Bremen

Multi-Objective Packing of 3D Objects into Arbitrary Containers

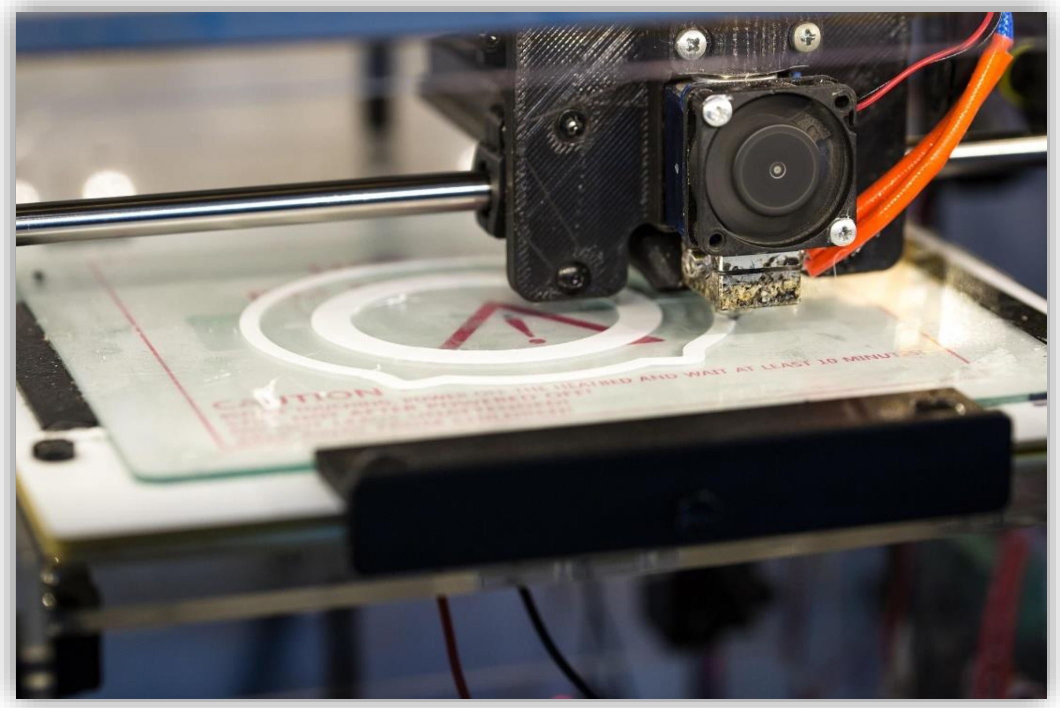
Hermann Meißenhelter, René Weller, Gabriel Zachmann
University of Bremen, Germany

12th-16th May 2025 SE1 8XT
EUROGRAPHICS LONDON 2025

Eurographics 2025, 12-16 May, London, England

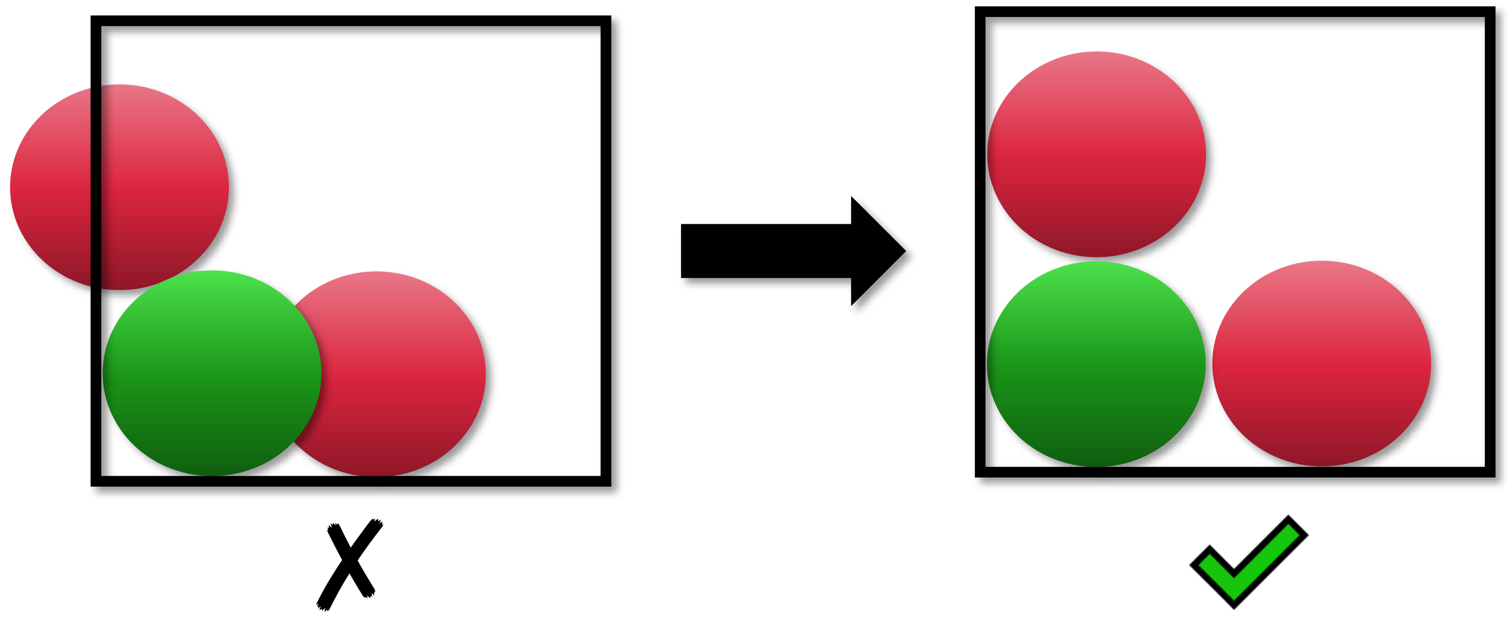
Motivation

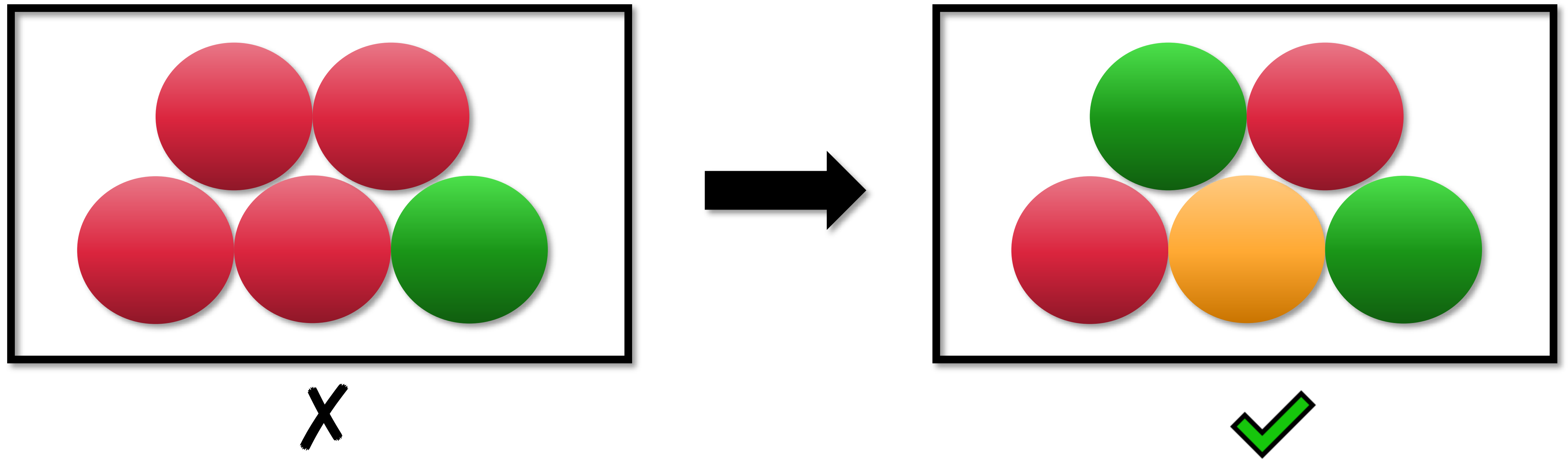
- Logistics: maximize goods per shipment
- Additive manufacturing: batch print diverse parts
- Art & design: visually pleasing arrangements



Problem Statement

- Given a container *C* and object types *T*
- Find rigid placements s.t. objects:
  - Do not overlap & lie inside
  - Maximise packed volume $\sum_{k=1}^{n} \mathrm{vol}(O_{t_i})/\mathrm{vol}(C)$
  - Match target type frequency (e.g. 40% red spheres)
  - Avoid spatial clustering





Related Work

- Knapsack problem and 3D box packing are NP-complete [Lu15]
- Most methods only handle simple objects (boxes/spheres) [Lozano16, Ali22]
- Arbitrary objects (state of the art):

Work	Max. Density	Arbitrary Container	Continuous Rotation	Type Distribution	Uniform Spatial Dist.	Open-source code
Romanova et al. 2018	✓	✗	✓	✗	✗	✗
Ma et al. 2018	✓	✓	✓	✗	✗	✗
Cui et al. 2023	✓	?	✗	✗	✗	✗
Zhuang et al. 2024	✓	?	✓	✗	✗	✗

Our Contribution & Pipeline

- First method to consider:
  - Packing density
  - Arbitrary container shapes
  - User-defined type distribution
  - Uniform spatial type distribution

Open-source code

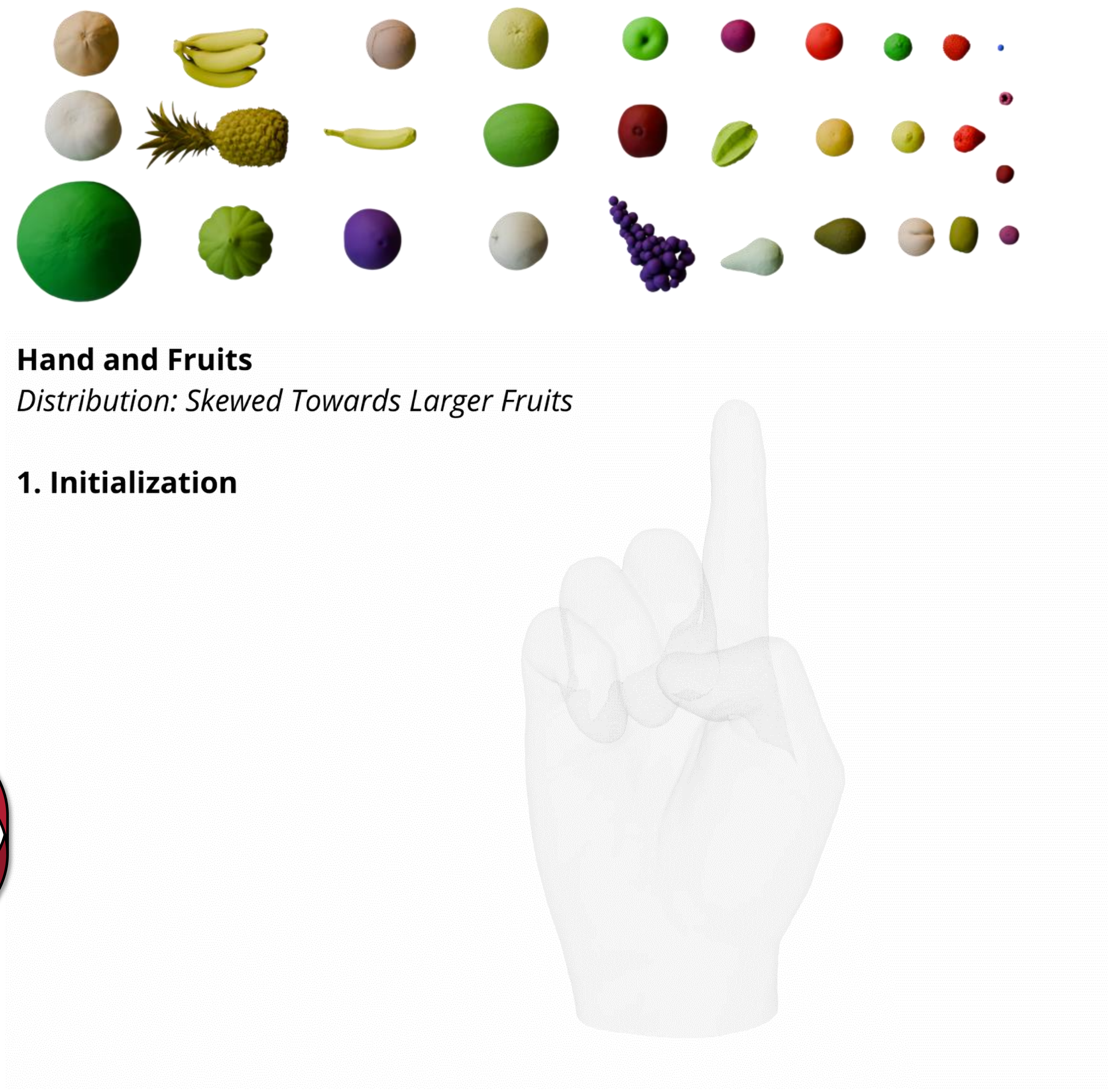
https://cgvr.cs.uni-bremen.de/research/packing

- Two-phase heuristic pipeline: Initialization + Optimization

Empty Packing → Initialization → Feasible Packing → Optimization → Final Packing

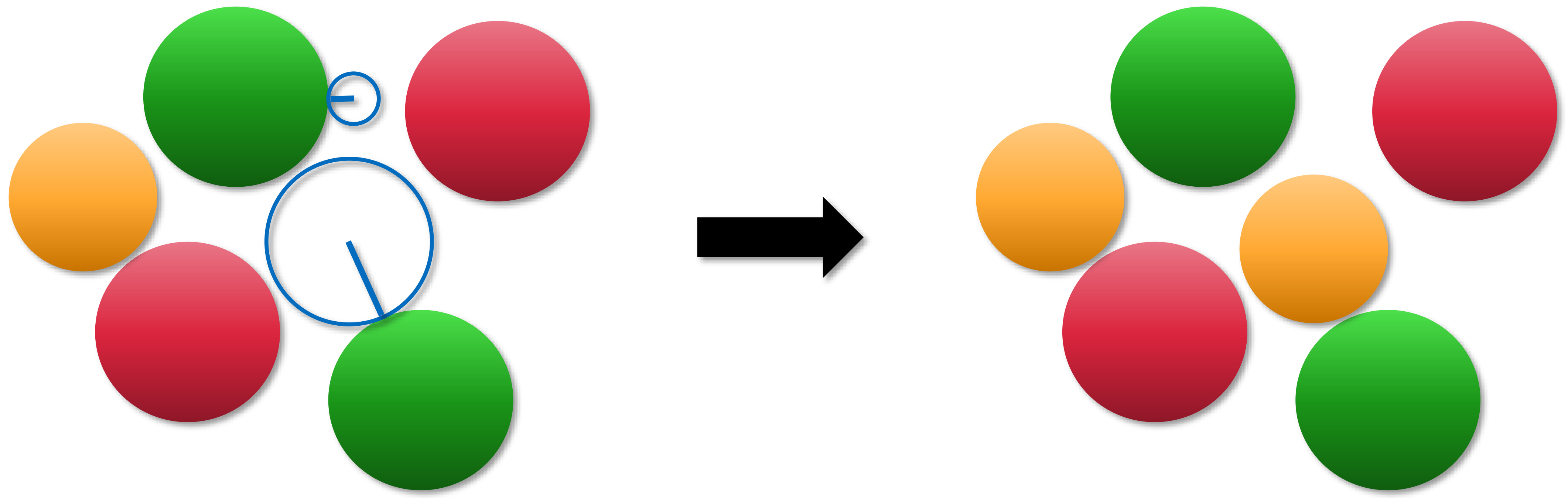
Phase 1 – Initialization: “Place & Grow”

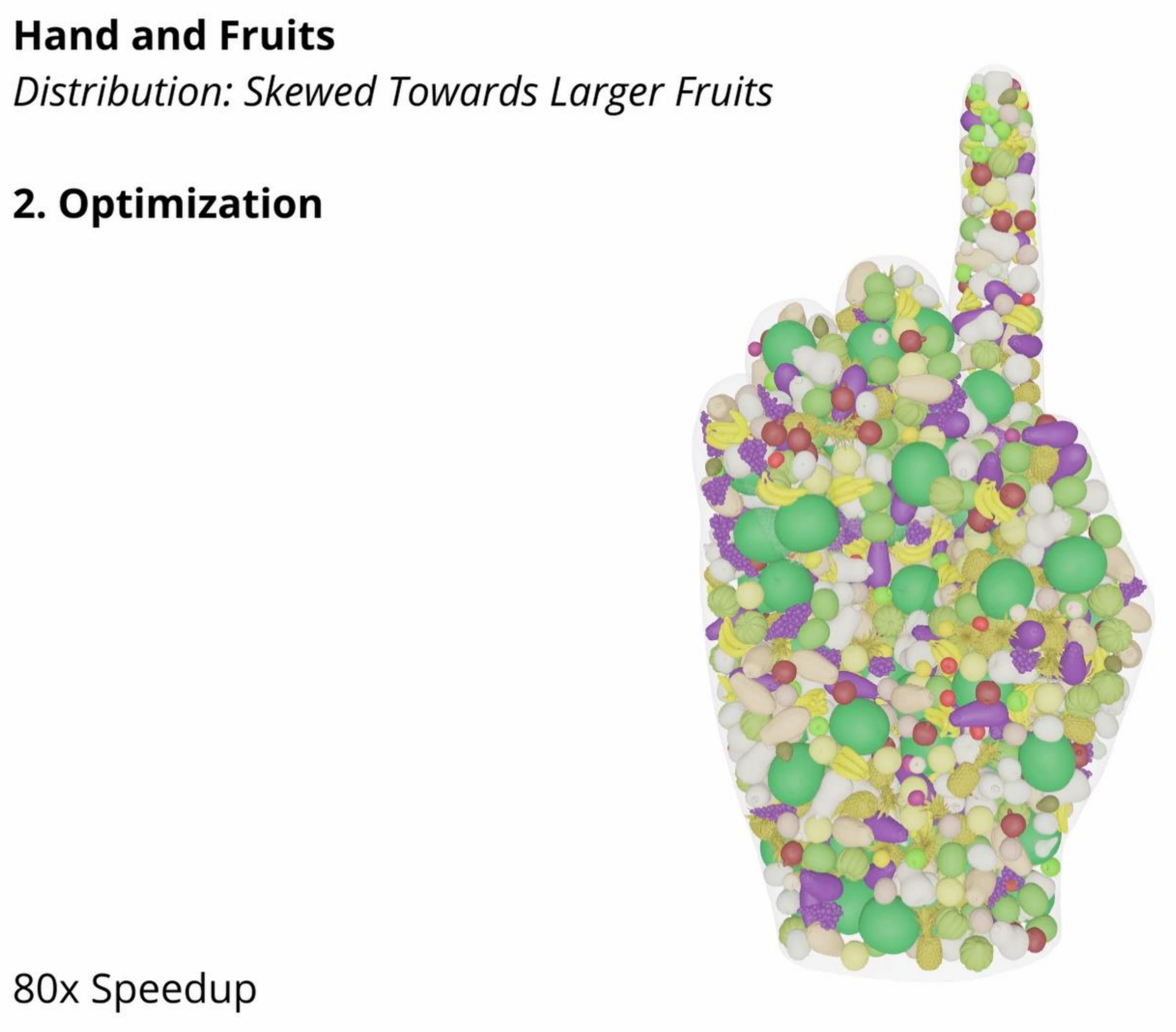
1. Regularly sample container *C*
2. Place shrunken objs. with random rotation, while matching type distribution
3. Select (largest) type greedily and respect target frequency
4. Incrementally scale this type while resolving overlaps



Phase 2 – Optimization: “Fill Cavities”

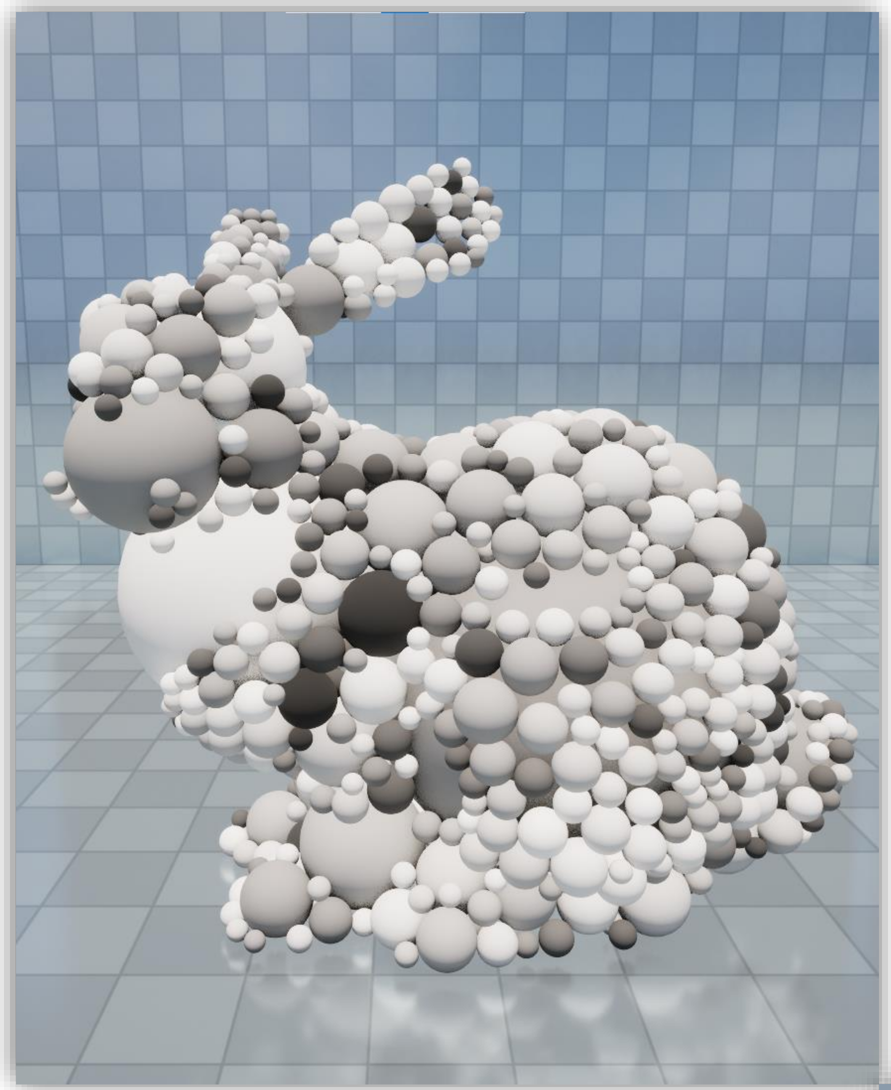
1. High-res sampling
   - Disregard voids with minimal distance below threshold d
   - d is derived from given distribution
2. Insert largest type greedily
3. If failed: 2 retries with smaller objects (density vs. distrib. tradeoff)





Fast Overlap Test & Resolving

- Use BVHs for fast overlap test:
  - Sphere packing BVH for object–object [Weller09]
  - k-DOP tree for object–container [Zachmann98]
  - SIMD-accelerated
- Use custom rigid body simulator
  - Penalty forces to separate objects
  - Limit max. #iterations
- Save last collision-free state to roll back



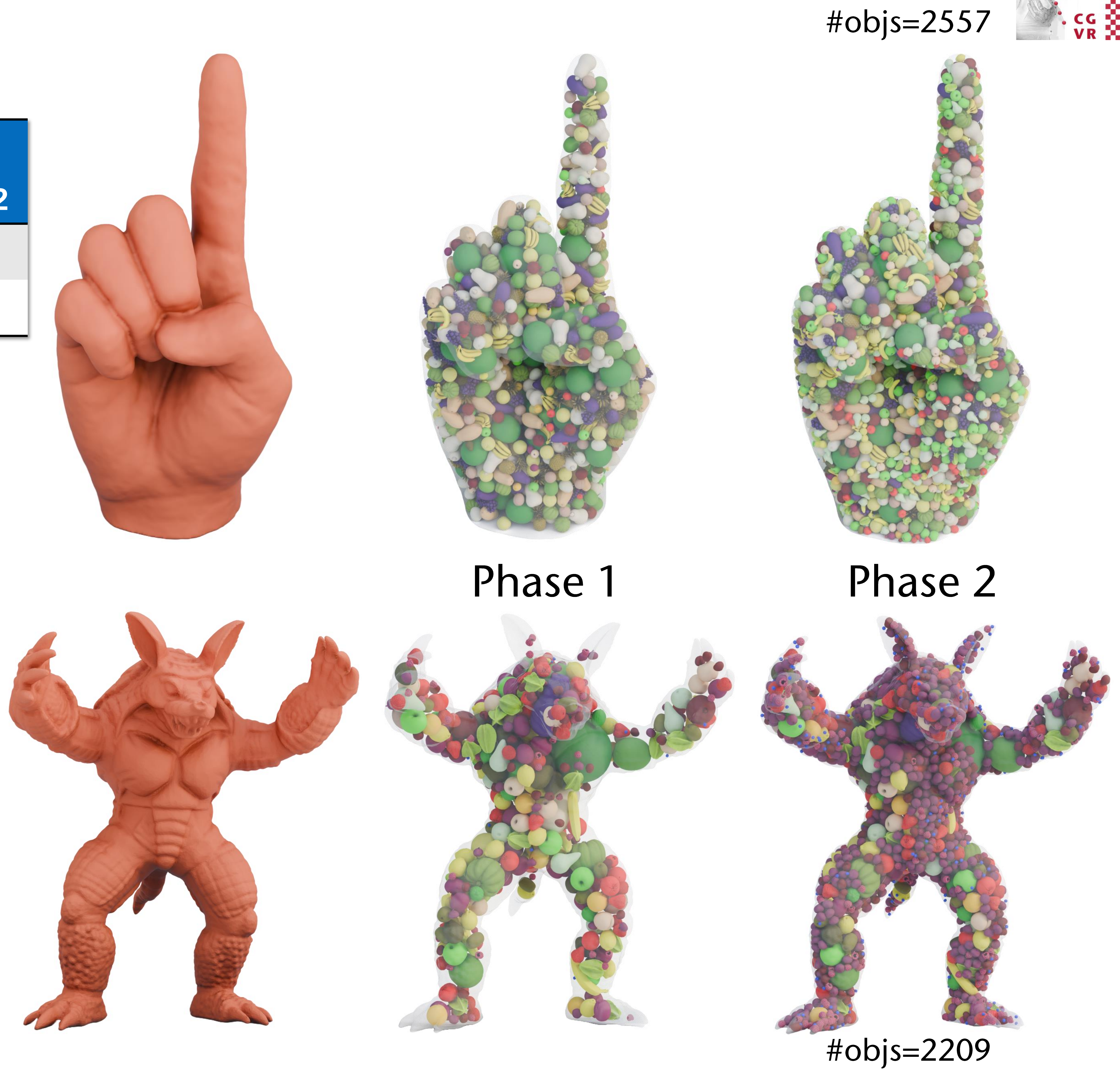
Sphere packing

Triangle mesh

Key Results

Case	Time	Density	TVD Phase1	TVD Phase2	NNI Phase1	NNI Phase2
Hand	33min	54.1%	0.02	0.31	1.25	1.24
Armadillo	23min	54.0%	0.15	0.54	1.43	1.20

- Total variation distance (TVD): measure deviation from predefined distrib., $0 \leq \text{TVD} \leq 1$
- Nearest neighbor index (NNI): measures clustering
  - Clustering NNI ≤ 1
  - Regular NNI ≥ 1



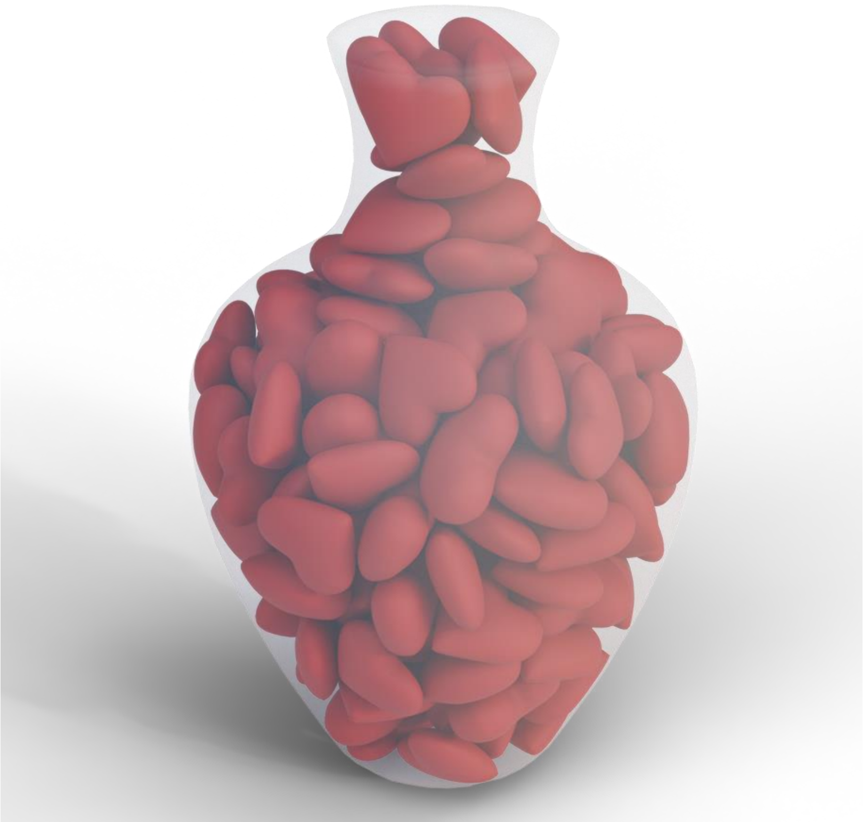
Performance Comparison

Vase	Time	Density	TVD	NNI
[Ma18]	1740s	40.3%	✗	✗
Ours	63s	51.7%	0.09	1.41

Torus	Time	Density	TVD	NNI
[Ma18]	3480s	19.6%	✗	✗
Ours	112s	19.6%	0.00	1.25

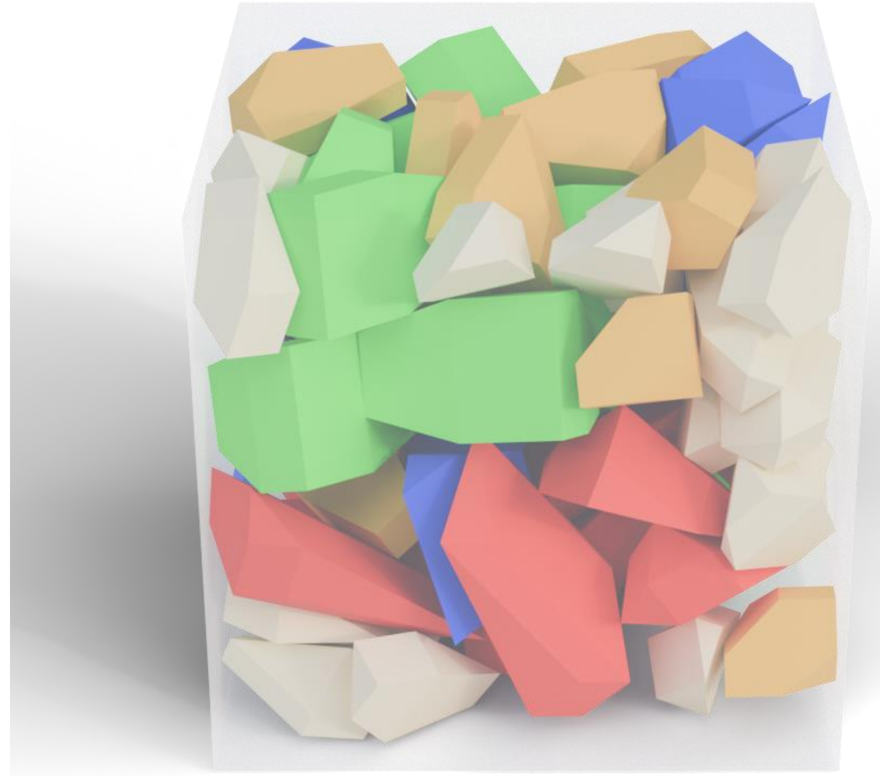
Chess	Time	Density	TVD	NNI
[Ma18]	2400s	32.5%	✗	✗
Ours	112s	32.1%	0.00	1.25

Polyhedron	Time	Density	TVD	NNI
[Romanova18]	42950s	53.66%	✗	✗
[Ma18]	2700s	51.30%	✗	✗
[Cui23]	329s	51.27%	✗	✗
[Zhuang24]	277s	54.47%	✗	✗
Ours	139s	45.50%	0.18	1.14


#objs=135


#objs=62


#objs=67


#objs=78

Conclusion and Future Work

- Two-phase heuristic pipeline for packing into arbitrary container:
  - Predefined type distribution and uniform spatial layout
  - Packing thousands objects in minutes

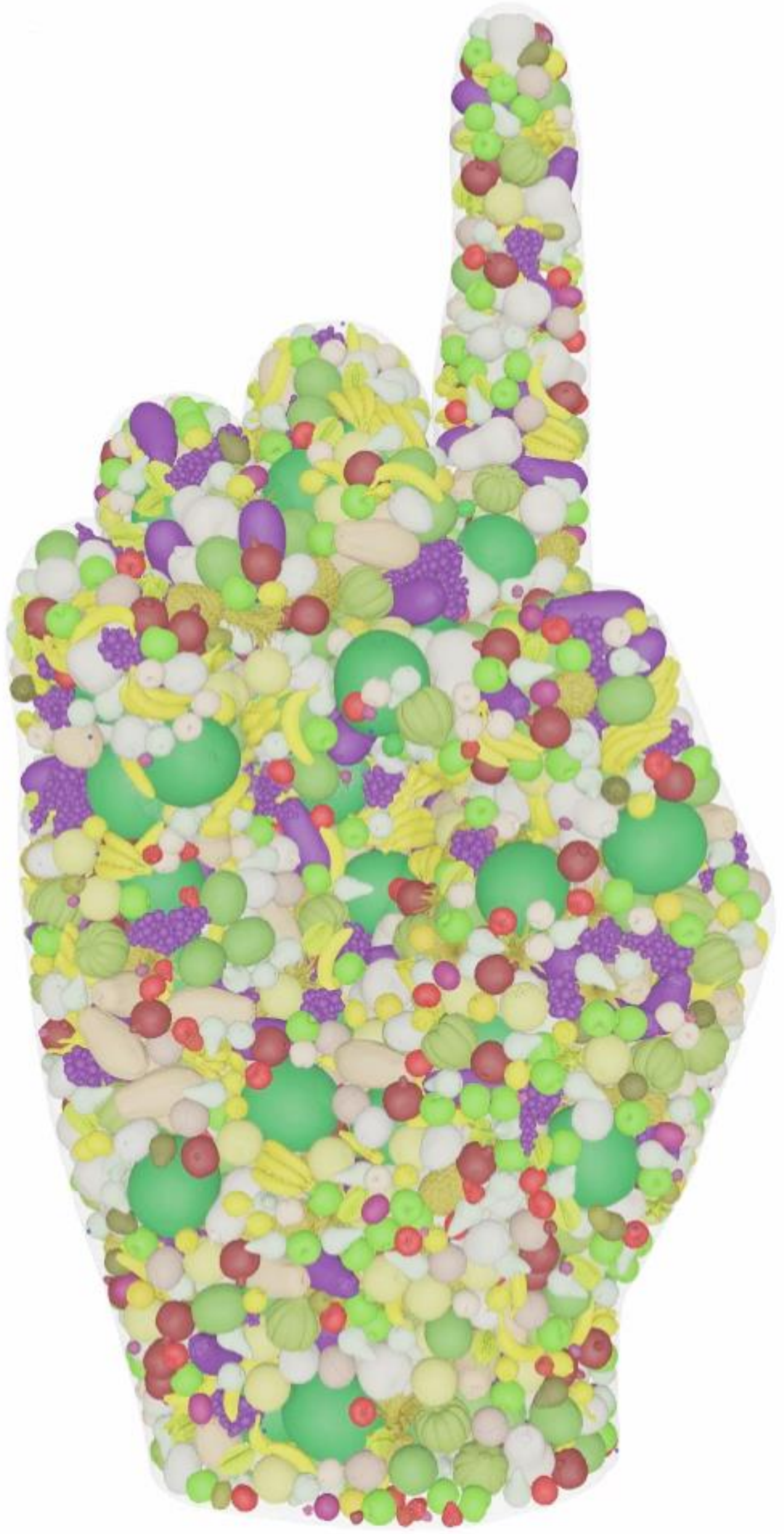
- Next steps
  - Attach free objects to nearest object
  - Run optimization multiple times with increasing sampling resolution

Thank You for Your Attention!

Packing app runs in your browser (also on mobile)

1. Click Init 2. Click Run
Change Scene/Config

https://cgvr.cs.uni-bremen.de/research/packing/app



{meissenhelter, weller, zach}@cs.uni-bremen.de