Stochastic Subsets for BVH Construction

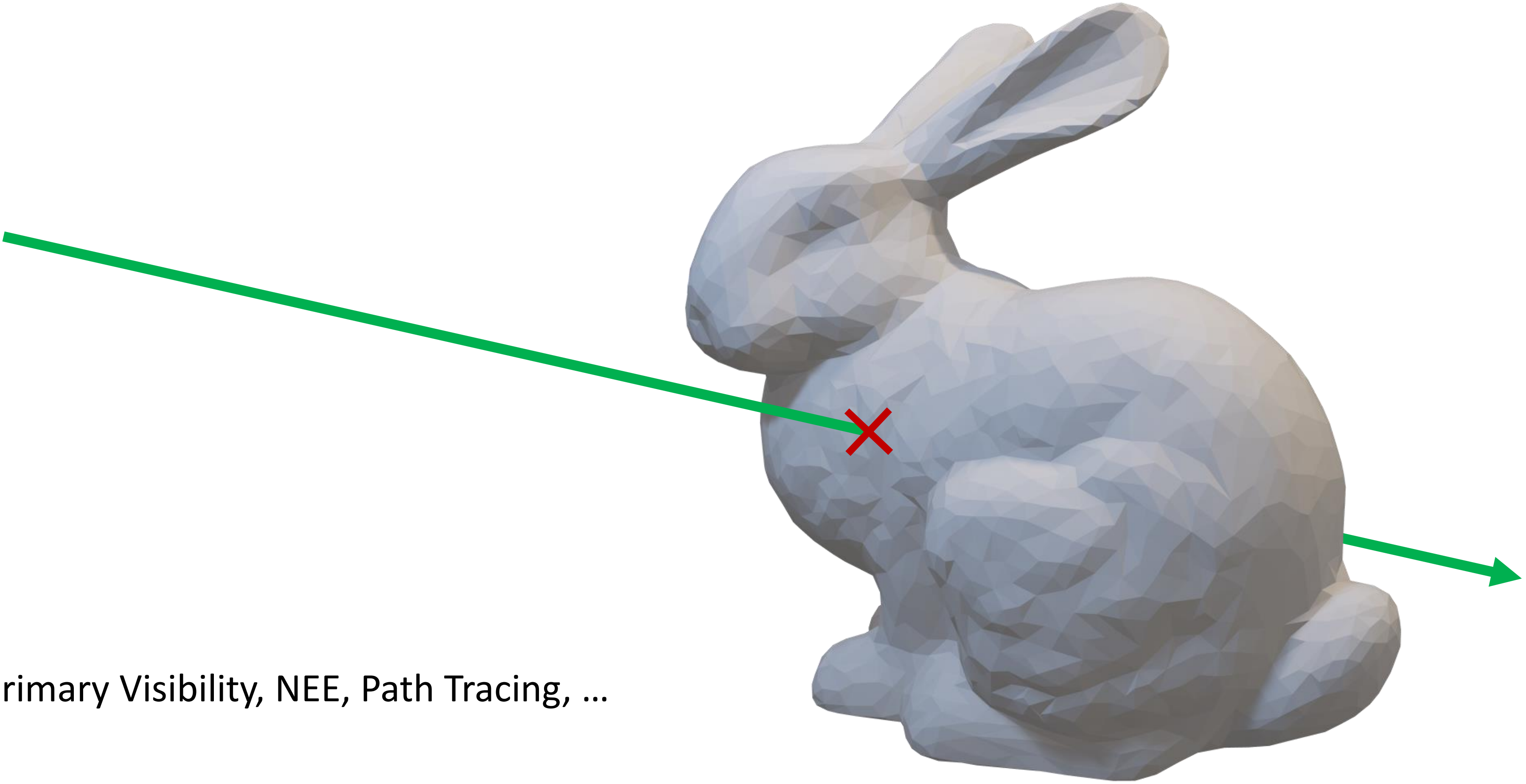
Lorenzo Tessari* [1], Addis Dittebrandt* [1, 2], Michael Doyle[1], Carsten Benthin[1]

[1]Intel, [2]KIT

EUROGRAPHICS 2023

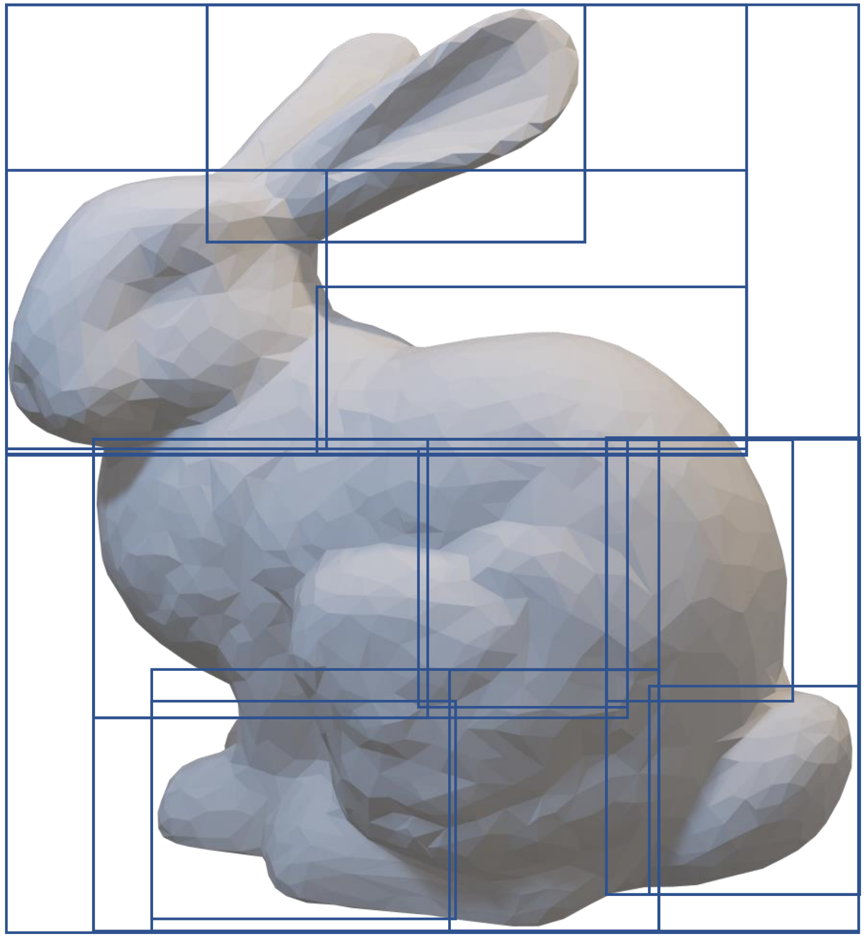
*joint first authorship

Ray Tracing

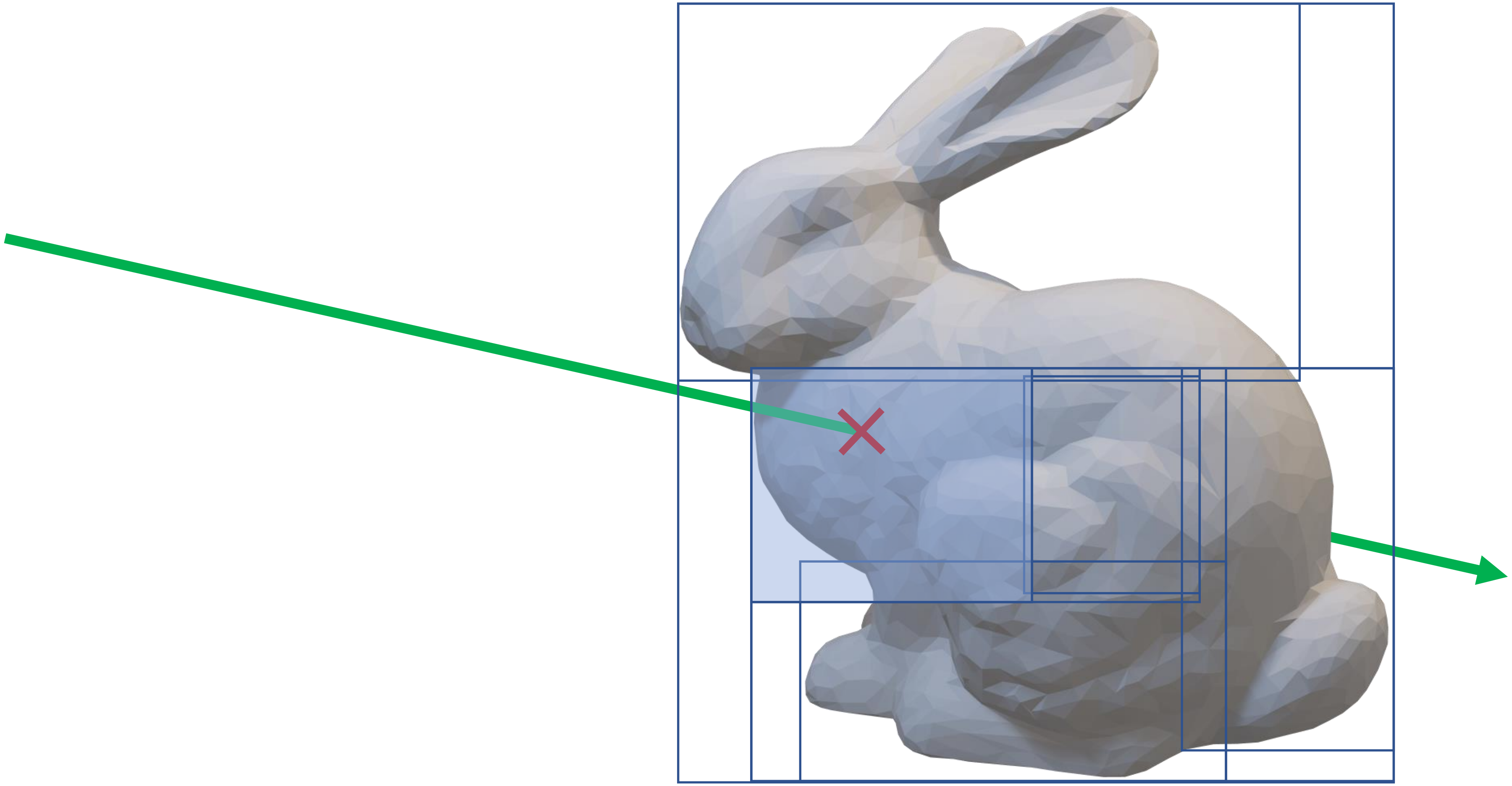


Use cases: Primary Visibility, NEE, Path Tracing, …

Bounding Volume Hierarchies (BVHs)

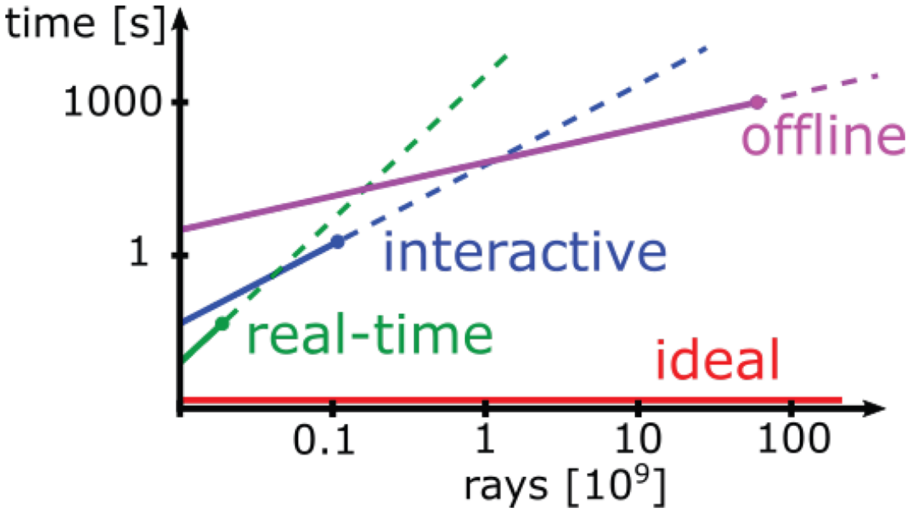


BVH Traversal



BVH Construction

- Tradeoff between build time & quality/traversal efficiency
- General goal: Better quality in less build time
- Targeted for GPUs

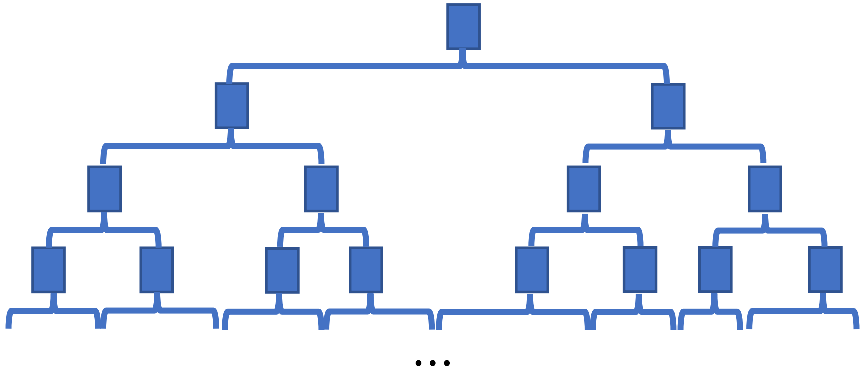


[Meister et al. 2021]

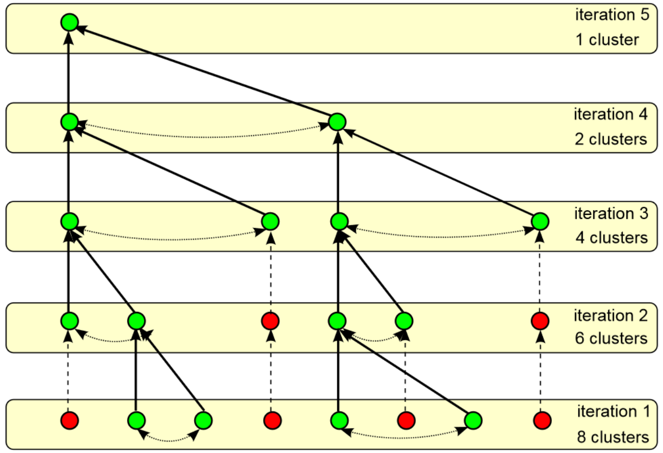
Previous Work

- Top-Down
  - Binning [Wald 2007]
  - Progressive Refinement [Jakub et al. 2017]
- Bottom-Up
  - Agglomerative clustering [Walter et al. 2008]
  - Parallel locally ordered clustering (PLOC) [Meister and Bittner 2017]
  - PLOC++ [Benthin et al. 2022]

Top-Down



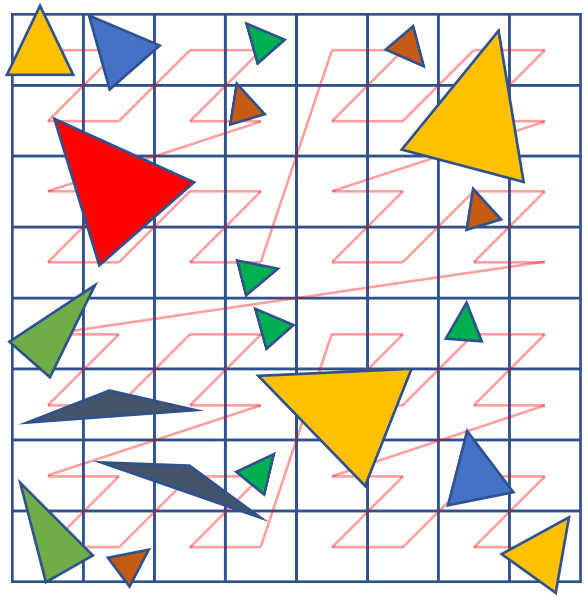
PLOC / PLOC++



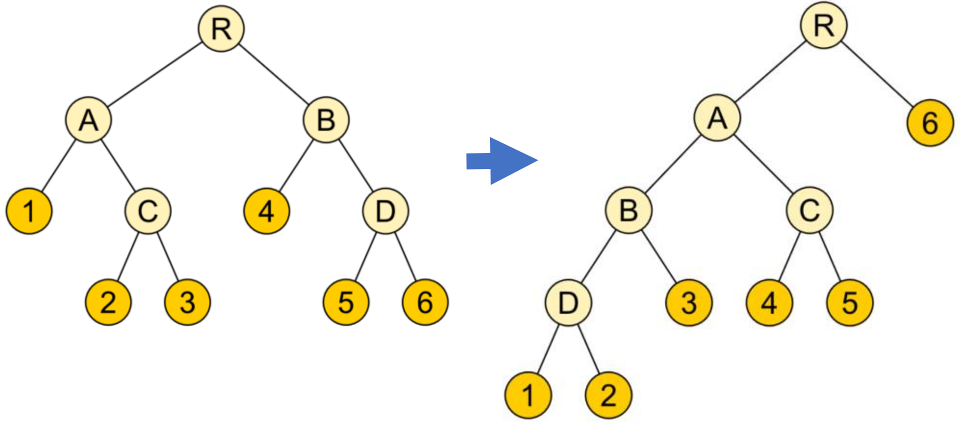
Previous Work

- Incremental construction
  - Greedy [Goldsmith and Salmon 1987]
  - Online [Bittner et al. 2015]
- Linear BVH [Lauterbach et al. 2009, Karras 2012]
- Topological optimization
  - Treelet restructuring (TRBVH / ATRBVH) [Karras and Aila 2015, Domingues and Pedrini 2015]
  - Parallel reinsertion [Meister and Bittner 2018]

LBVH

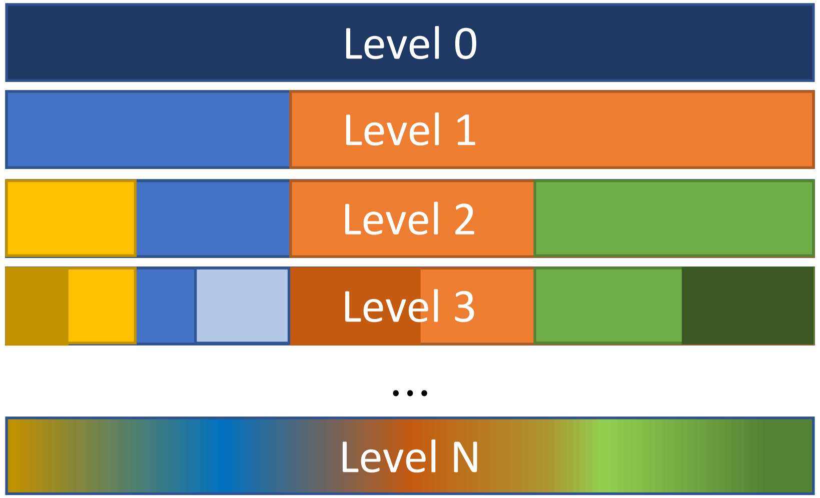


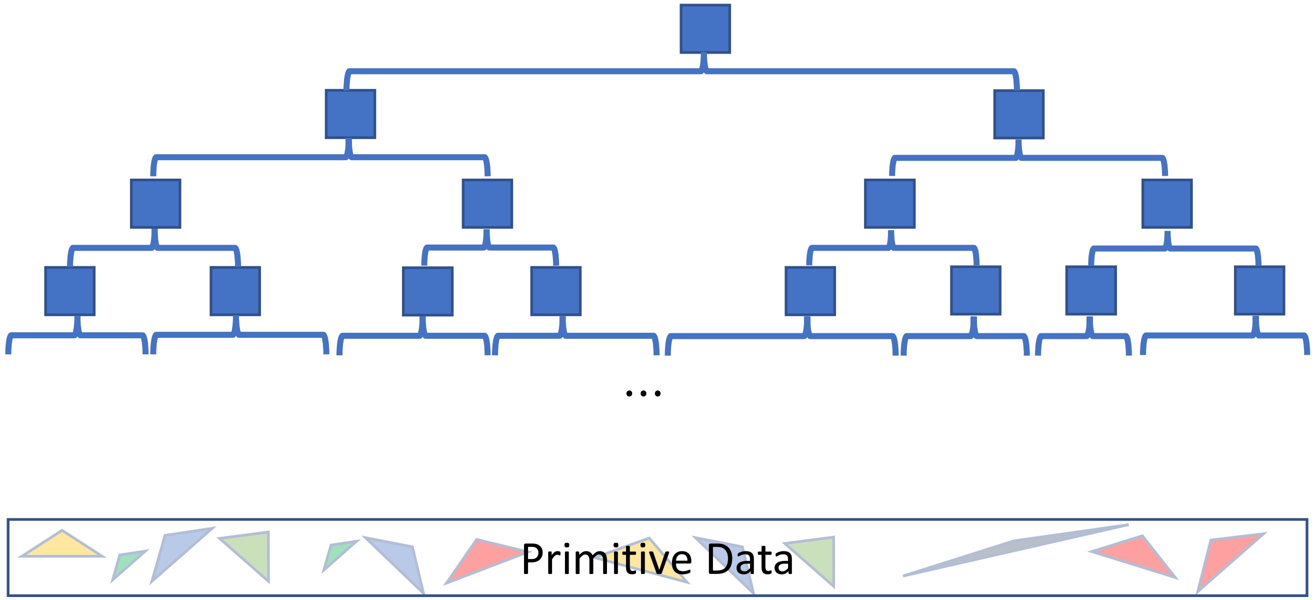
TRBVH / ATRBVH



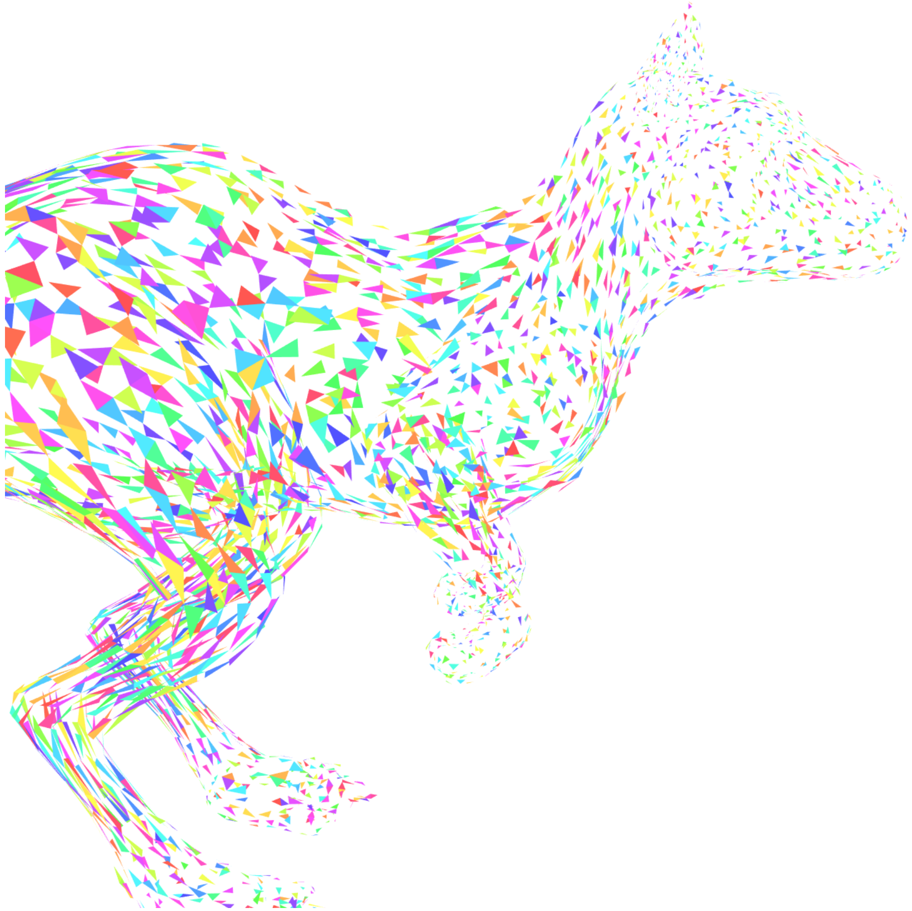
Top-Down Construction

- Superior quality to other build algorithms [Aila et al. 2013]
- Problem: Repeated access of all primitives $O(N \log N)$





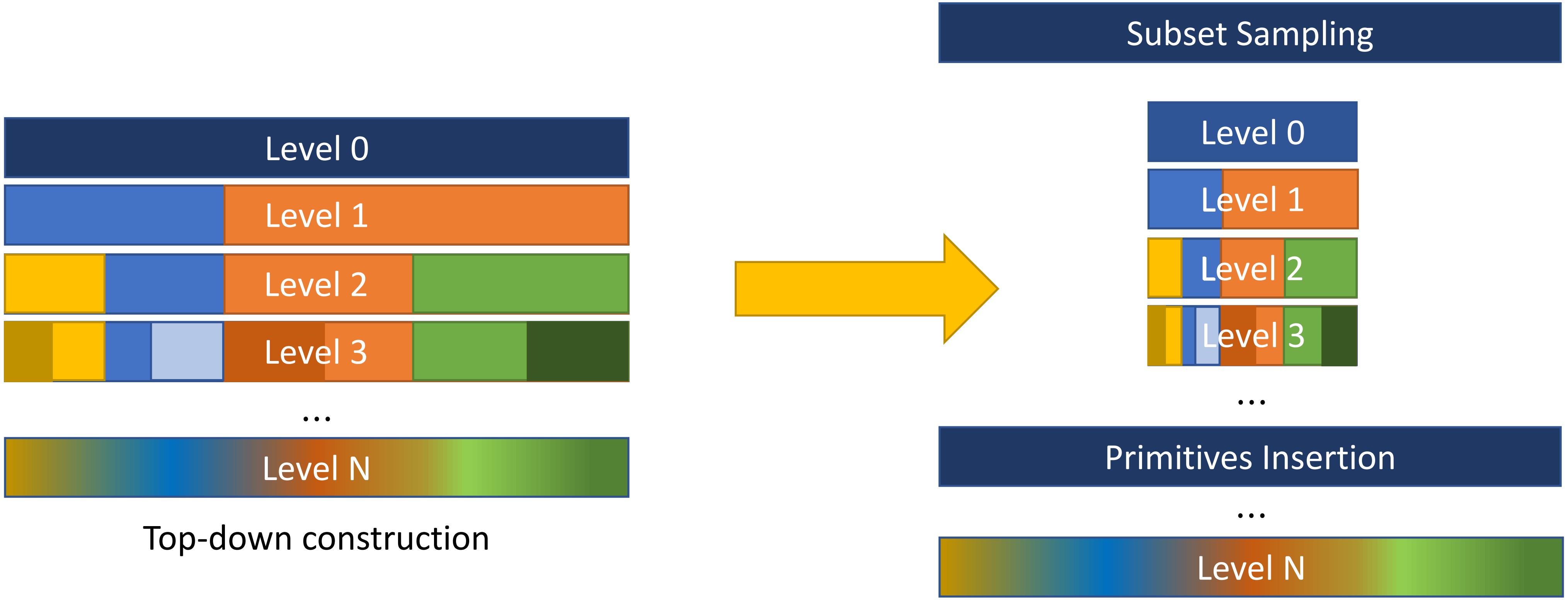
Key Concept: Stochastic Subsets



- Primitive subset can preserve high-level structure of a mesh
- Needs to be chosen in a representative fashion (e.g., stochastically)
- Top-levels of BVH show aggregated behavior of the data

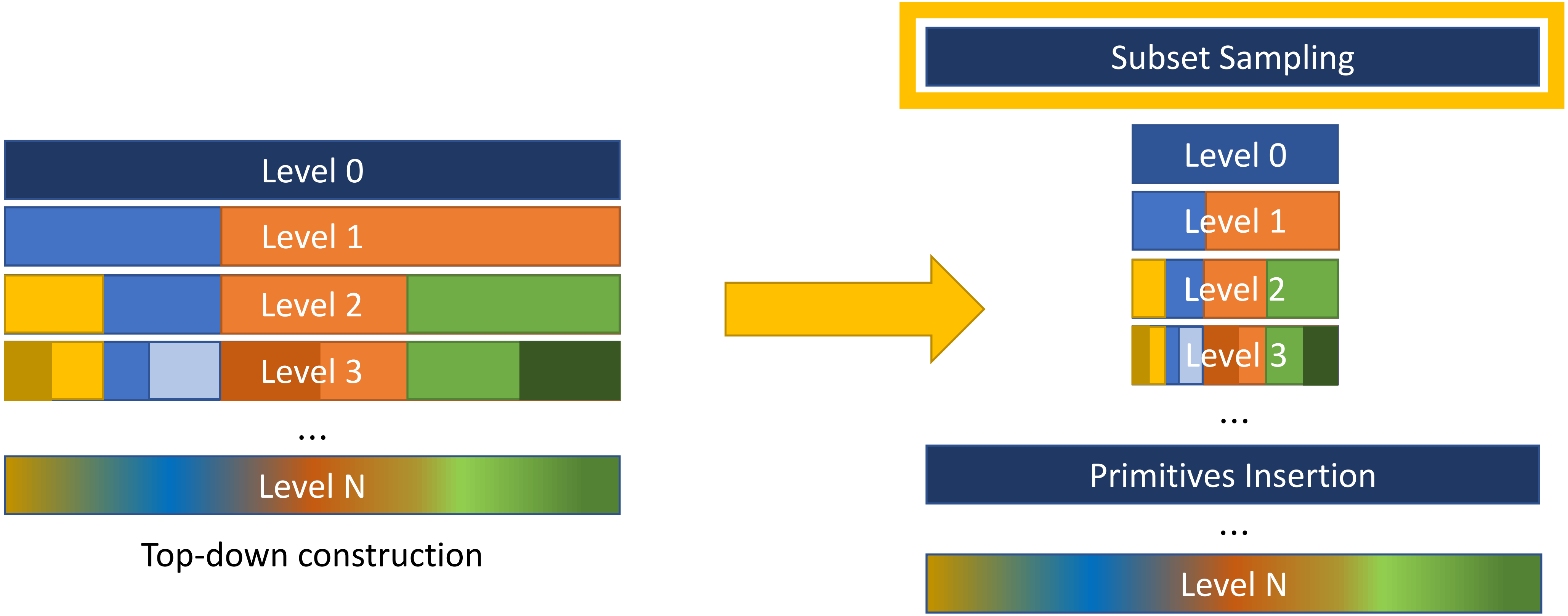
Stochastic Subset BVH Construction

Idea: Operate on subset for first levels to achieve speedup

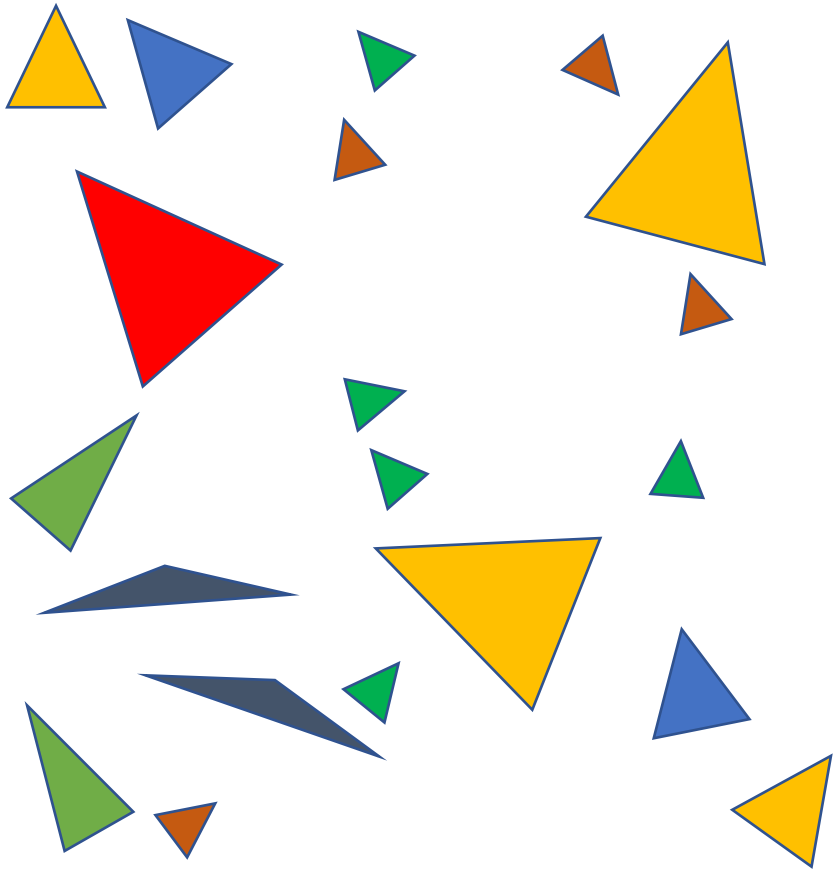


Stochastic Subset BVH Construction

Idea: Operate on subset for first levels to achieve speedup



Subset Sampling

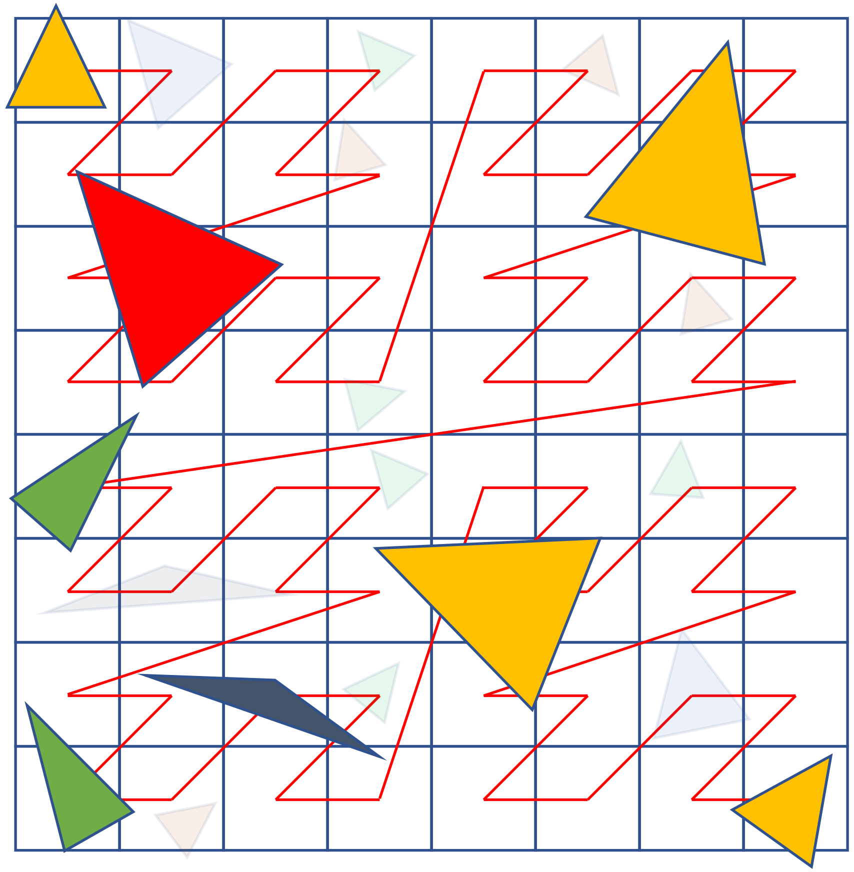


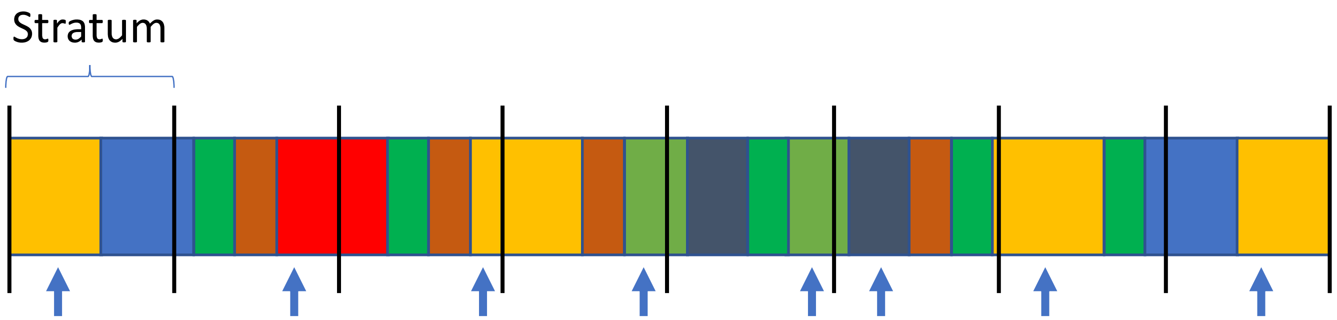
Requirements

- Follow primitive distribution
- Selection proportional to primitive size
- Guaranteed selection of large primitives

→ Combination of **Spatial Ordering**, **Importance Sampling** & **Stratification**

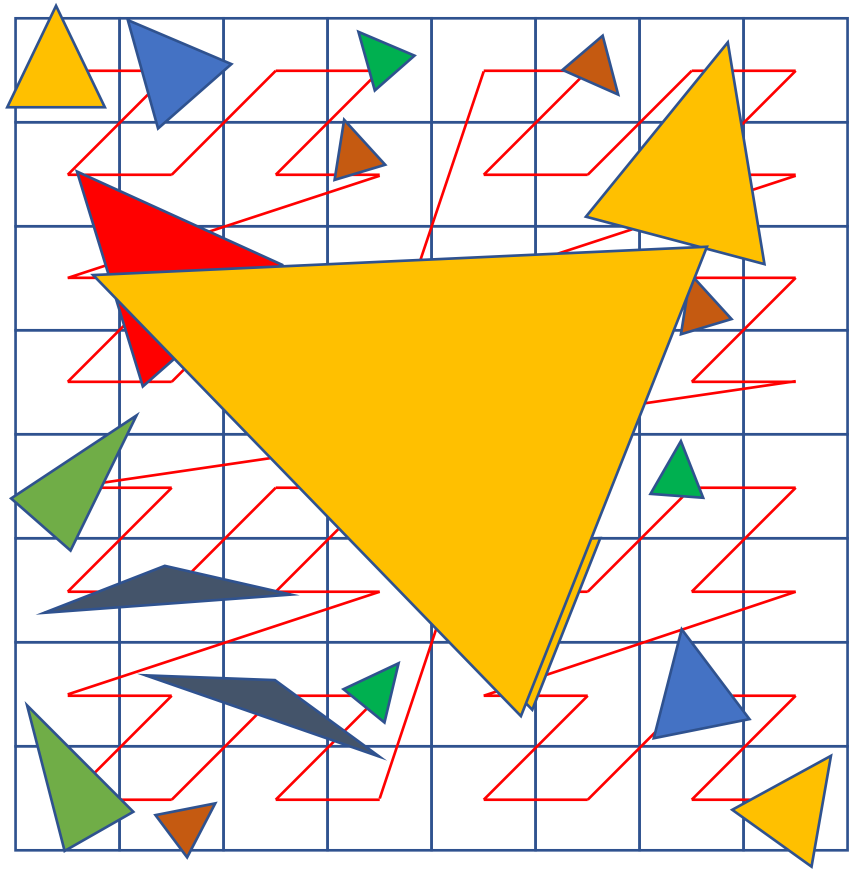
Subset Sampling

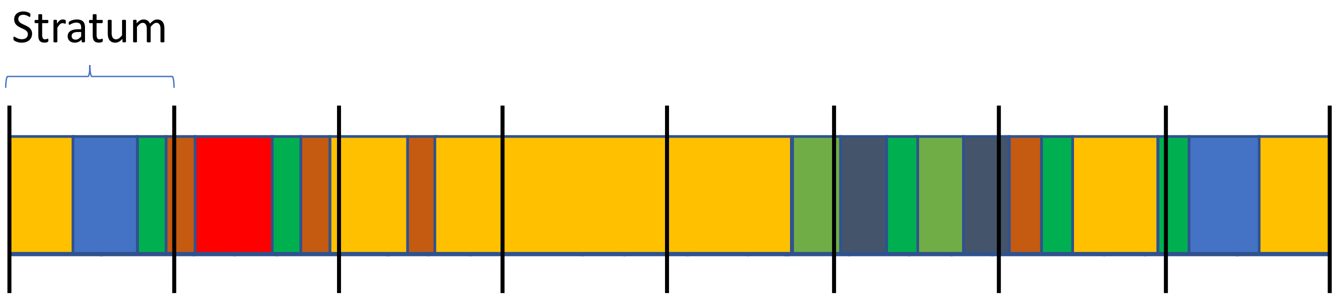




→ Combination of **Spatial Ordering**, **Importance Sampling** & **Stratification**

Subset Sampling





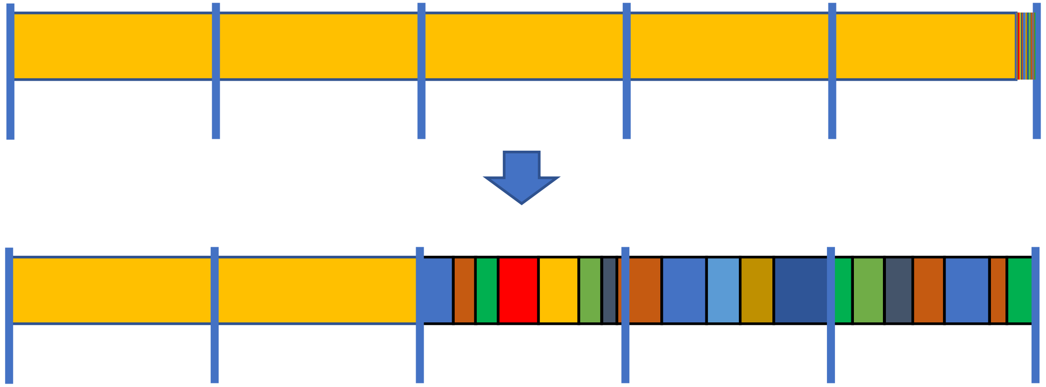
→ Combination of **Spatial Ordering**, **Importance Sampling** & **Stratification**

Subset Sampling

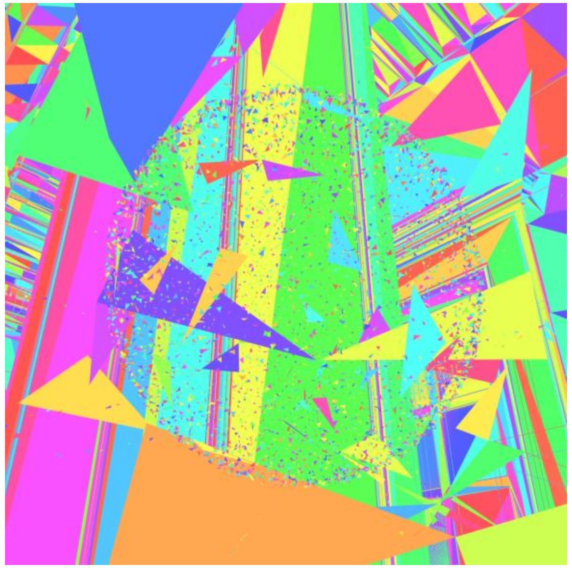
In the paper:

- Weight clampling
- Defensive sampling

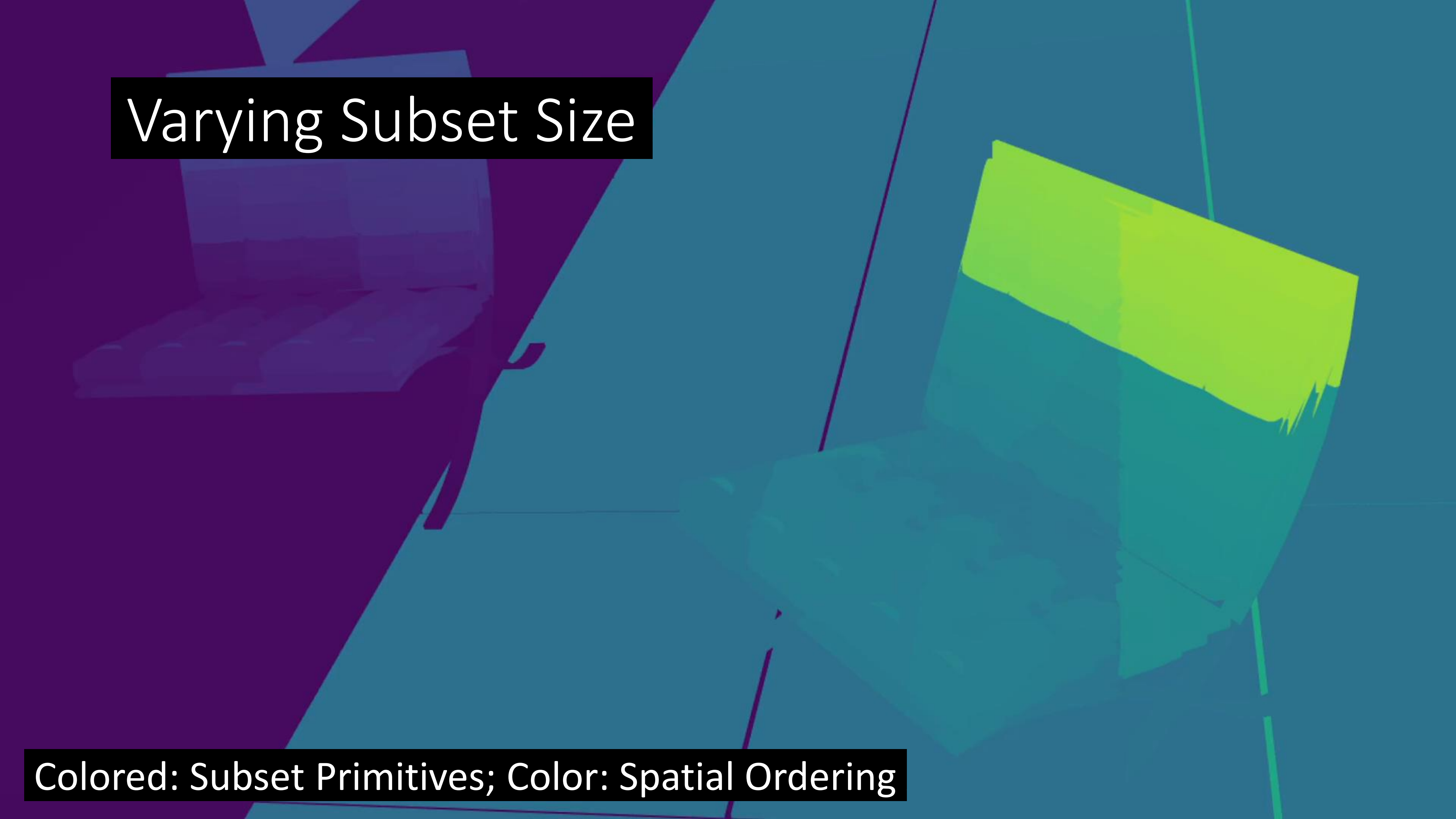
Weight clampling



Defensive sampling



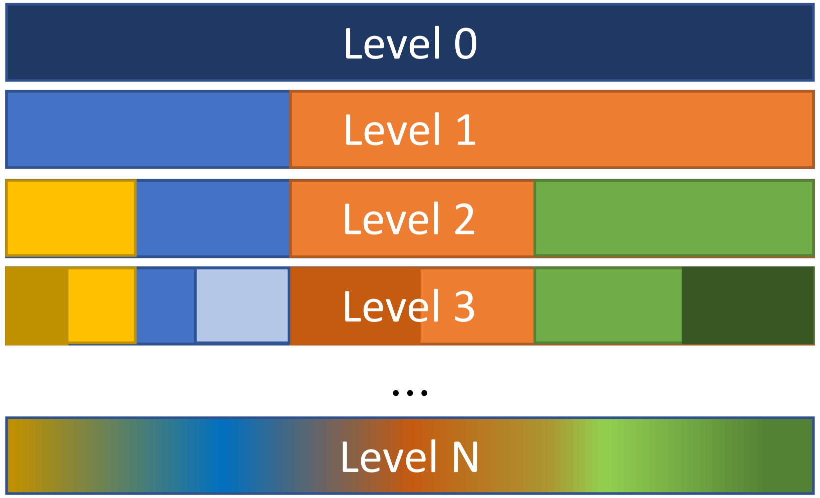
Varying Subset Size



Colored: Subset Primitives; Color: Spatial Ordering

Stochastic Subset BVH Construction

Idea: Operate on subset for first levels to achieve speedup


Level 0

Level 1

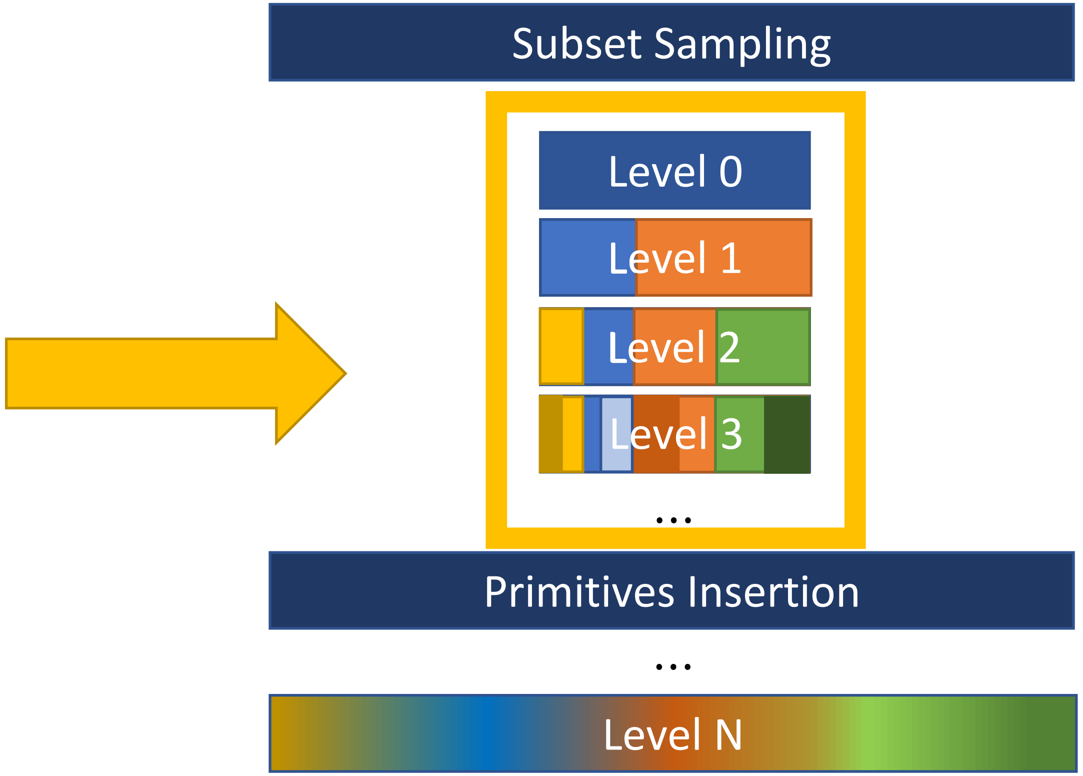
Level 2

Level 3

…

Level N

Top-down construction


Subset Sampling

Level 0

Level 1

Level 2

Level 3

…

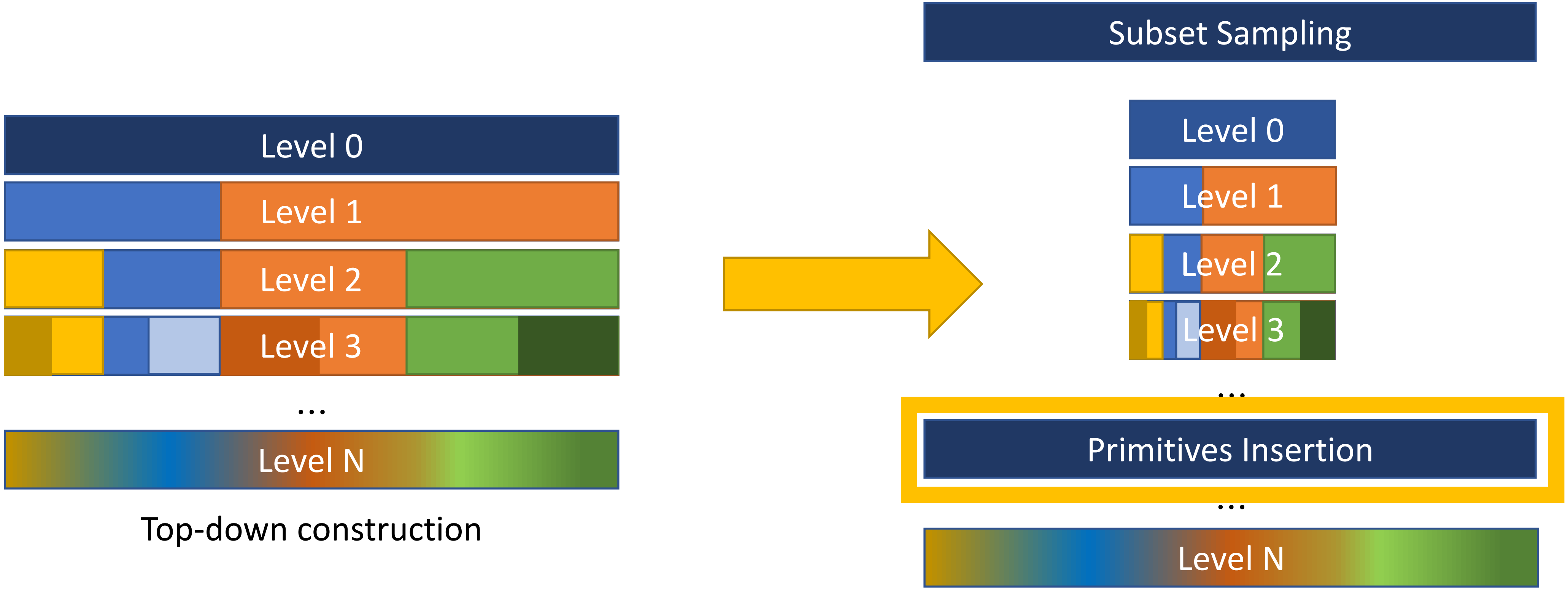
Primitives Insertion

…

Level N

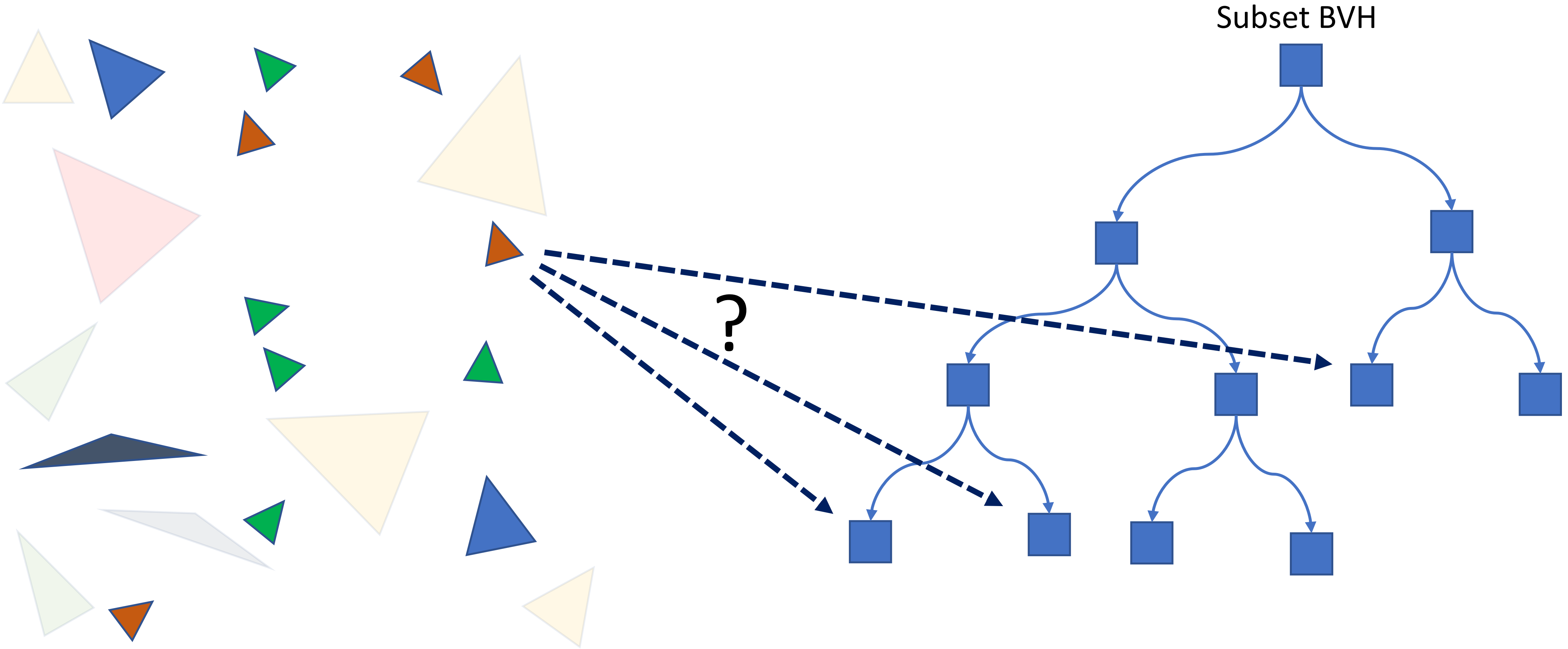
Stochastic Subset BVH Construction

Idea: Operate on subset for first levels to achieve speedup



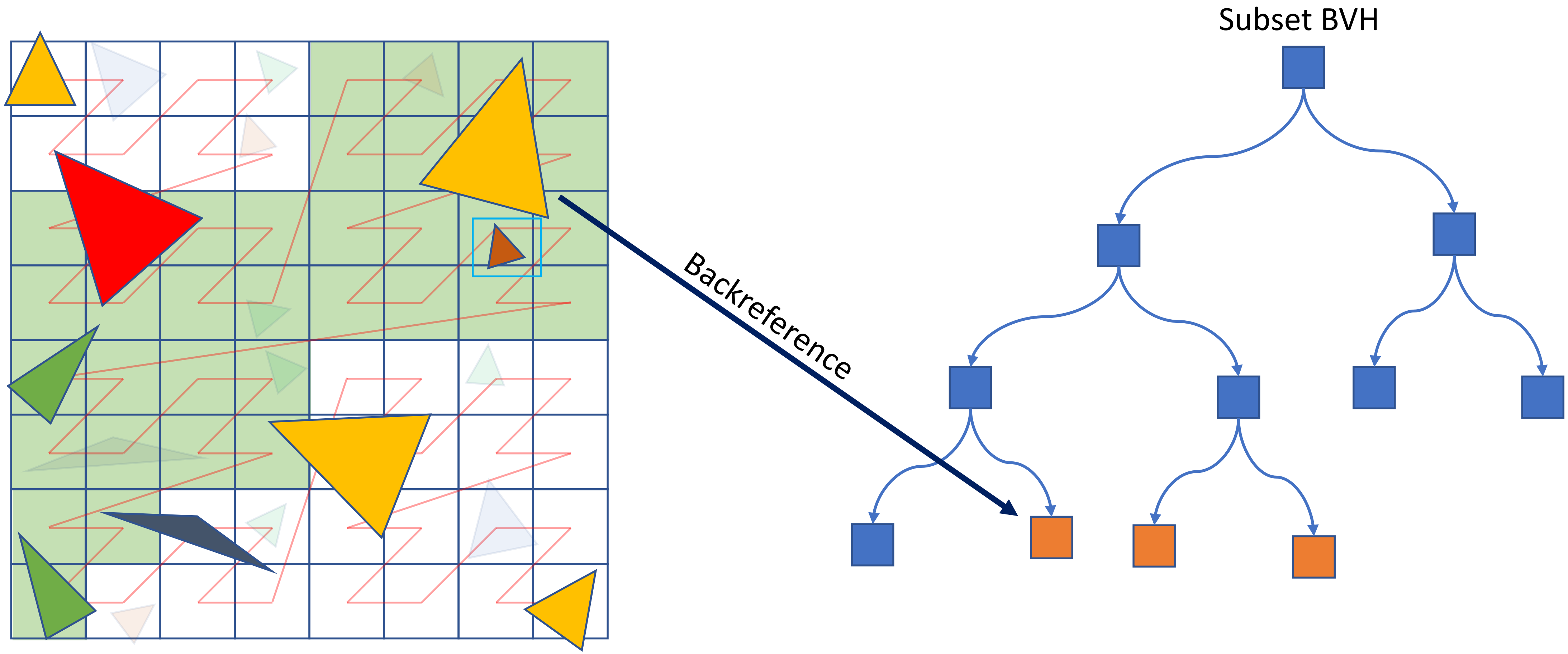
Primitives Insertion

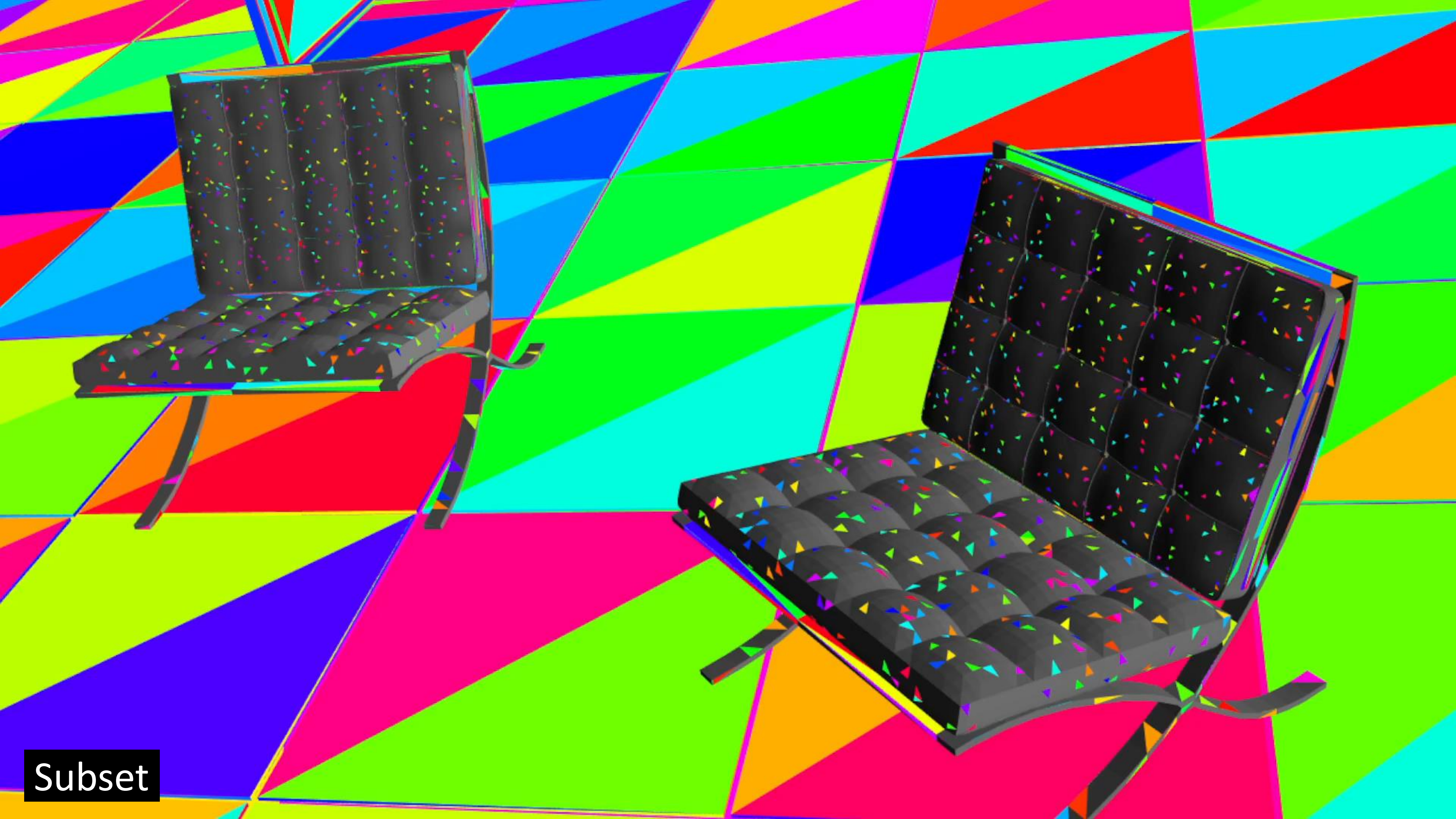
For each primitive, find *best* leaf to insert into



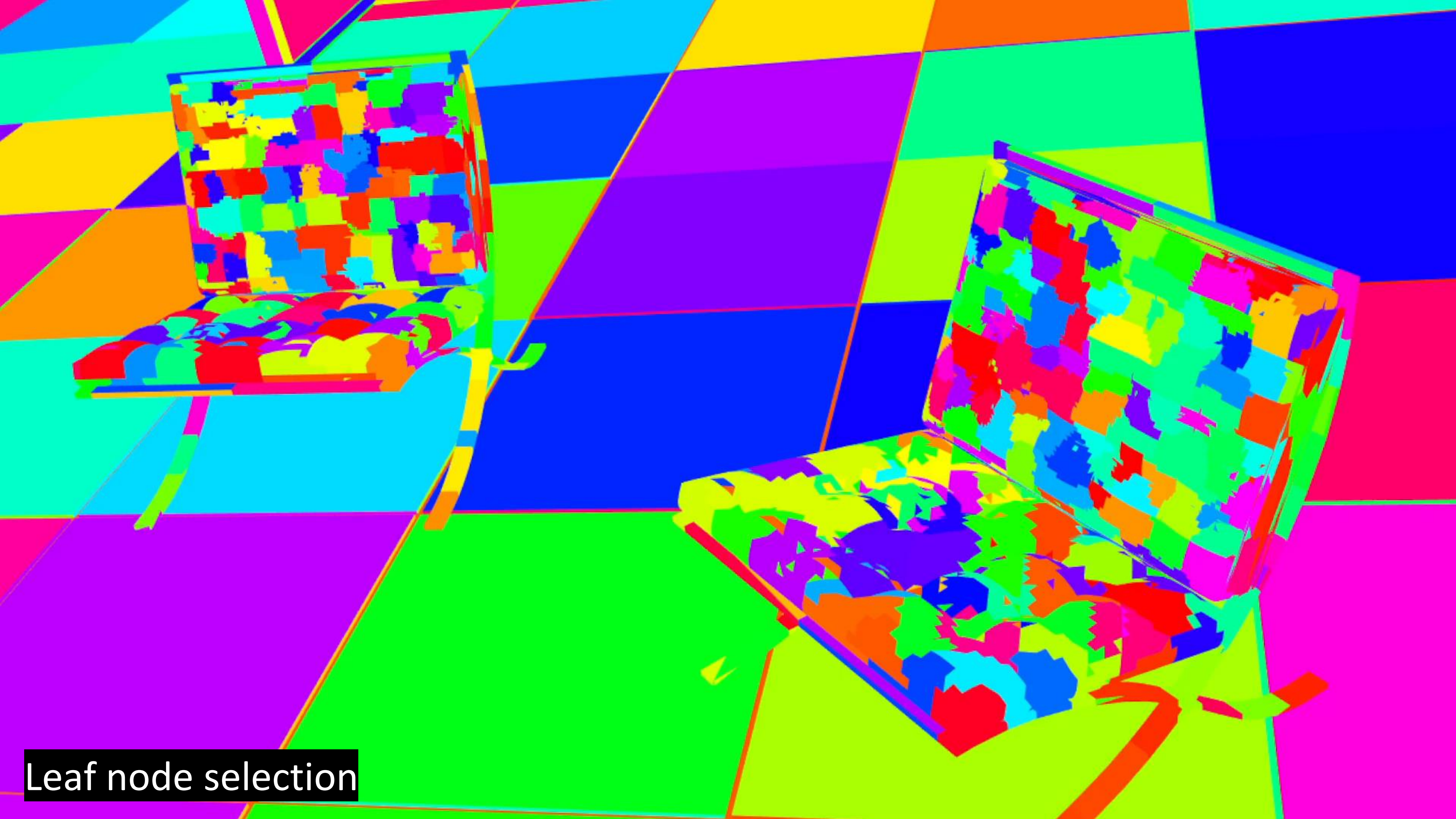
Primitives Insertion

Use approximate neighborhood [Meister et al. 2017]





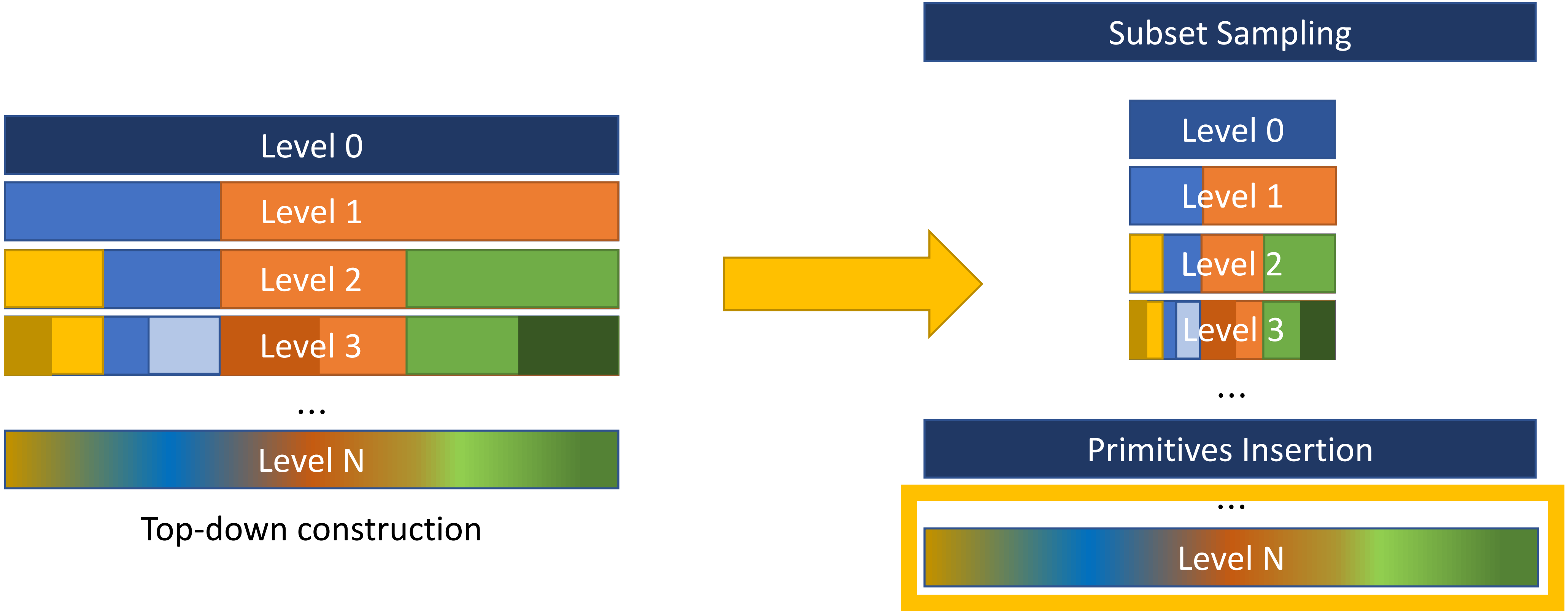
Subset



Leaf node selection

Stochastic Subset BVH Construction

Idea: Operate on subset for first levels to achieve speedup



Top-down construction

Evaluation

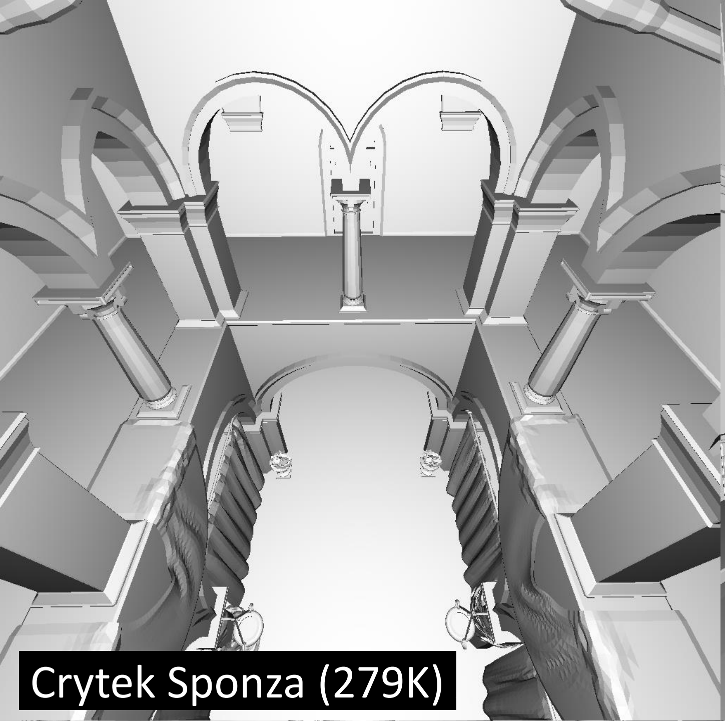
Environment

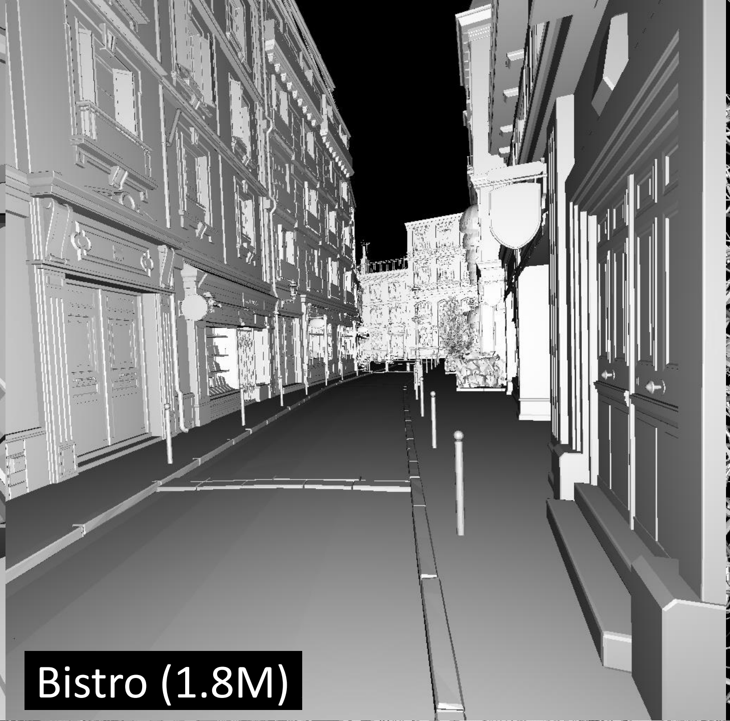
- Intel® Alchemist A770 GPU (32 Xe cores).
- Intel® Core i5 9600K @ 3.7 GHz
- 16GB DDR4 RAM
- Ubuntu 20.04 LTS

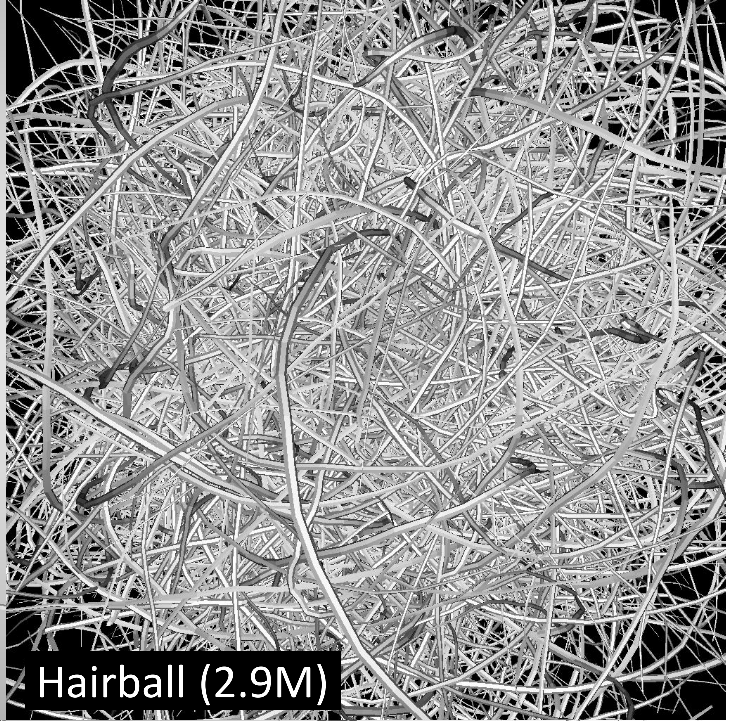
- oneAPI DPC++ / SYCL

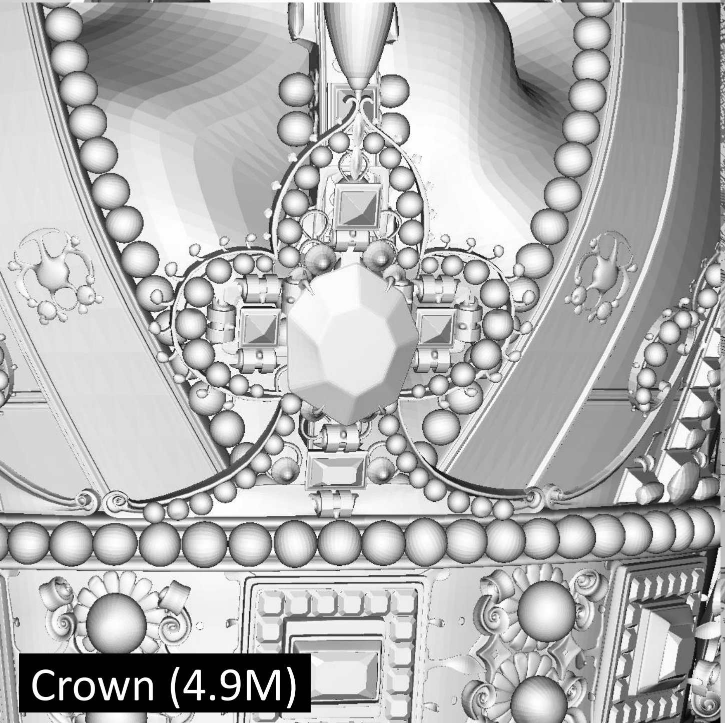
- Subset size of 20%

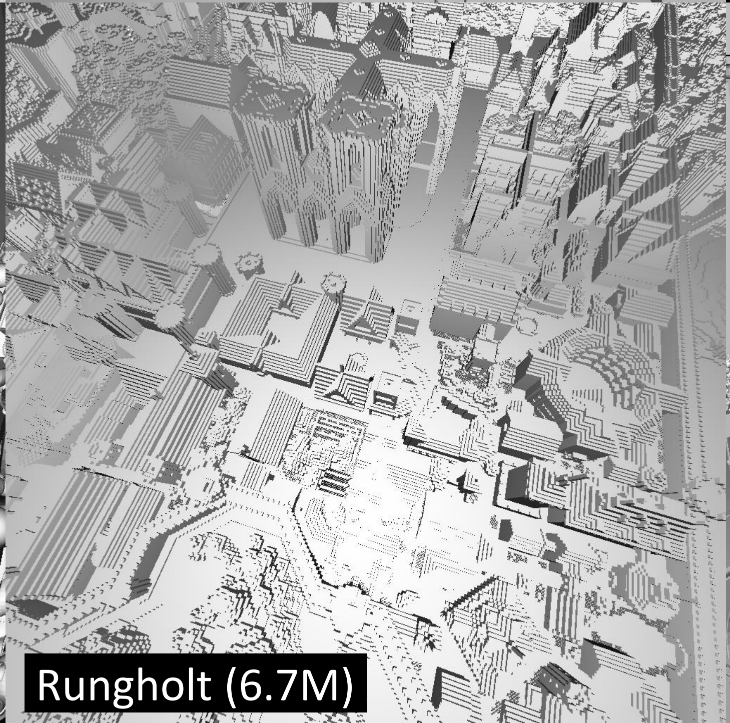
Scenes


Crytek Sponza (279K)


Bistro (1.8M)


Hairball (2.9M)


Crown (4.9M)

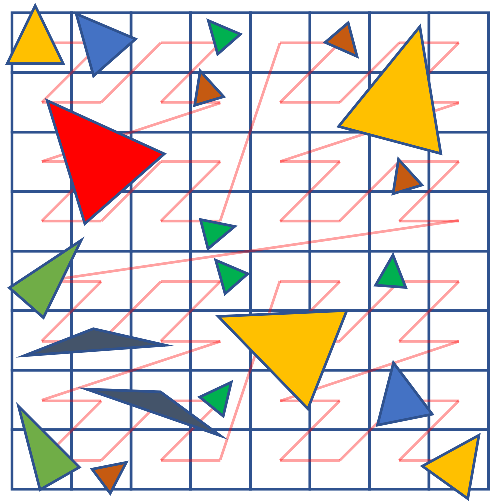

Rungholt (6.7M)

San Miguel (7.9M)

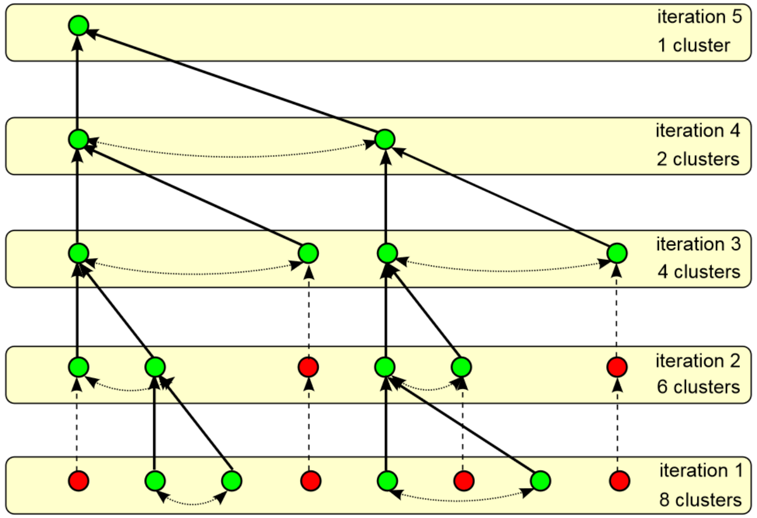
Compared Builders

- GPU implementation of top-down binned SAH builder [Wald 2007]
  - Also used as interior builder
- Other state of the art builders:

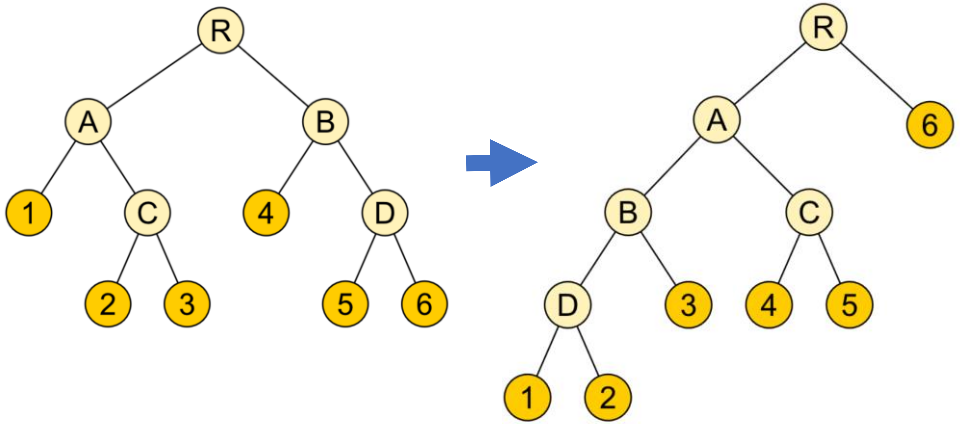
LBVH



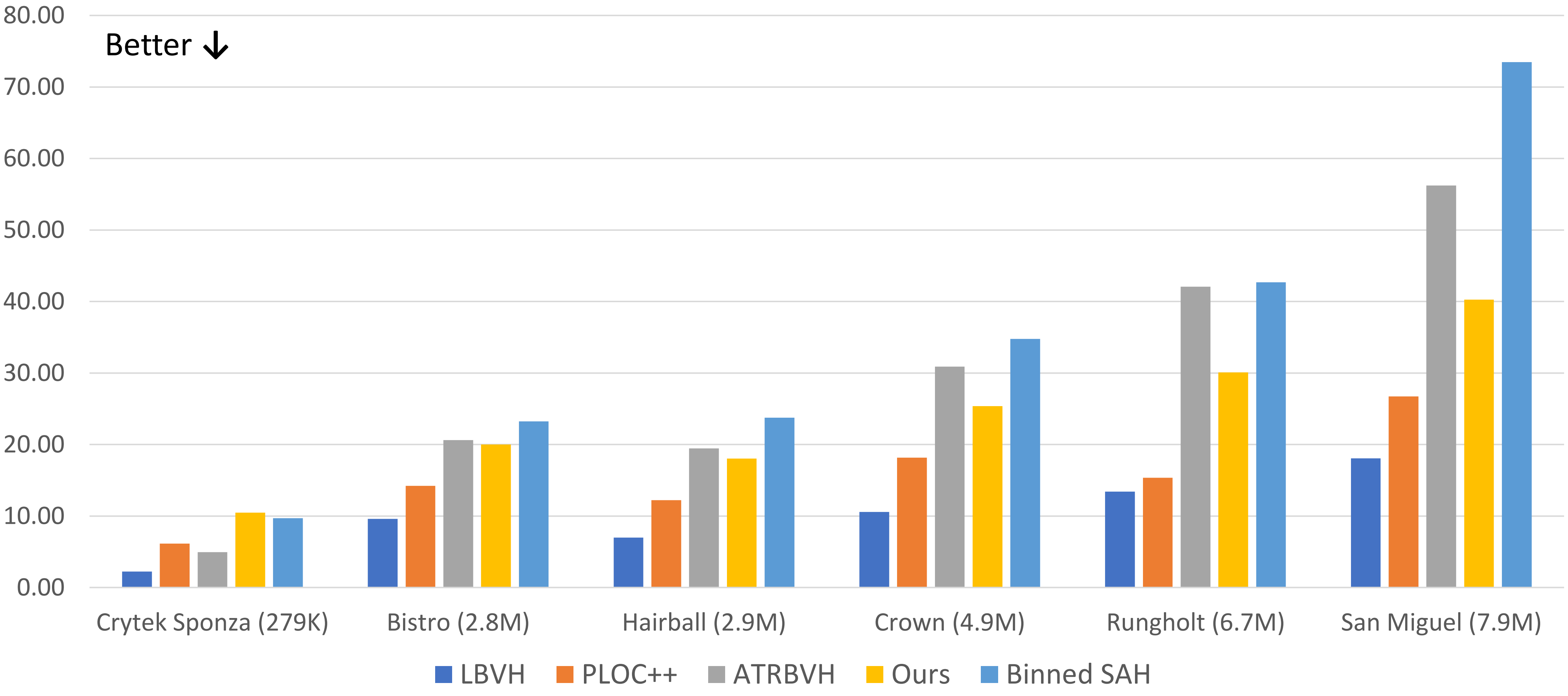
PLOC / PLOC++



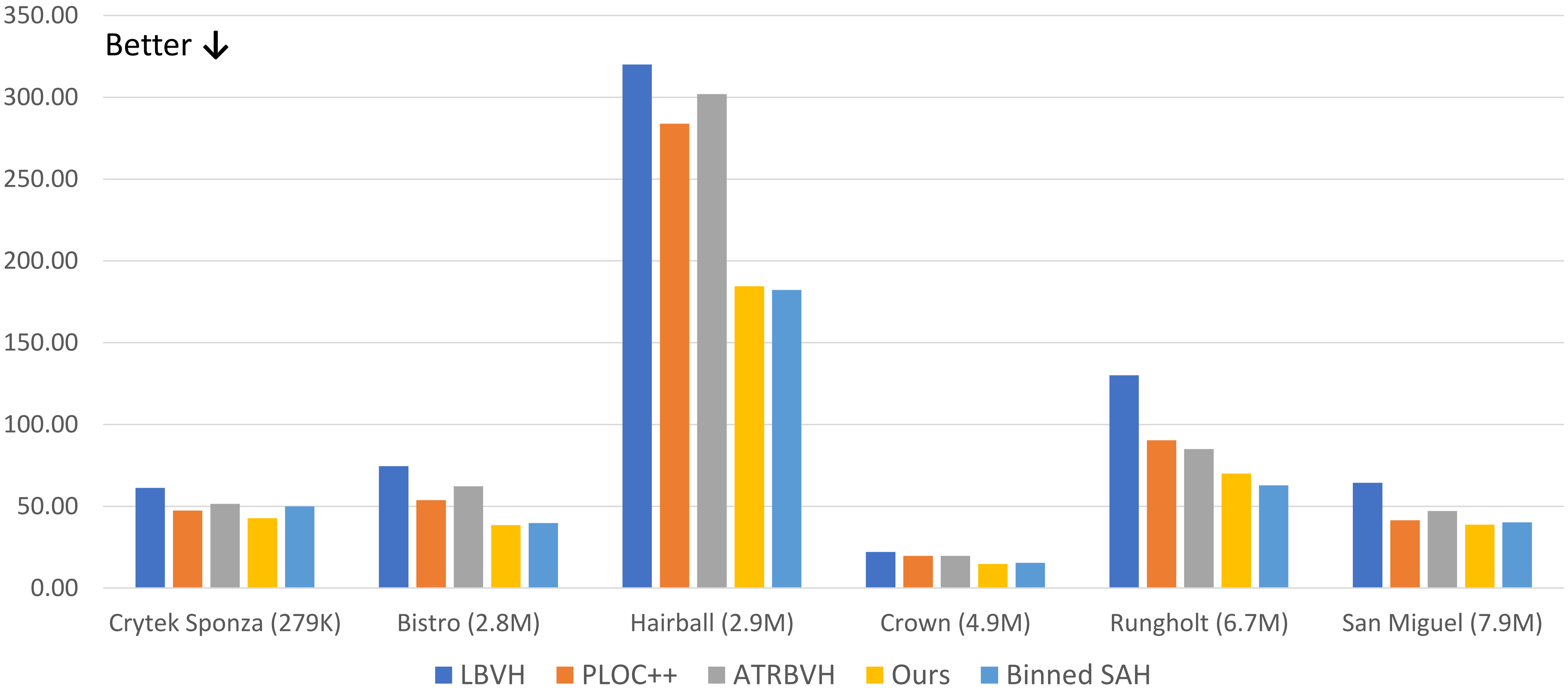
TRBVH / ATRBVH



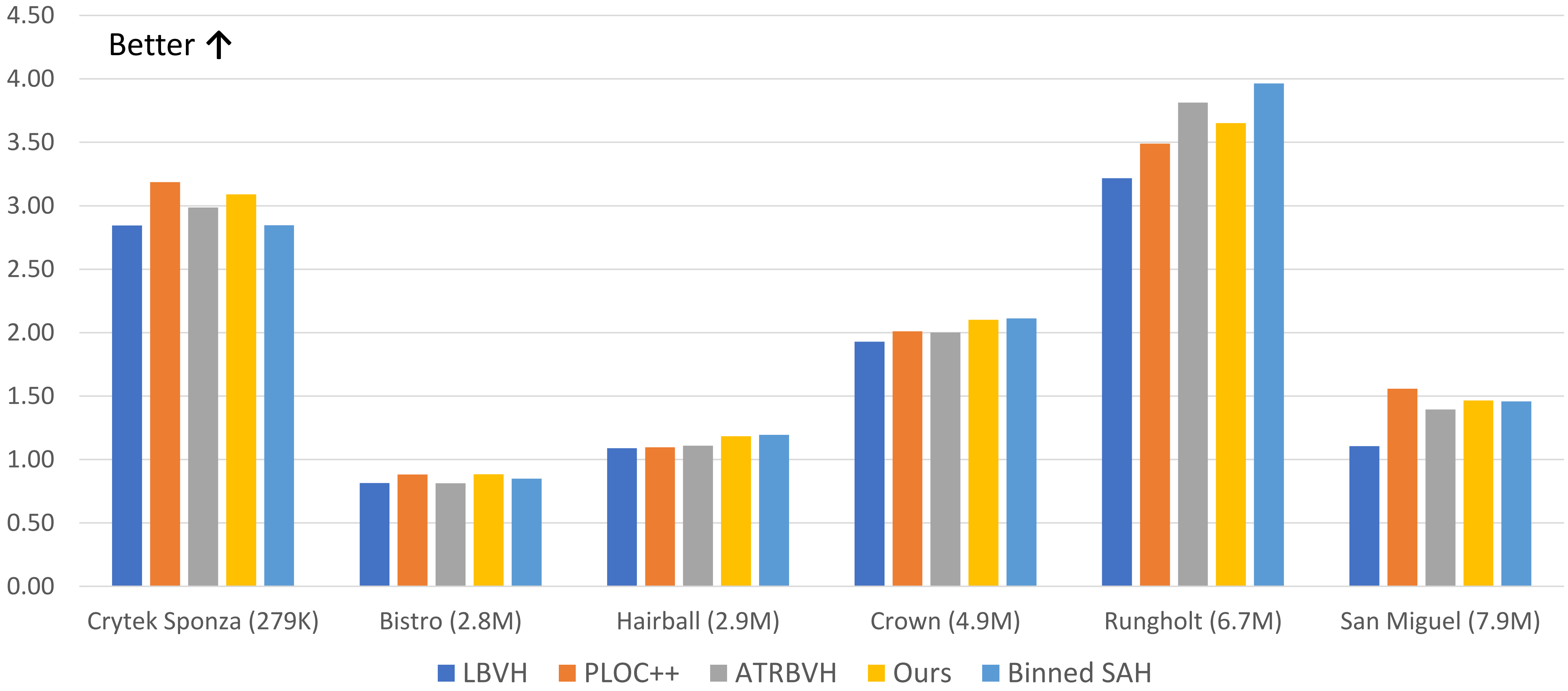
Binary BVH Host Build Time (ms)



Quality Metric: Binary BVH SAH Cost



Hardware Ray Throughput with AO (GRays/s)

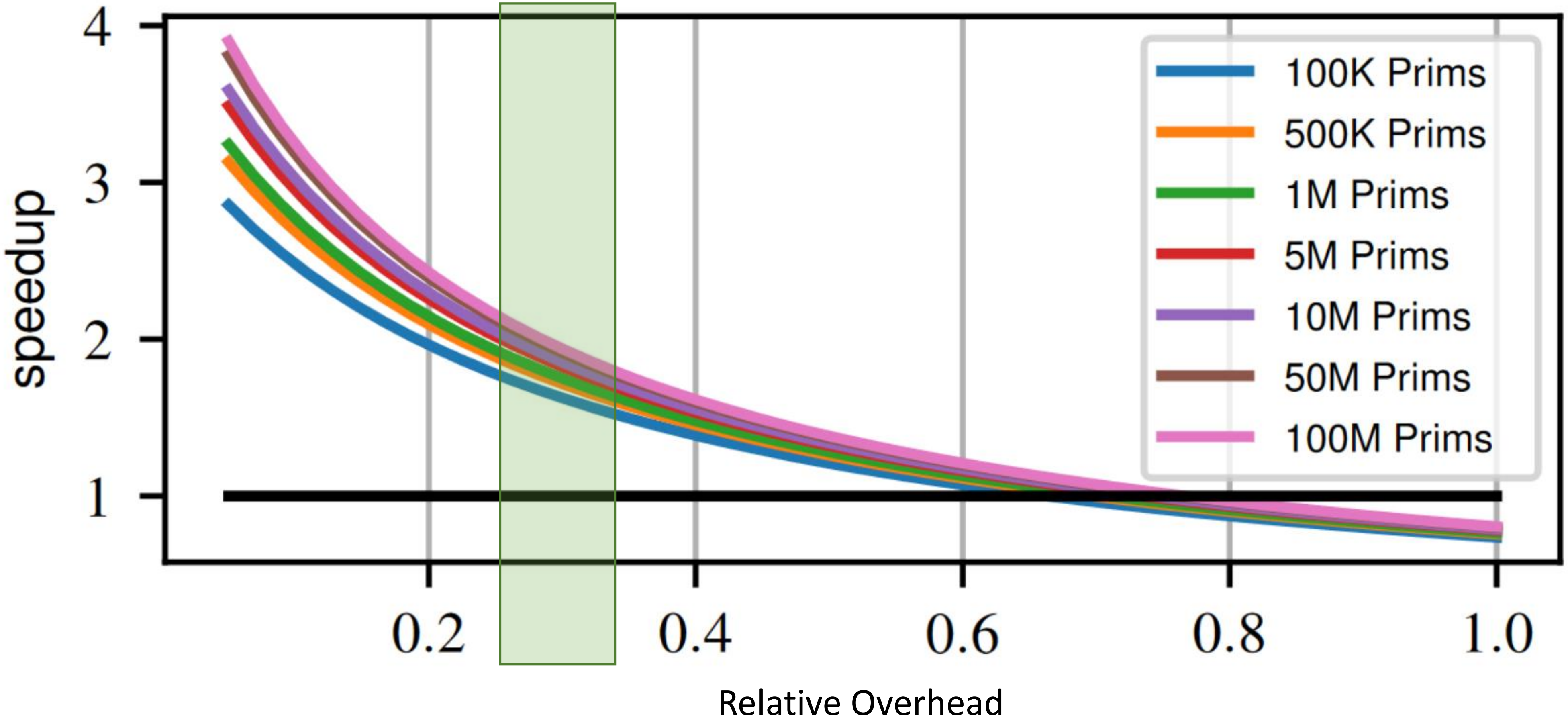


⚠ Additional conversion to hardware-specific BVH format

Discussion

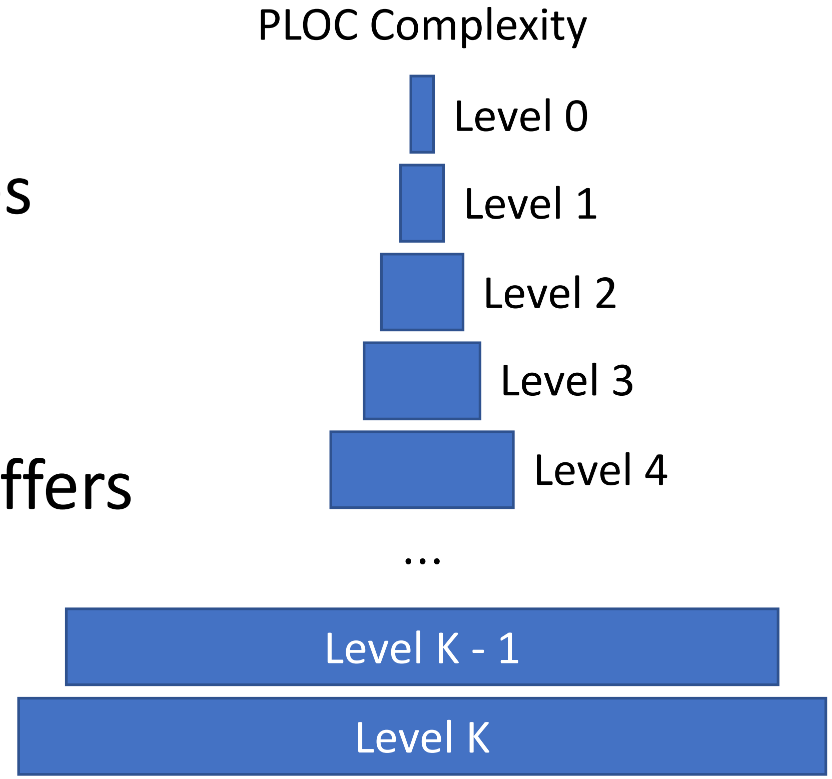
Discussion

- The relative overhead impacts the final speedup



Discussion

- Currently benefits $O(N \log N)$ algorithms
- Not directly applicable to $O(N)$ approaches in its current form
- Higher memory footprint for additional buffers



Conclusion

- New primitive in BVH construction
  - Now possible to importance sample multiple features of the scene
- Retains quality of top-down builder at 1.5x avg. (~1.9x max.) speedup
- Reducing the gap to faster builders (e. g. PLOC++)
  - Untapped theoretical speedup still on the table

https://cg.ivd.kit.edu/stochbvh.php

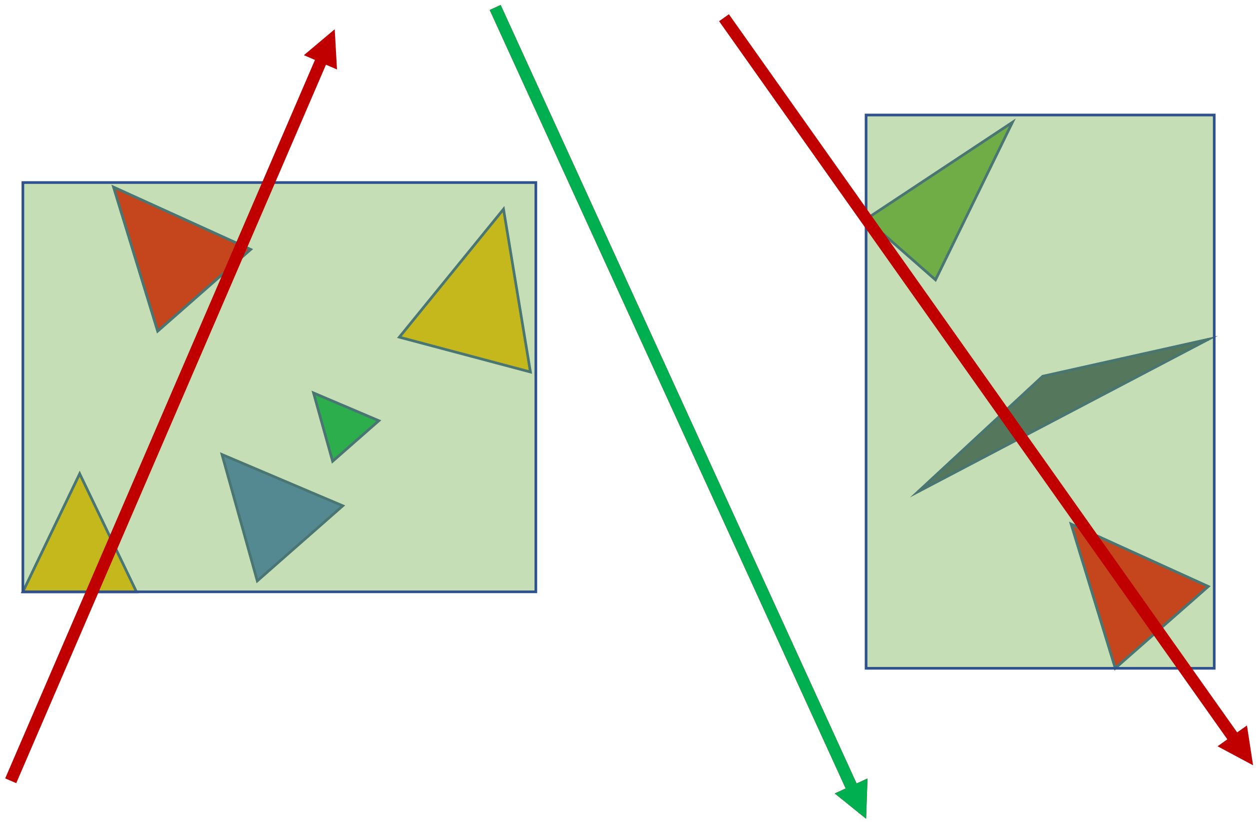
Thanks for your attention! Questions?

Contact: lorenzo.tessari@intel.com, addis.dittebrandt@kit.edu

Backup

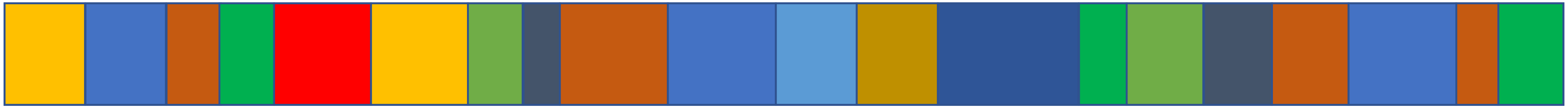
Introduction - Surface Area Heuristic (SAH)

- Quantifies intersection cost through bounding box areas

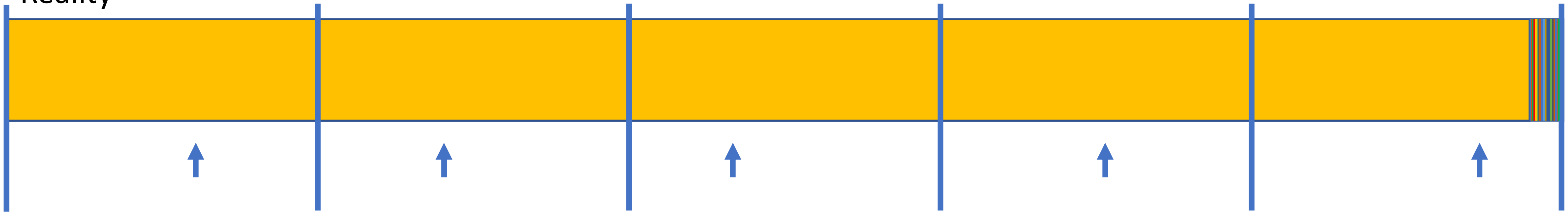


Subset Sampling - Weight Clamping

Ideal



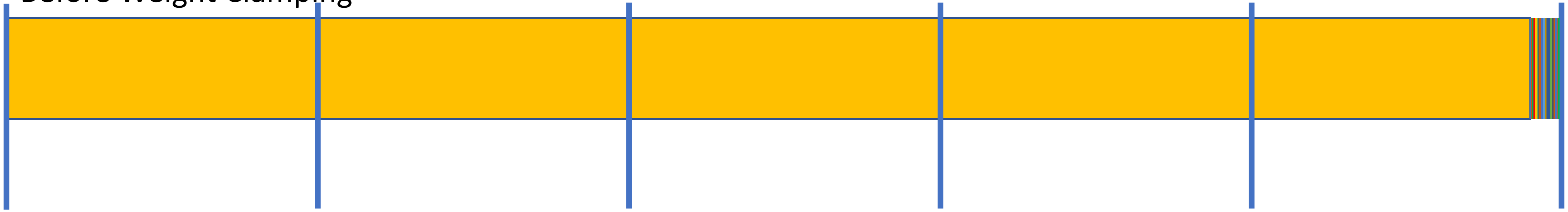
Reality



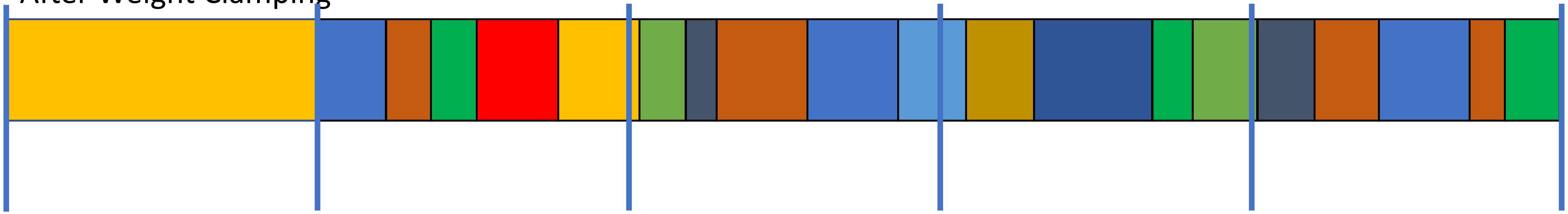
→ Duplicates

Subset Sampling - Weight Clamping

Before Weight Clamping



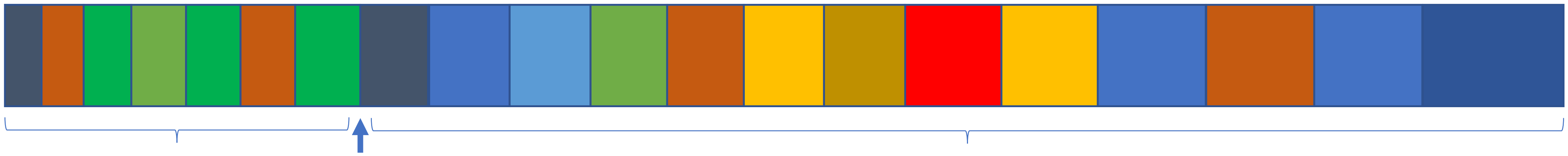
After Weight Clamping



Subset Sampling - Weight Clamping

$$\mathbf{clamp} \geqq \text{stratum} \cdot \sum_{j}^{N} \min\left(\text{weight}_j, \mathbf{clamp}\right)$$

$\mathbf{clamp} = \text{weight}_k$

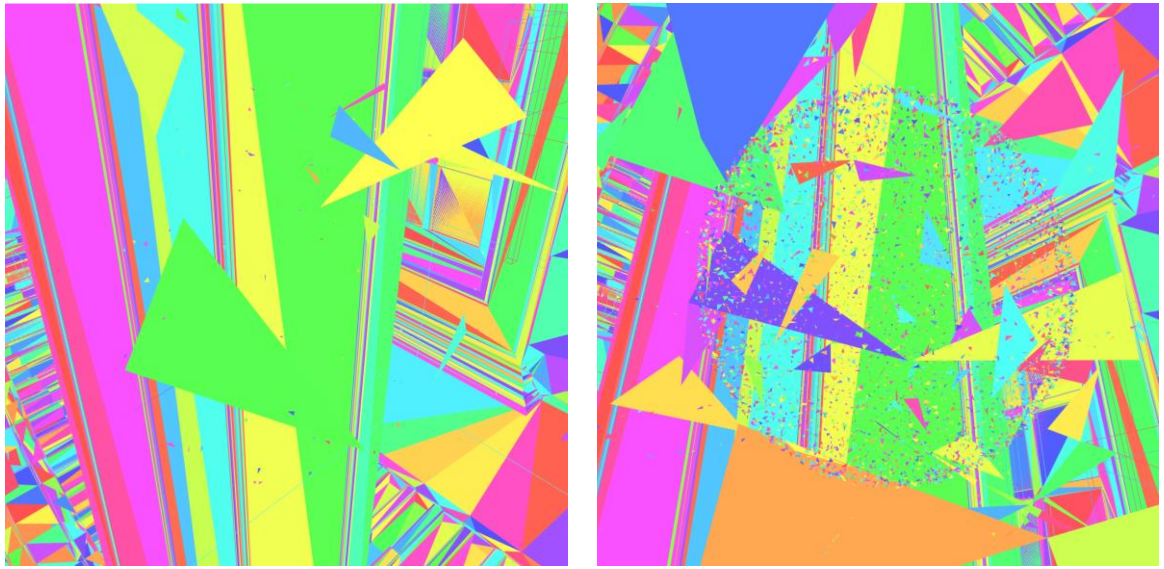


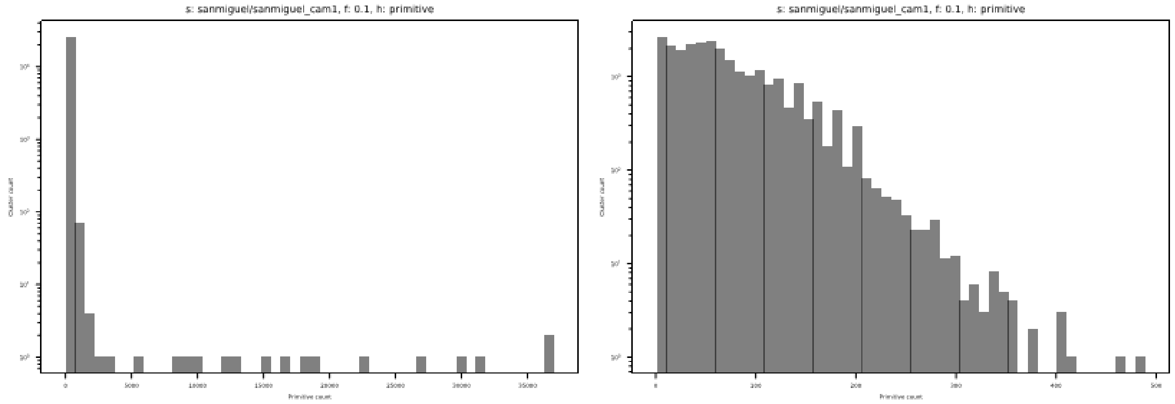
k

$$\sum_{j=0}^{k} \text{weight}_j \qquad + \qquad (\text{N} - \mathbf{k}) \cdot \mathbf{clamp}$$

Find smallest **k** which satisfies $\text{weight}_k \leq \text{stratum} \cdot \sum_j^N \min\left(\text{weight}_j, \text{weight}_k\right)$

Subset Sampling - Defensive Sampling





Cluster size histogram: left max > 35k primitives, right max 500

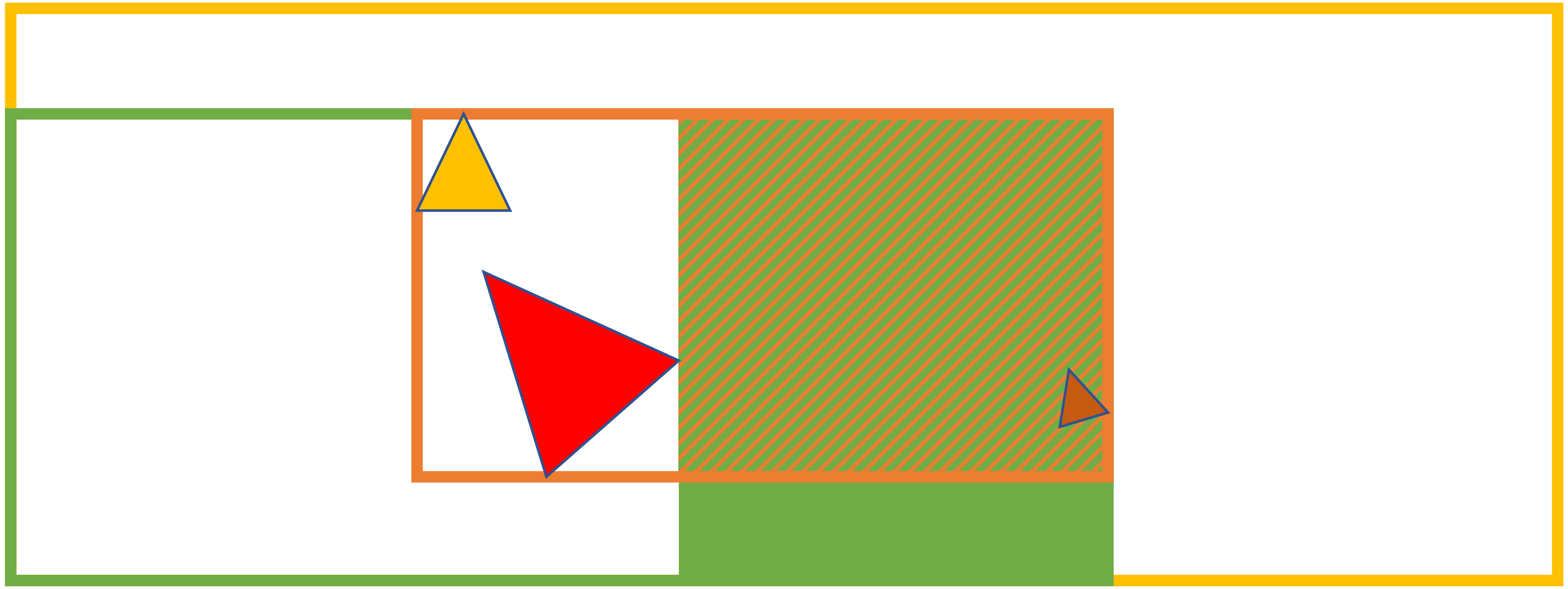
Small, highly tessellated flower in San Miguel.

Forcing some more uniform selection levels the cluster size after insertion.

Primitives Insertion - Metric

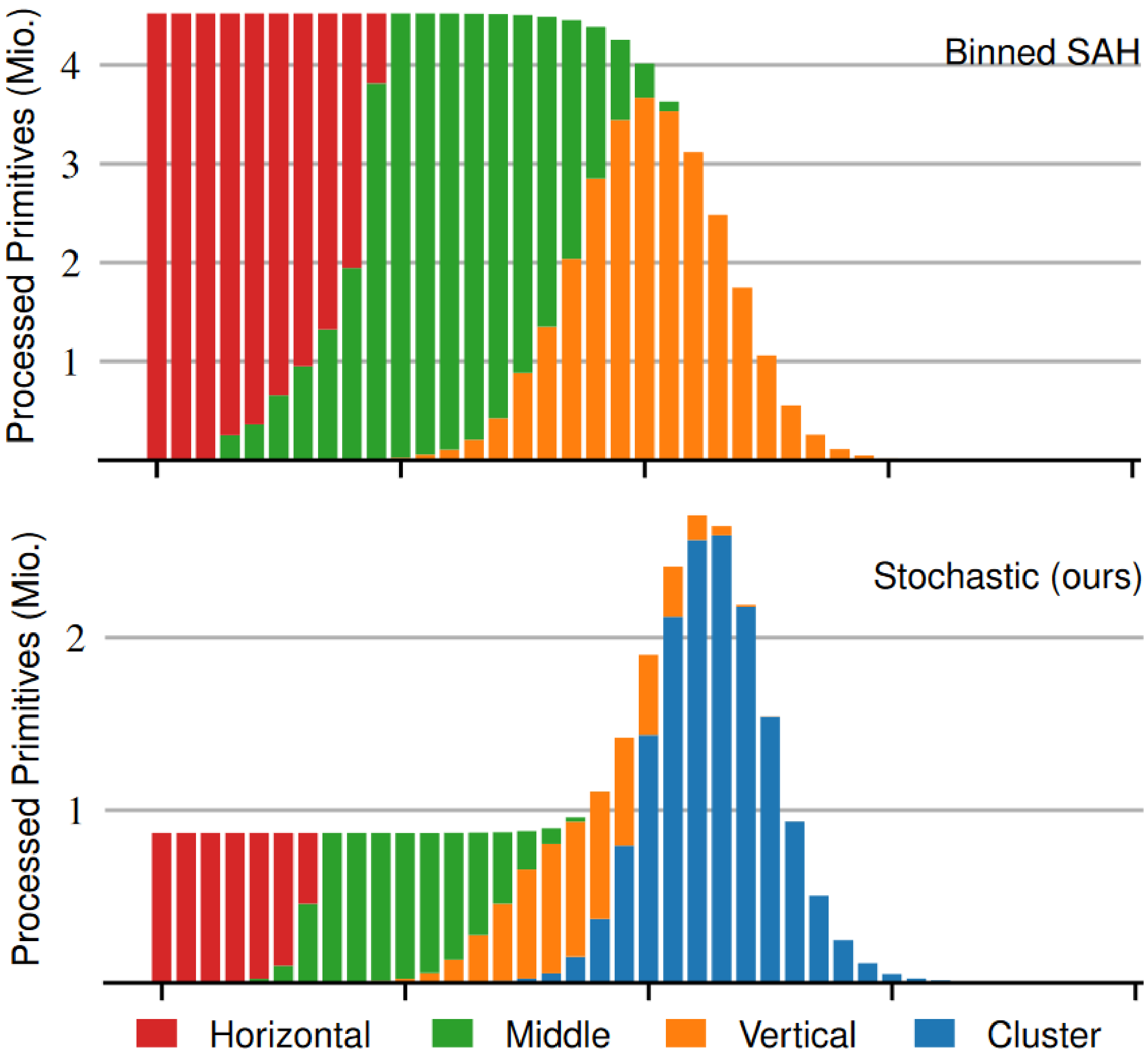
Metric: Increase of SAH [Bittner et al. 2013]

- Only need to evaluate leaf and ancestor nodes
  - Search can be parallelized in groups



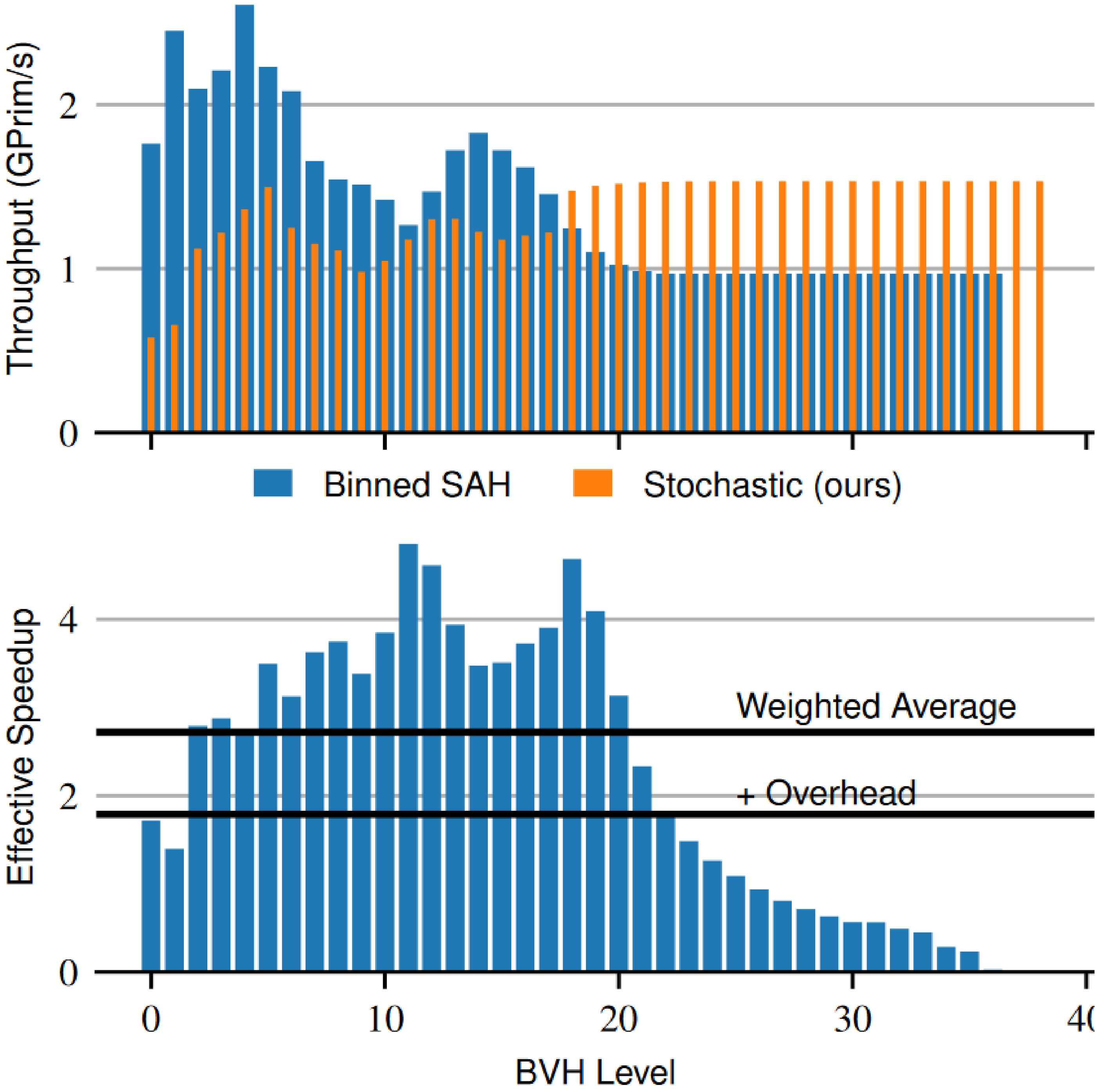
Throughput

- much fewer primitives in the first levels …



Throughput

- much fewer primitives in the first levels ...
- ... but also less throughput
- there is still a net speedup, but less than anticipated


Throughput (GPrim/s)
Binned SAH
Stochastic (ours)
Effective Speedup
Weighted Average
+ Overhead
4(
BVH Level