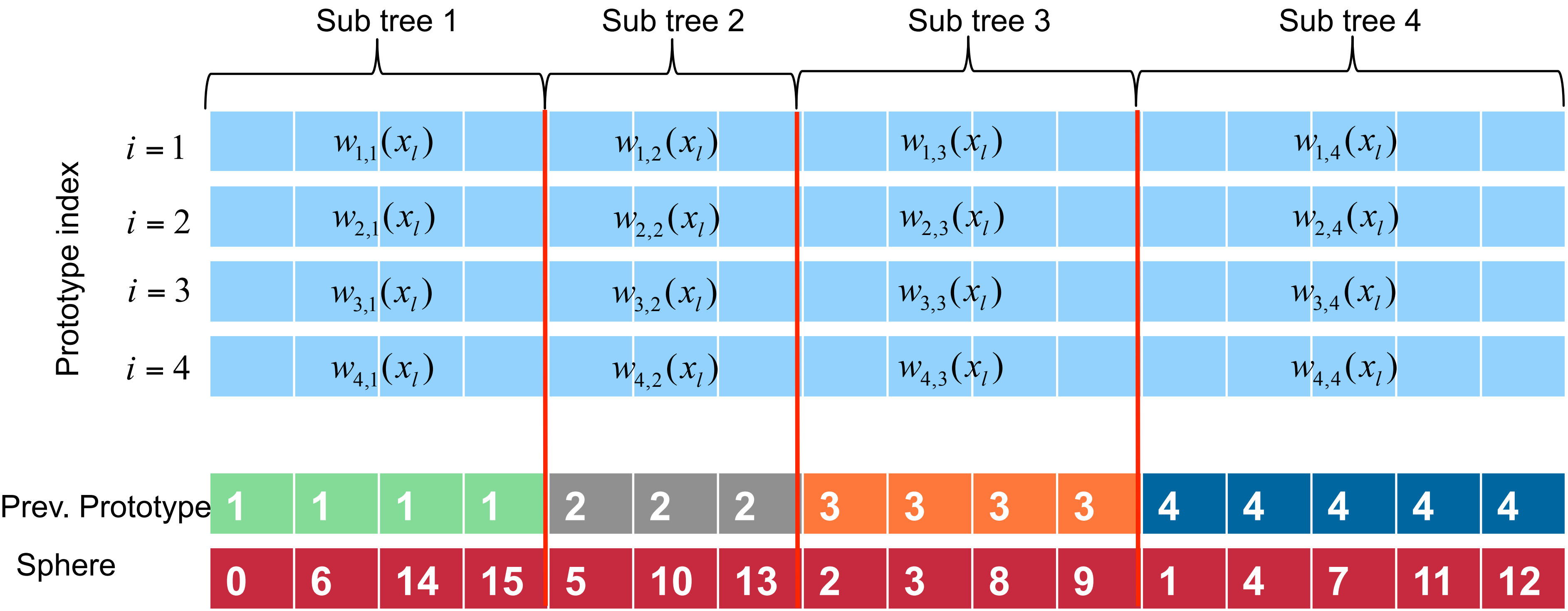


$$\boldsymbol{w}_i = \frac{\sum_{j=0}^{m} h_\lambda(k_{i,j})\, v(\boldsymbol{x}_j)\, \boldsymbol{x}_j}{\sum_{j=0}^{m} h_\lambda(k_{i,j})\, v(\boldsymbol{x}_j)}$$

$$k_{ij} := \left| \{ \boldsymbol{w}_k : d(\boldsymbol{x}_i, \boldsymbol{w}_k) < d(\boldsymbol{x}_j, \boldsymbol{w}_i) \} \right| \in \{0, \ldots, m\}$$

Parallel BNG

- BNG Hierarchy construction has a complexity of $\mathcal{O}(n \log n)$
- BNG perfectly suited for parallelization
- Parallelization in first level of hierarchy is straightforward
  - Ordering of k_{ij} and $h_\lambda(k_{ij})v(x_j)x_j$ can be computed independently for each sphere
  - Summing up → using parallel scan algorithm [SHG08]
  - Distance sphere to all prototypes → assignment sphere to prototypes (a sphere is assigned to exactly one prototype)
- Triggering an own parallel process for each sub-tree is not efficient for parallel processing and is not memory efficient

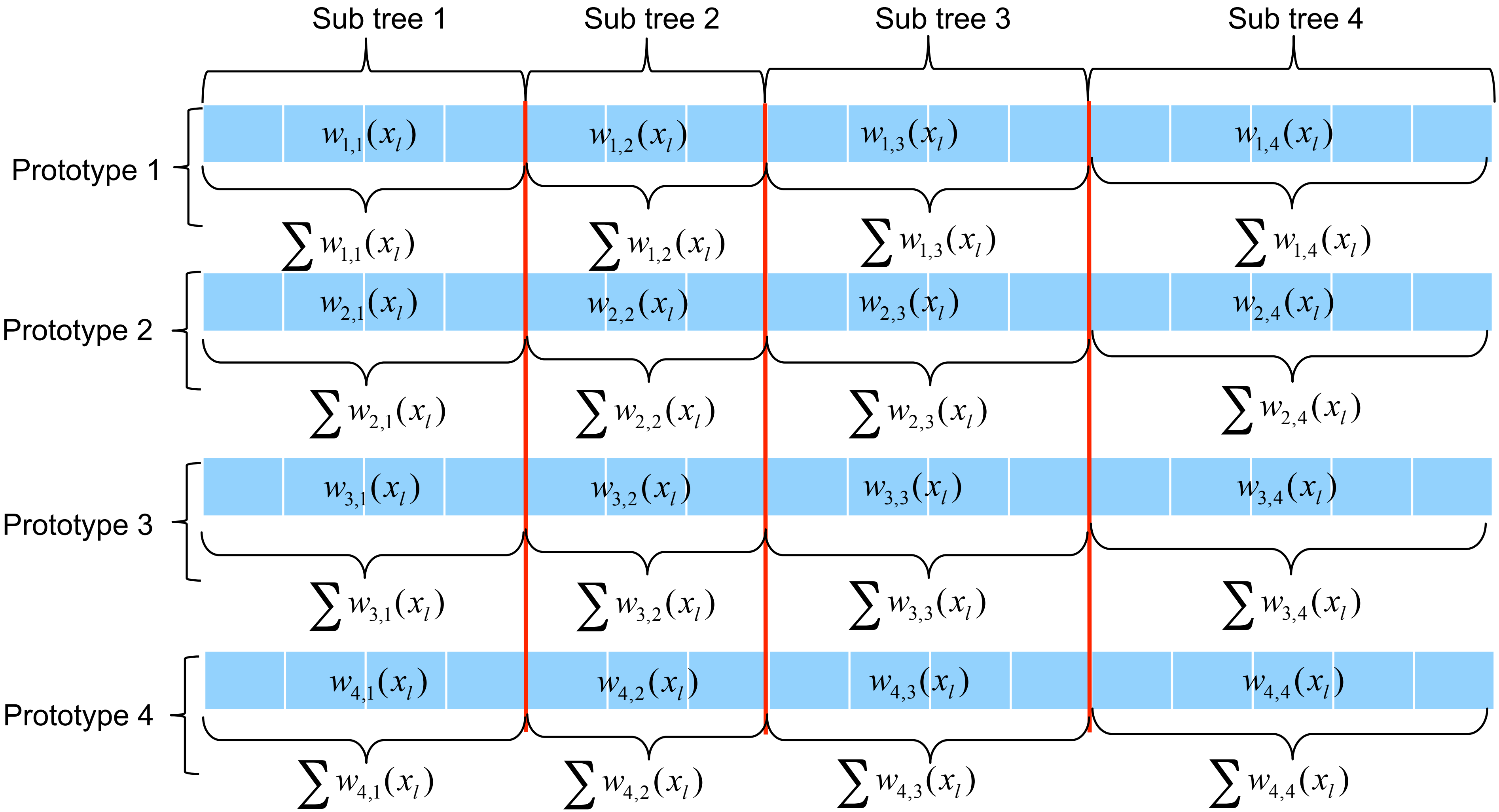


$$w_{i,j}(\boldsymbol{x}_l) = h_\lambda(k_{i,l})v(\boldsymbol{x}_l)$$

Prototypes index

Sub-trees index

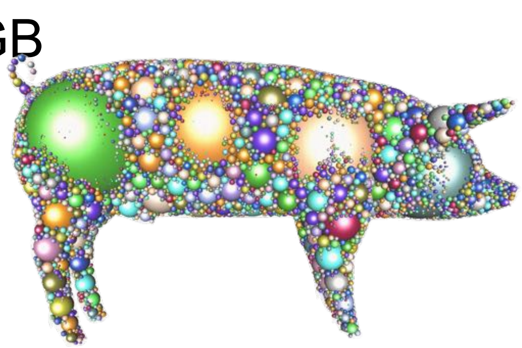
Spheres index

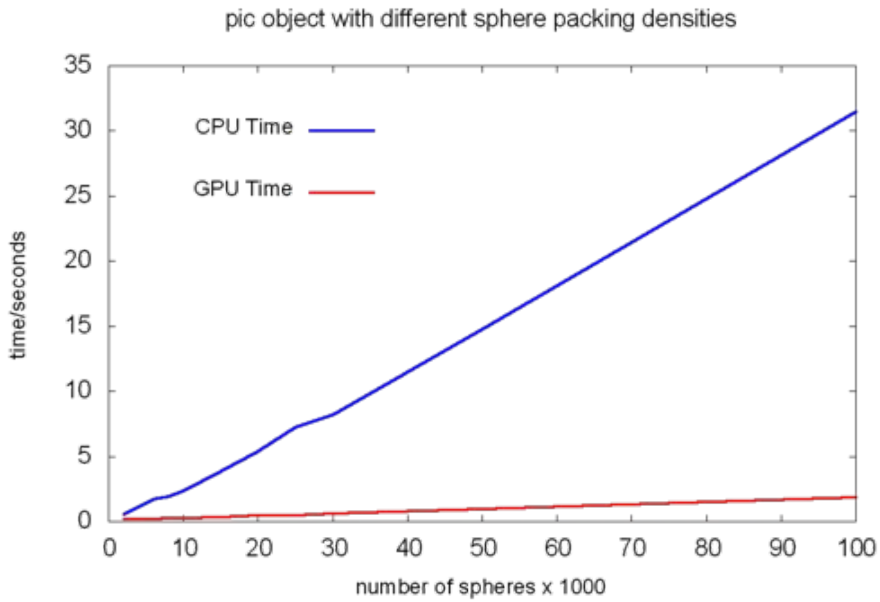

Sub tree 1
Sub tree 2
Sub tree 3
Sub tree 4
Prototype 1
$w_{1,1}(x_l)$
$w_{1,2}(x_l)$
$w_{1,3}(x_l)$
$w_{1,4}(x_l)$
$\sum w_{1,1}(x_l)$
$\sum w_{1,2}(x_l)$
$\sum w_{1,3}(x_l)$
$\sum w_{1,4}(x_l)$
Prototype 2
$w_{2,1}(x_l)$
$w_{2,2}(x_l)$
$w_{2,3}(x_l)$
$w_{2,4}(x_l)$
$\sum w_{2,1}(x_l)$
$\sum w_{2,2}(x_l)$
$\sum w_{2,3}(x_l)$
$\sum w_{2,4}(x_l)$
Prototype 3
$w_{3,1}(x_l)$
$w_{3,2}(x_l)$
$w_{3,3}(x_l)$
$w_{3,4}(x_l)$
$\sum w_{3,1}(x_l)$
$\sum w_{3,2}(x_l)$
$\sum w_{3,3}(x_l)$
$\sum w_{3,4}(x_l)$
Prototype 4
$w_{4,1}(x_l)$
$w_{4,2}(x_l)$
$w_{4,3}(x_l)$
$w_{4,4}(x_l)$
$\sum w_{4,1}(x_l)$
$\sum w_{4,2}(x_l)$
$\sum w_{4,3}(x_l)$
$\sum w_{4,4}(x_l)$

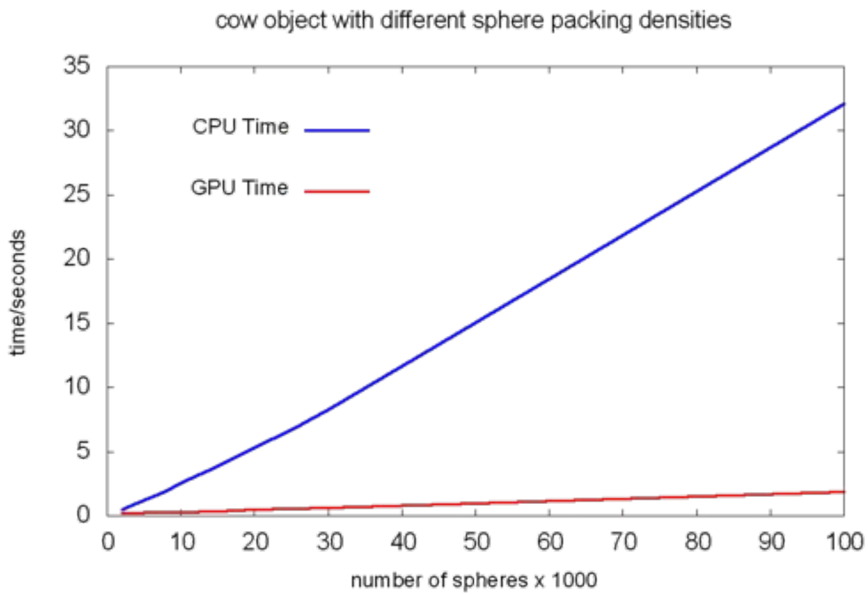
- BNG hierarchy construction on CPU has complexity of $\mathcal{O}(n \log n)$
- Parallelization of BNG reduces complexity to $\mathcal{O}(\log^2 n)$

Results: Hierarchy Construction Performance

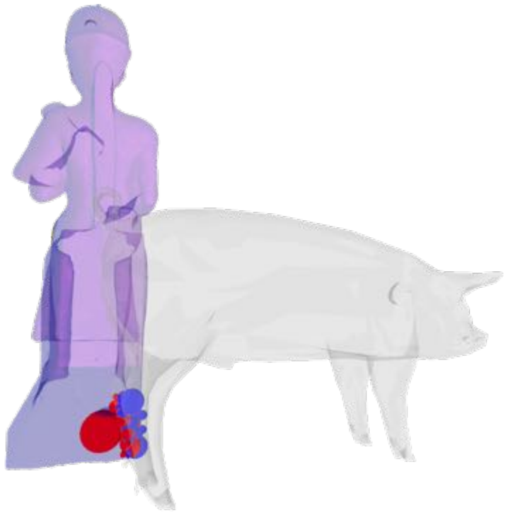
- Intel i7 CPU with 8GB RAM & NVIDIA Geforce GTX 780 with 3GB

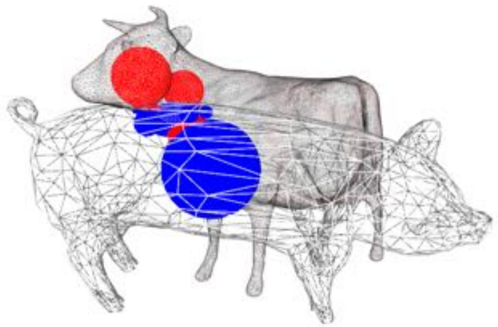


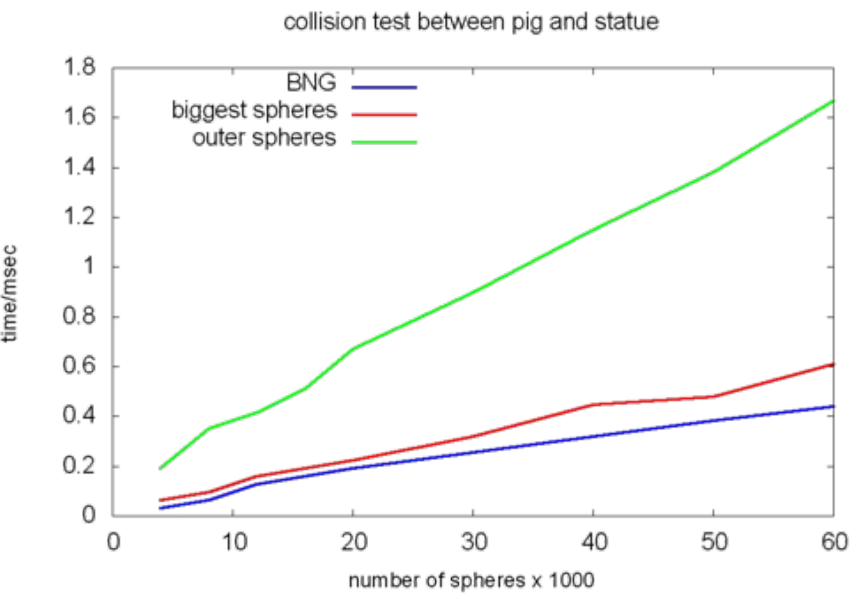



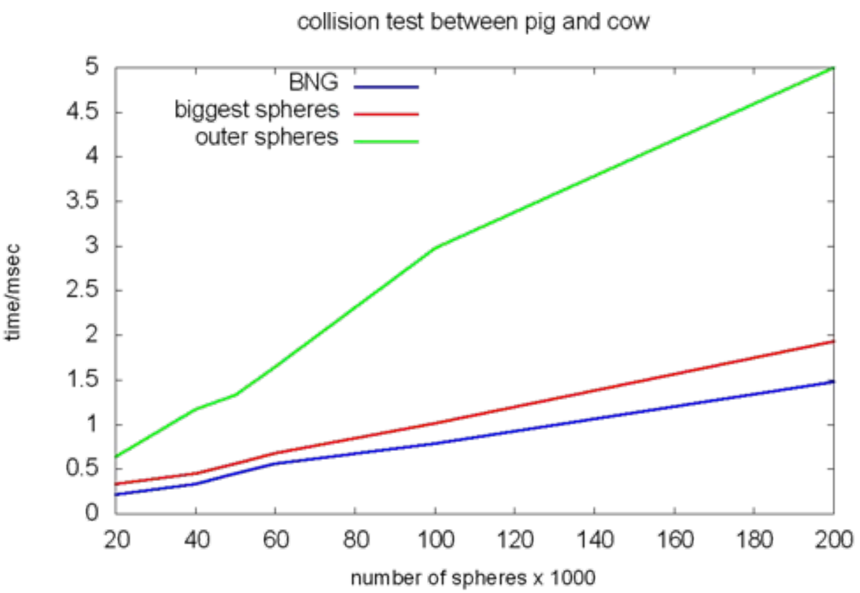


Results: Runtime Performance







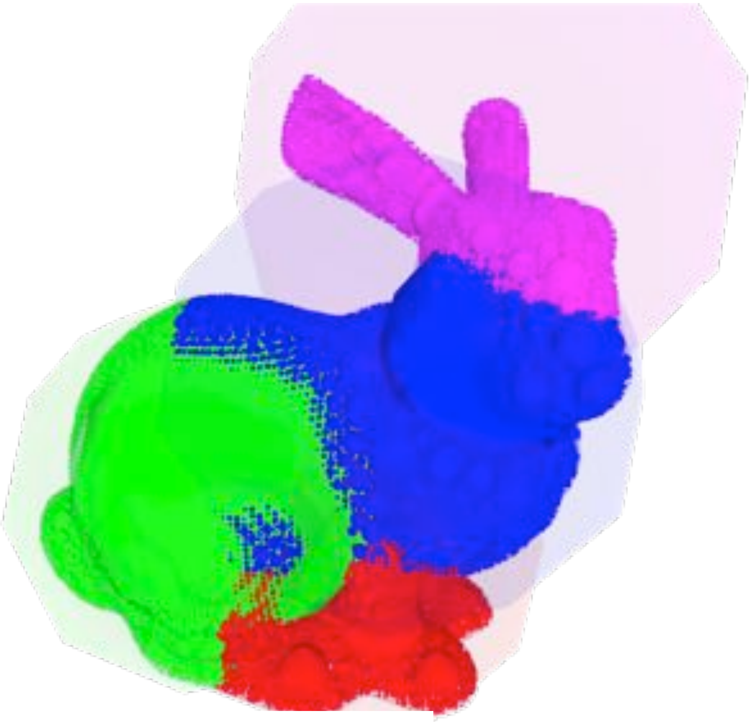


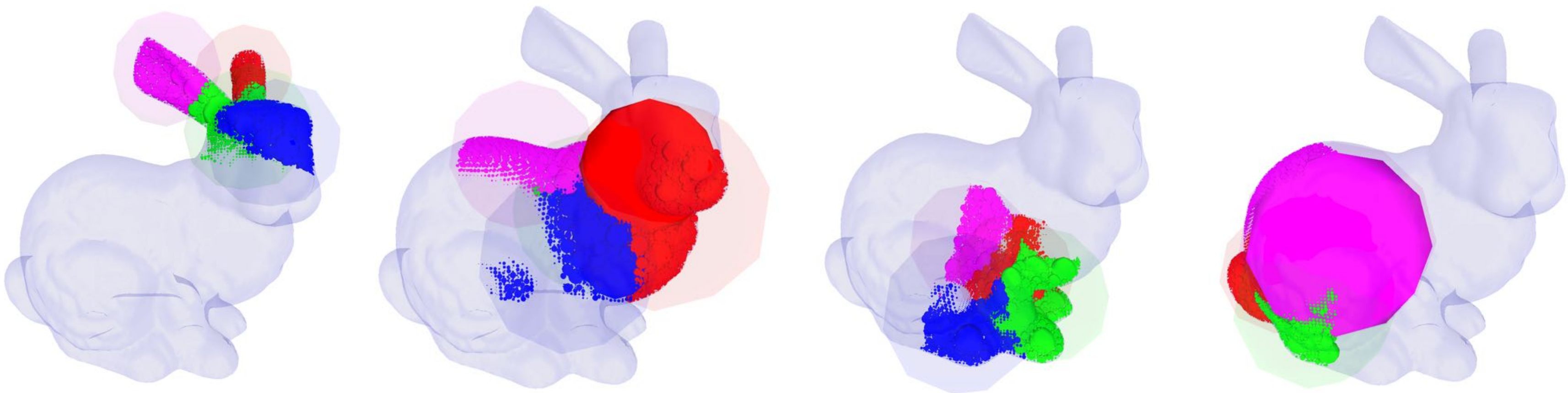
Conclusions/Future Work

- Completely GPU-based method for BVH for volumetric object representations
- Parallel BNG reduce complexity from $\mathcal{O}(n \log n)$ to $\mathcal{O}(\log^2 n)$
- Outperforms CPU version by factor 15
- Better hierarchy : faster than naïve splitting approaches for collision queries
- Our method can also be used with different branching factors

- Apply approach to other volumetric object representations than sphere packing's, e. g. tetrahedral or ellipses
- Use this approach for classical *outer* BVH
- Ray tracing and occlusion culling

Thank you!
Any Questions?





DFG grant TRR 8/3-2013