Selective Restructuring of Bounding Volume Hierarchies for Dynamic Models

Sung-Eui Yoon

KAIST (Korea Advanced Institute of Science and Technology)

At Previous Class

- **Studied multi-resolutions, culling, cache-coherent layout techniques**

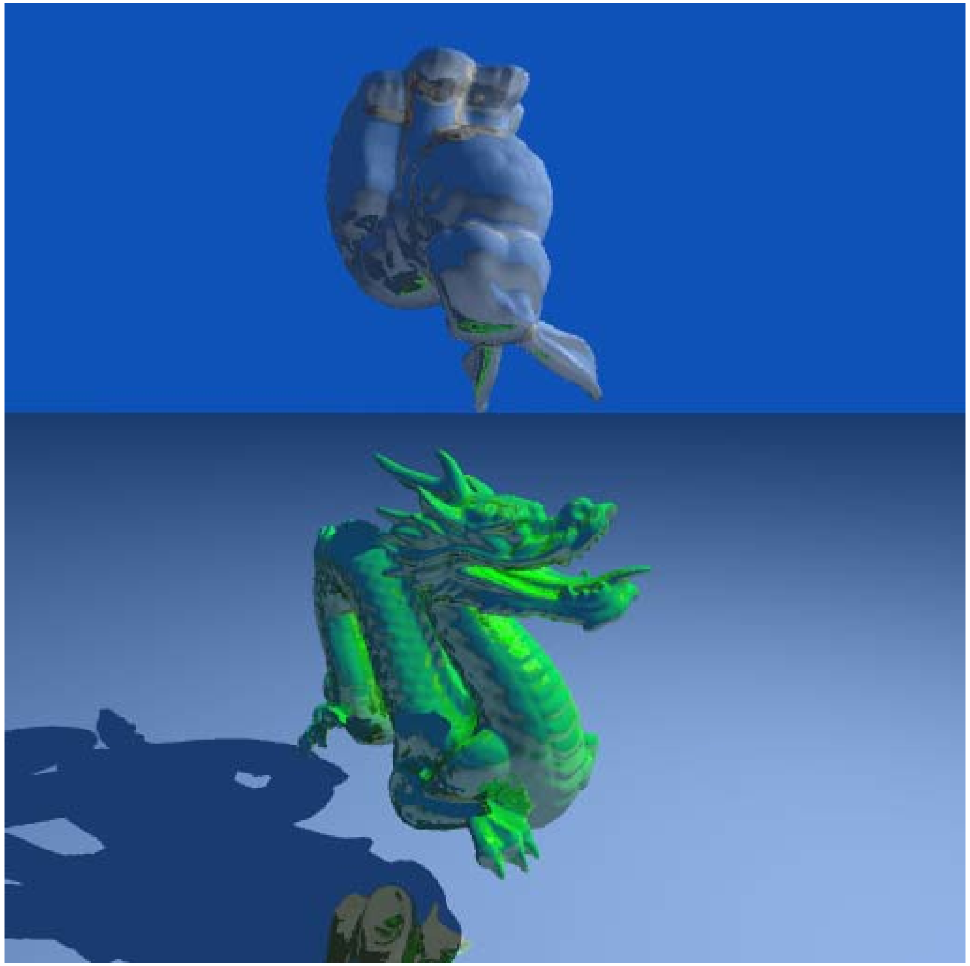
What is one of major problems of these techniques?

KAIST

Motivations

- **Dynamic scenes are widely used**
  - **Movies, VR applications, and games**
- **Complex and large dynamic scenes**
  - **E.g, high-resolution explosion, tears, and fractures**

An Example of Exploding Dragon (252K triangles)



Ray Tracing Dynamic Scenes

- **Acceleration hierarchy construction**
  - **e.g., kd-trees, bounding volume hierarchies, grids, etc**
- **Hierarchy traversal**
  - **Perform ray-triangle intersection tests**
- **Key issue**
  - **Update the hierarchy as triangles deform**

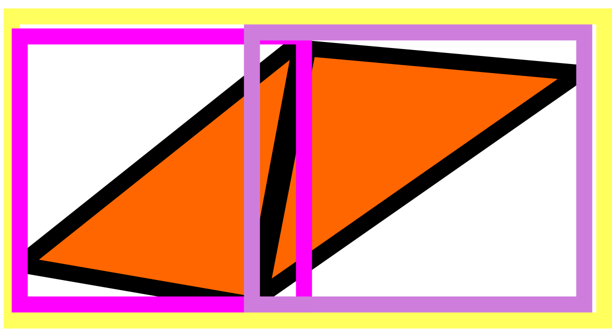
Bounding Volume Hierarchies (BVH) based Ray Tracing

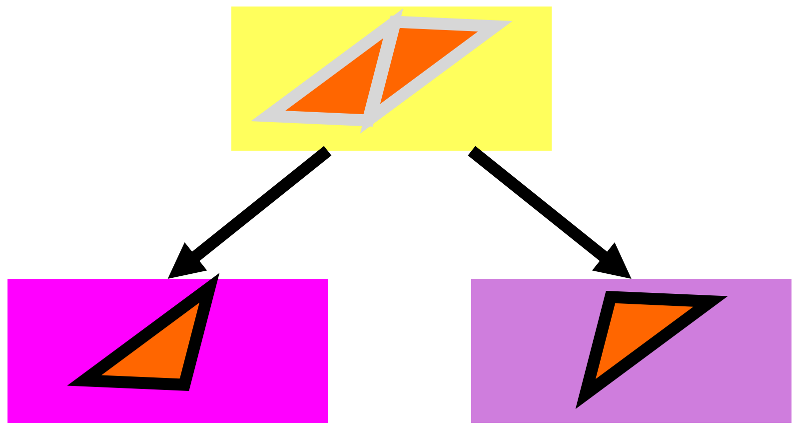
- Employed early in [Whitted 80]
  - kd-trees and grids became popular for static models in 90's
- Recently get renewed interest in ray tracing dynamic scenes [Wald et al. 07, Lauterbach et al. 07, Larsson et al. 03]
  - Simple, but efficient BVH update method is available
  - Can have better performance

KAIST

BVHs

- **Object partitioning hierarchies**
  - **Uses axis-aligned bounding boxes**
  - **Considers surface-area heuristic (SAH) [Goldsmith and Salmon 87]**

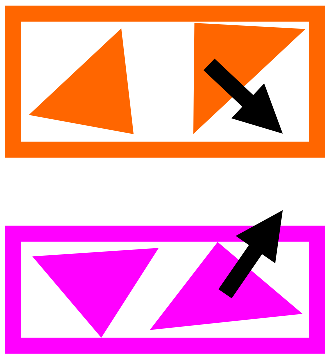




A BVH

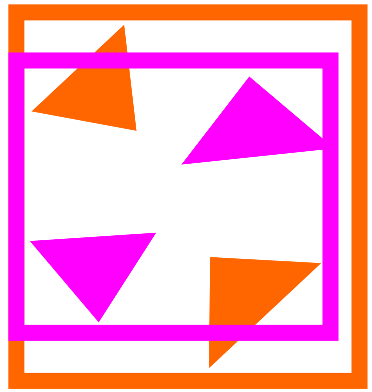
Two BVH Update Methods

Frame 1



BV refitting

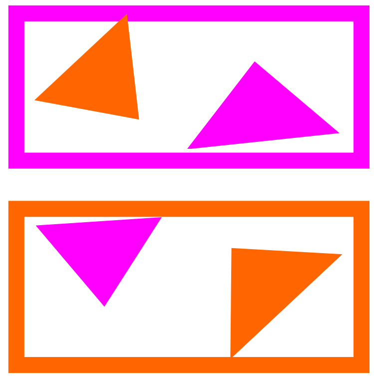
- O(n)
- Poor-quality BVs



Frame 2

BV reconstruction

- O(n log n)
- Good-quality BVs



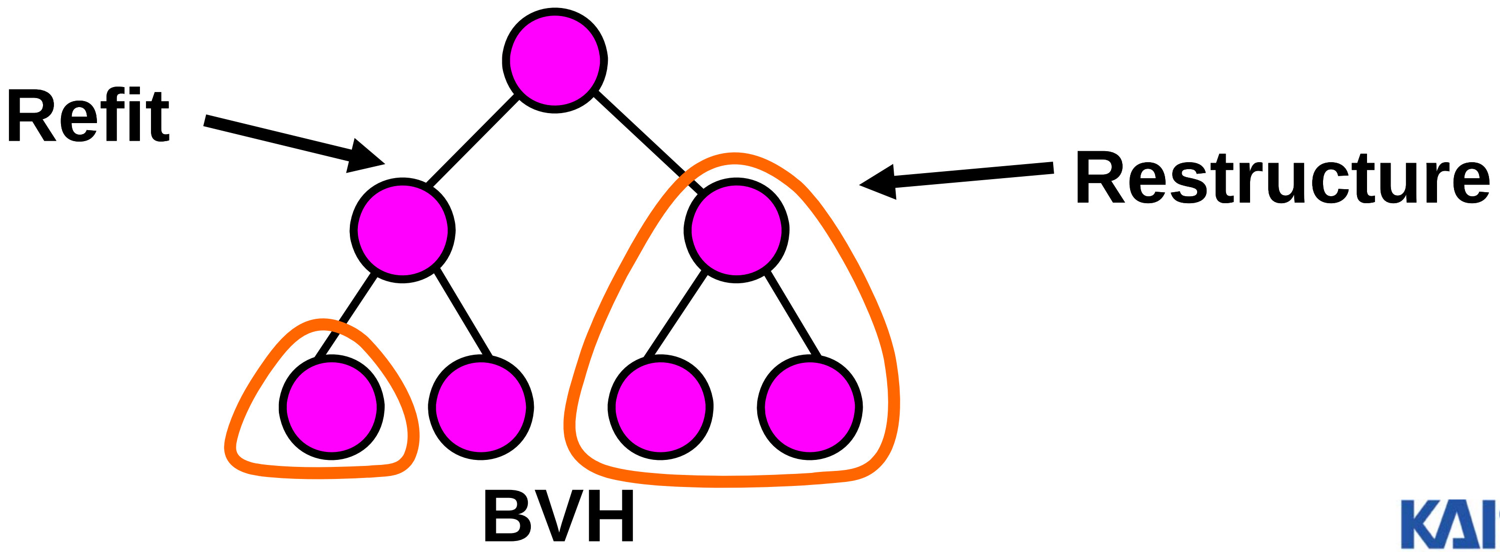
Our Goal

- Existing BVH update methods
  - Work at particular classes of dynamic scenes
- Design a robust BVH update method
  - Works well with wide classes of dynamic scenes
  - Improves the performance of ray tracing

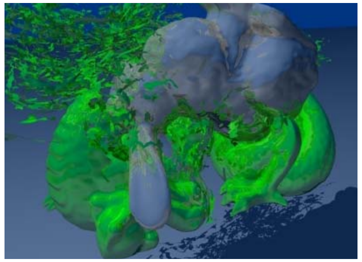
KAIST

Our Contributions

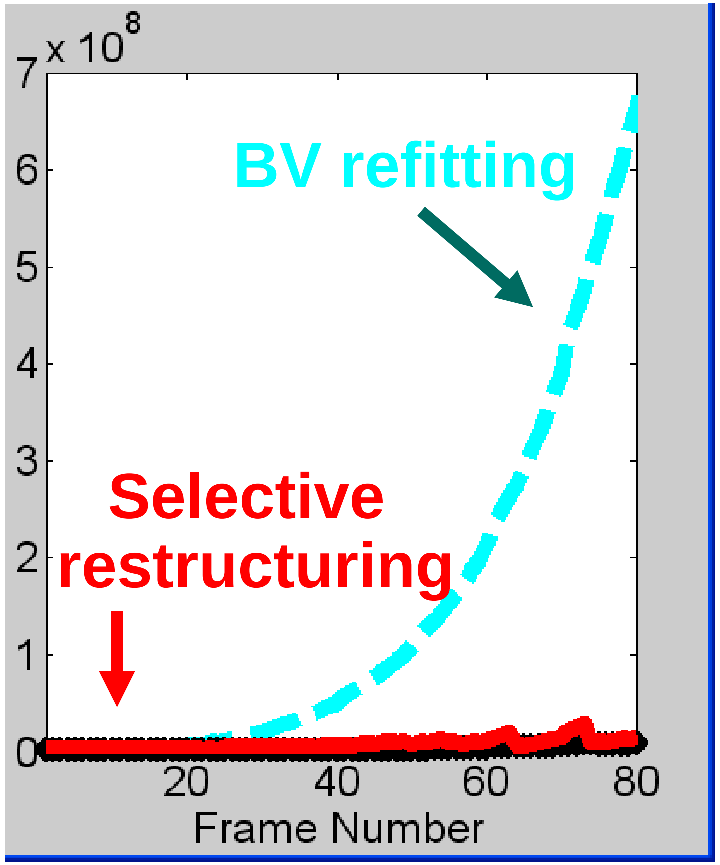
- Proposes a novel algorithm to selectively restructure BVHs [Yoon et al., EGSR 07]
  - Selective restructuring operations
  - Two probabilistic metrics: culling efficiency and restructuring benefit



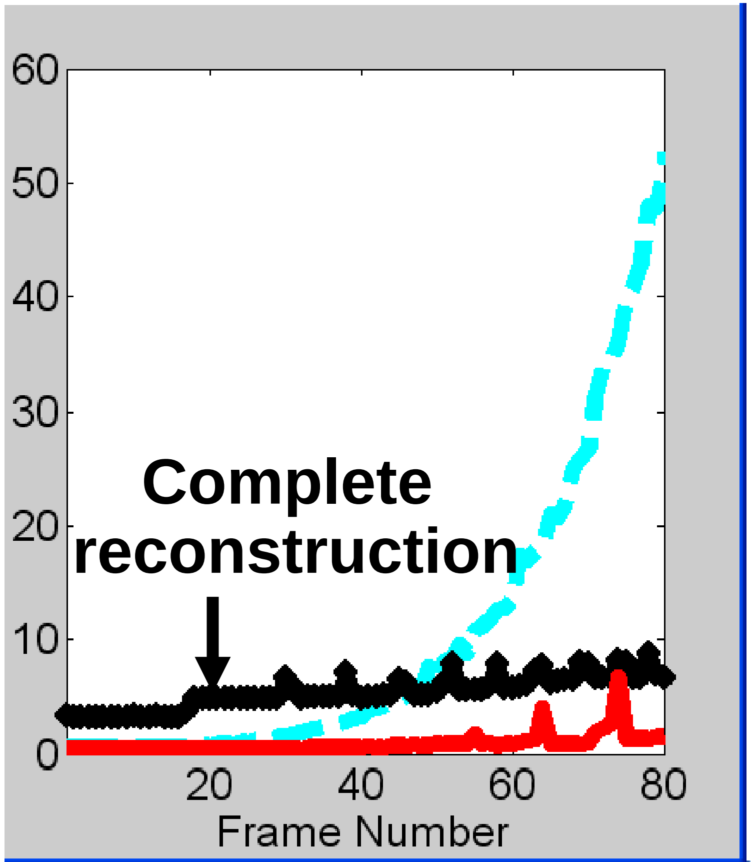
Example of Exploding Dragon Model



# of intersections

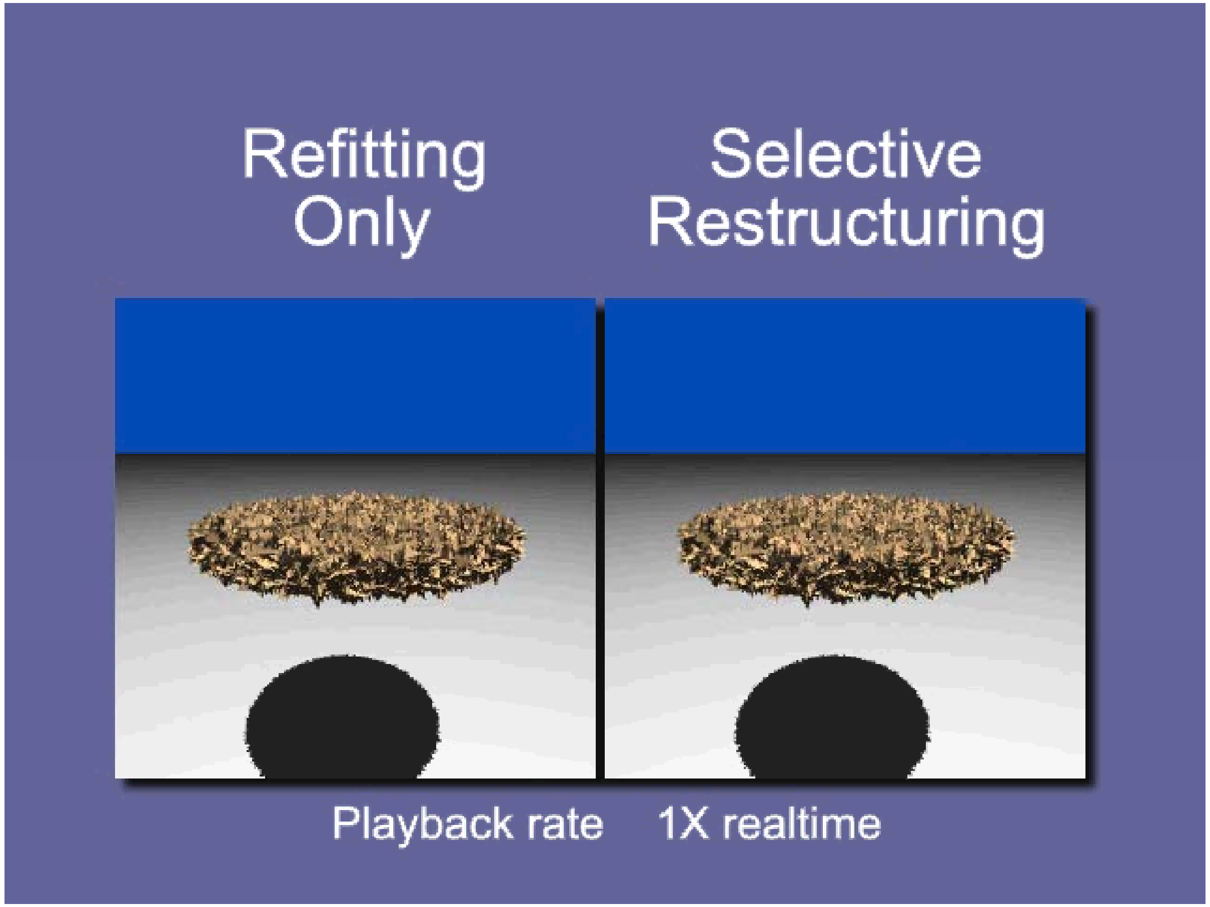


Ray tracing time (sec): construction + traversal



Runtime Captured Video – BART Model (65K triangles)

- **Compared with the BV refitting method**



Enabled primary & shadow rays

Single thread

Probabilistic BVH Metrics for Ray Tracing

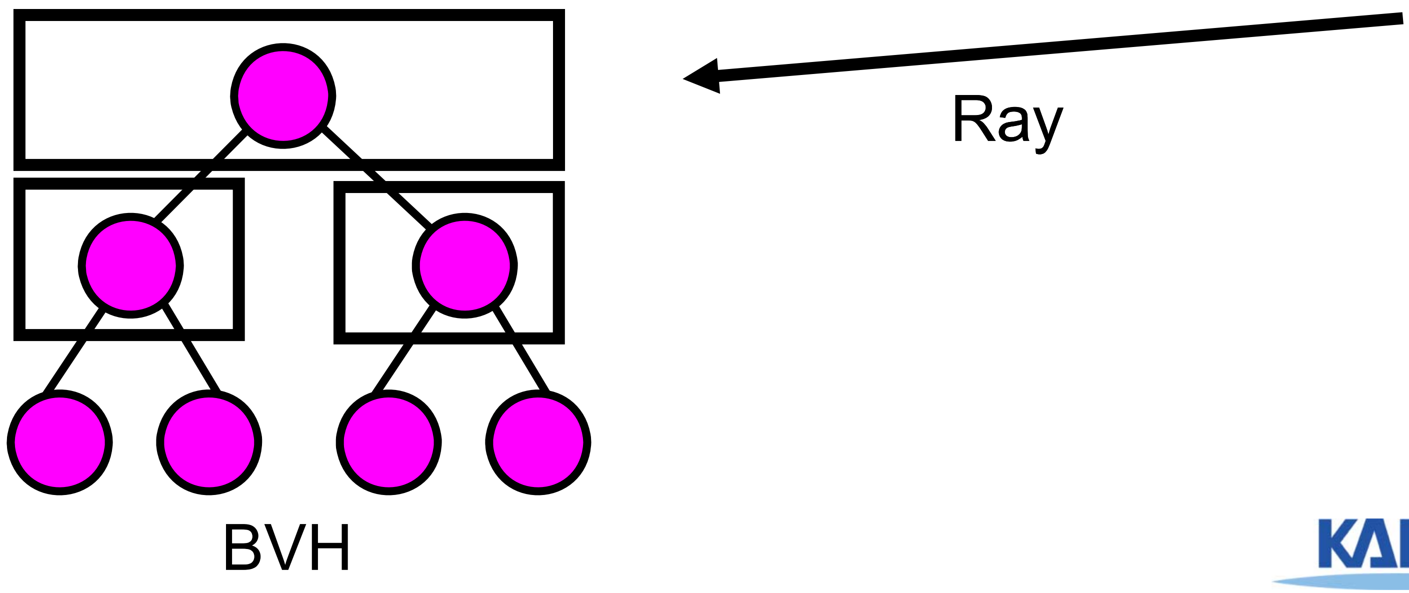
- **Culling efficiency**
  - **Quantifies the quality of any sub-BVHs**
  - **Measures the expected # of intersection tests for a ray**

- **Restructuring benefit**
  - **Predicts the performance improvement**
  - **Measures improved culling efficiency when restructuring sub-BVHs**

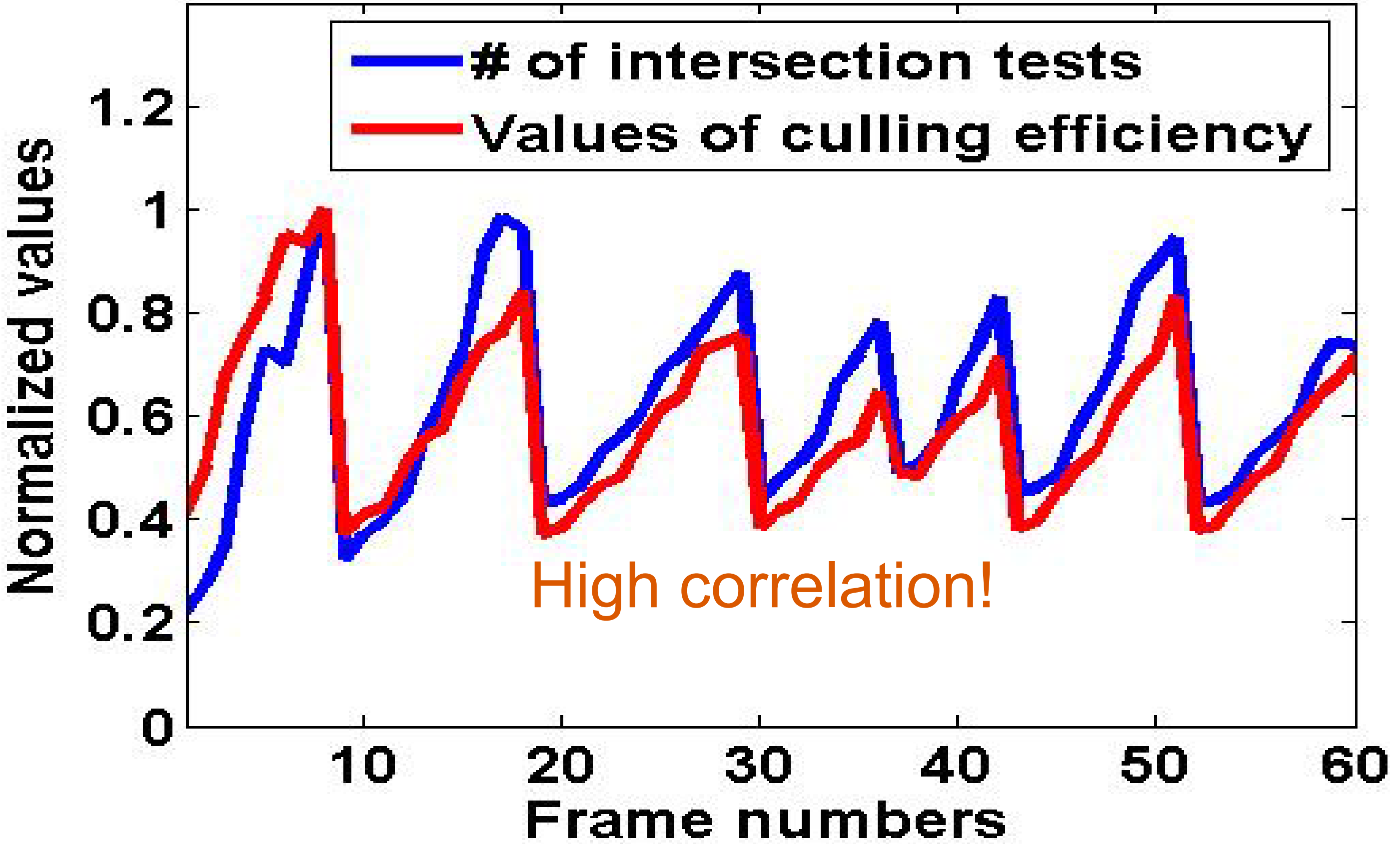
KAIST

Culling Efficiency Metric

- **Measure the expected # of intersection tests for a ray**
  - **Measured in a view-independent manner**
  - **Recursively computed with child nodes considering SAH [Goldsmith and Salmon 87]**



Validation of Culling Efficiency Metric



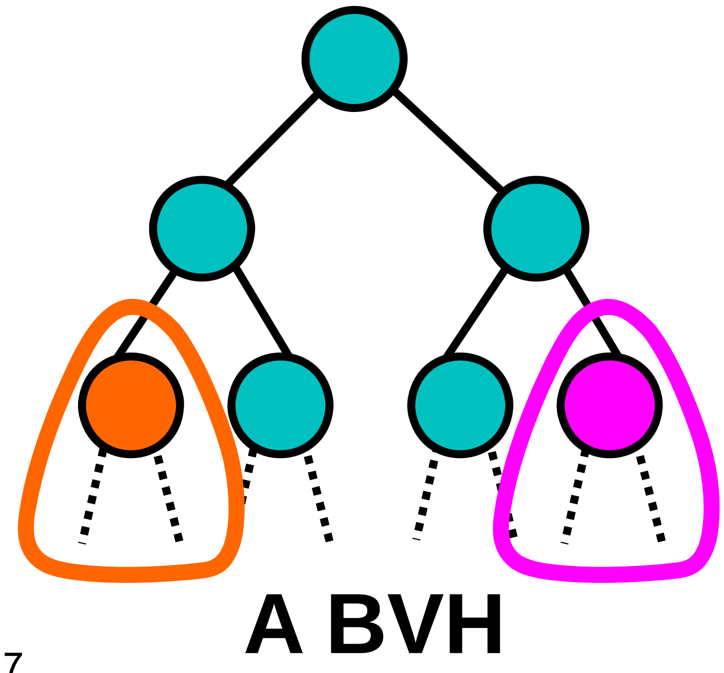
A good metric measuring the quality of BVHs

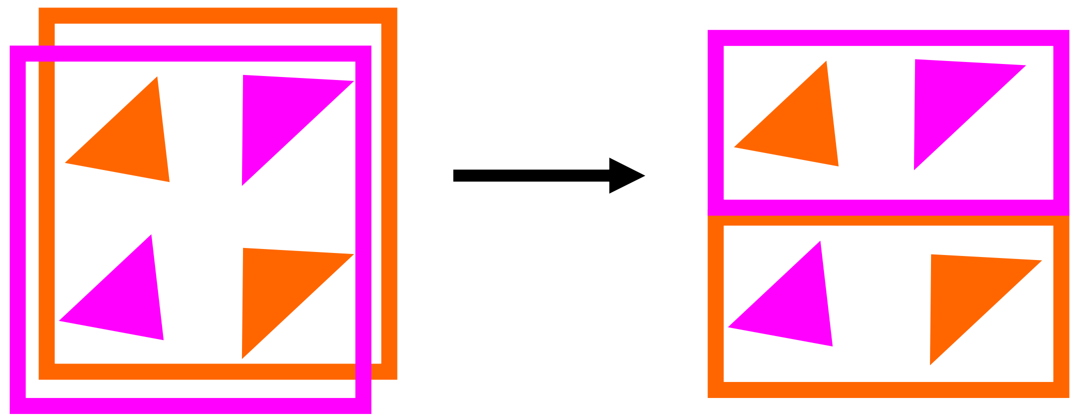
Restructuring Benefit Metric

- Predicts improved culling efficiency when restructuring sub-BVHs
  - Should not perform actual restructuring
- Restructure the sub-BVHs
  - Only if the restructuring benefit is bigger than the restructuring cost

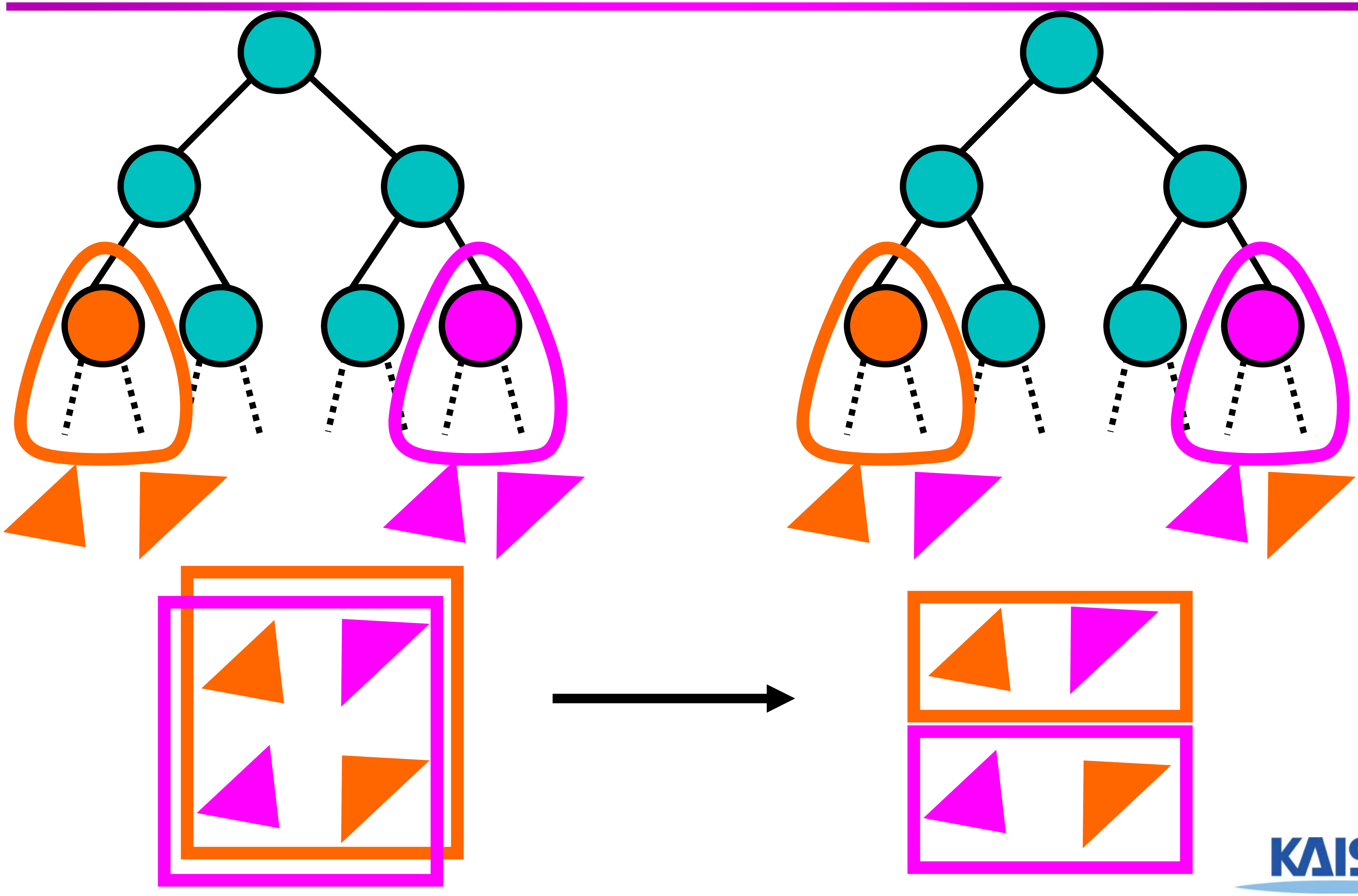
Major Observation

- Restructuring two nodes with BV overlaps can improve the culling efficiency
  - Assumes that restructuring operation will remove all the BV overlaps



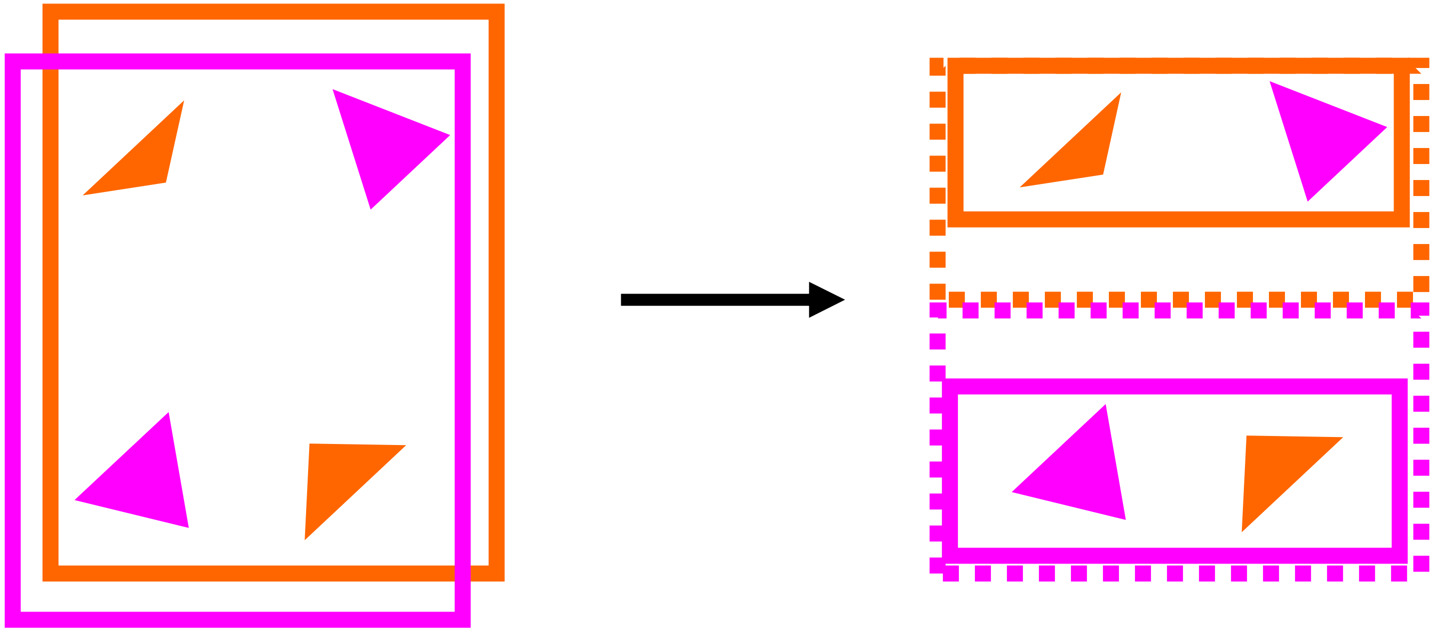


Selective Restructuring Operations



Validation of Restructuring Benefit Metric

- **Compare the expected values against the observed values**
  - **80% of the observed values are 25% off from the expected values**



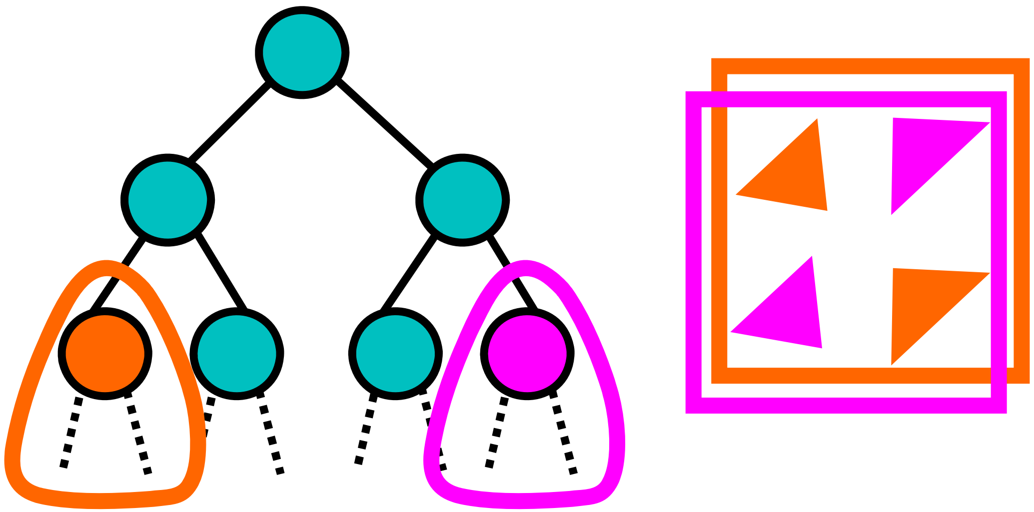
KAIST

Overall Framework

- **At a new frame**
  - **Refits BVs with deformed triangles**
  - **Performs our selective restructuring algorithm**
  - **Runs BVH-based ray tracing**

Detecting BV Overlaps

- **Brute-force method**
  - **Requires O(m^2) where m is # of BVs**
- **Hierarchical traversal and culling**
  - **Inspired by efficient collision detection methods**

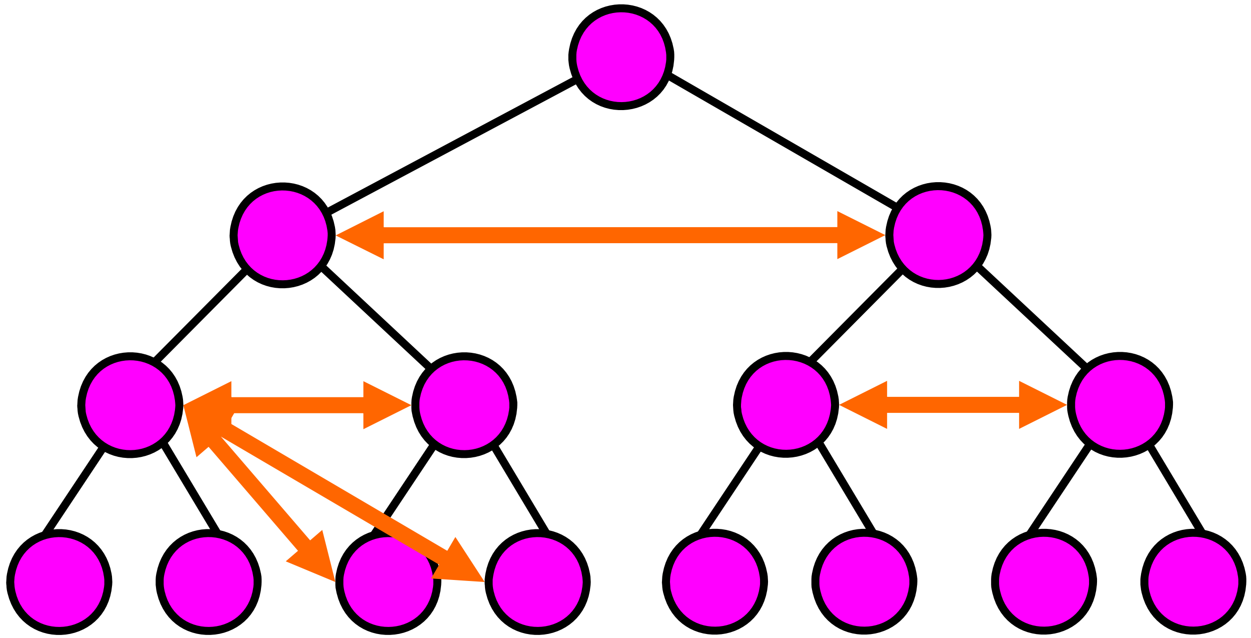


Overview of
Selective Restructuring Algorithm

- **Hierarchical refinement phase**
- **Restructuring phase**

Overview of Selective Restructuring Algorithm

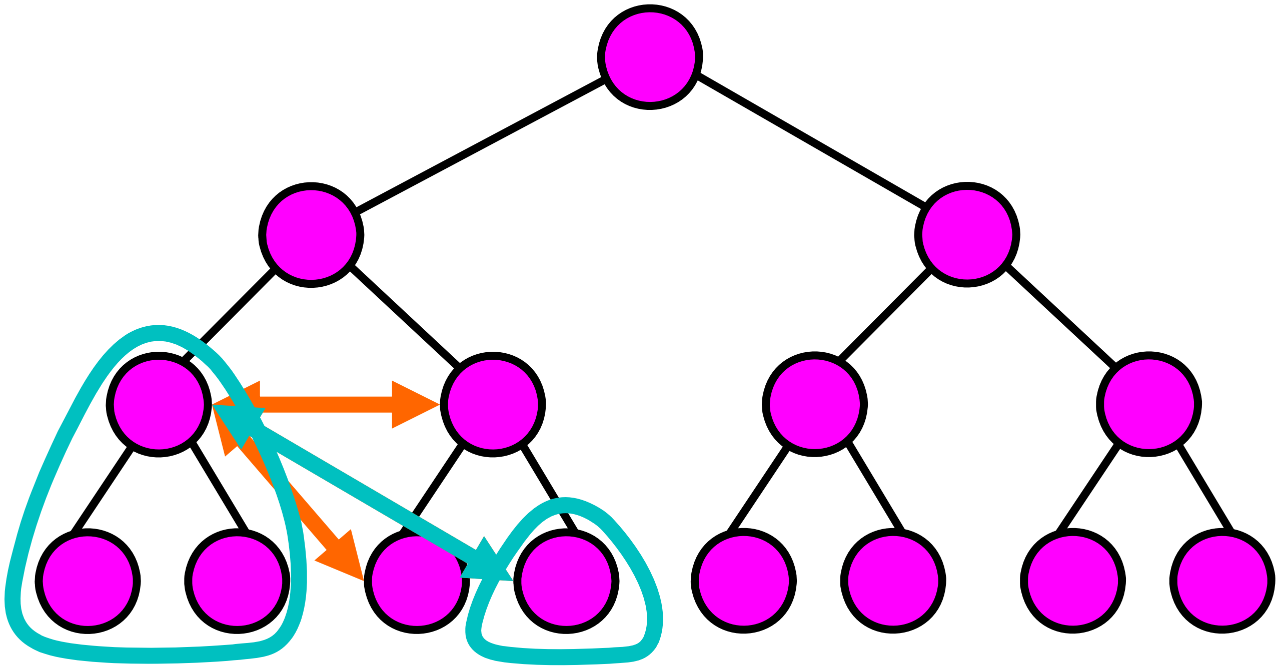
- Hierarchical refinement phase
  - Detects nodes with BV overlaps during hierarchy traversal
- Restructuring phase



KAIST

Overview of Selective Restructuring Algorithm

- Hierarchical refinement phase
- Restructuring phase
  - Restructure node pairs with higher benefits in a greedy manner

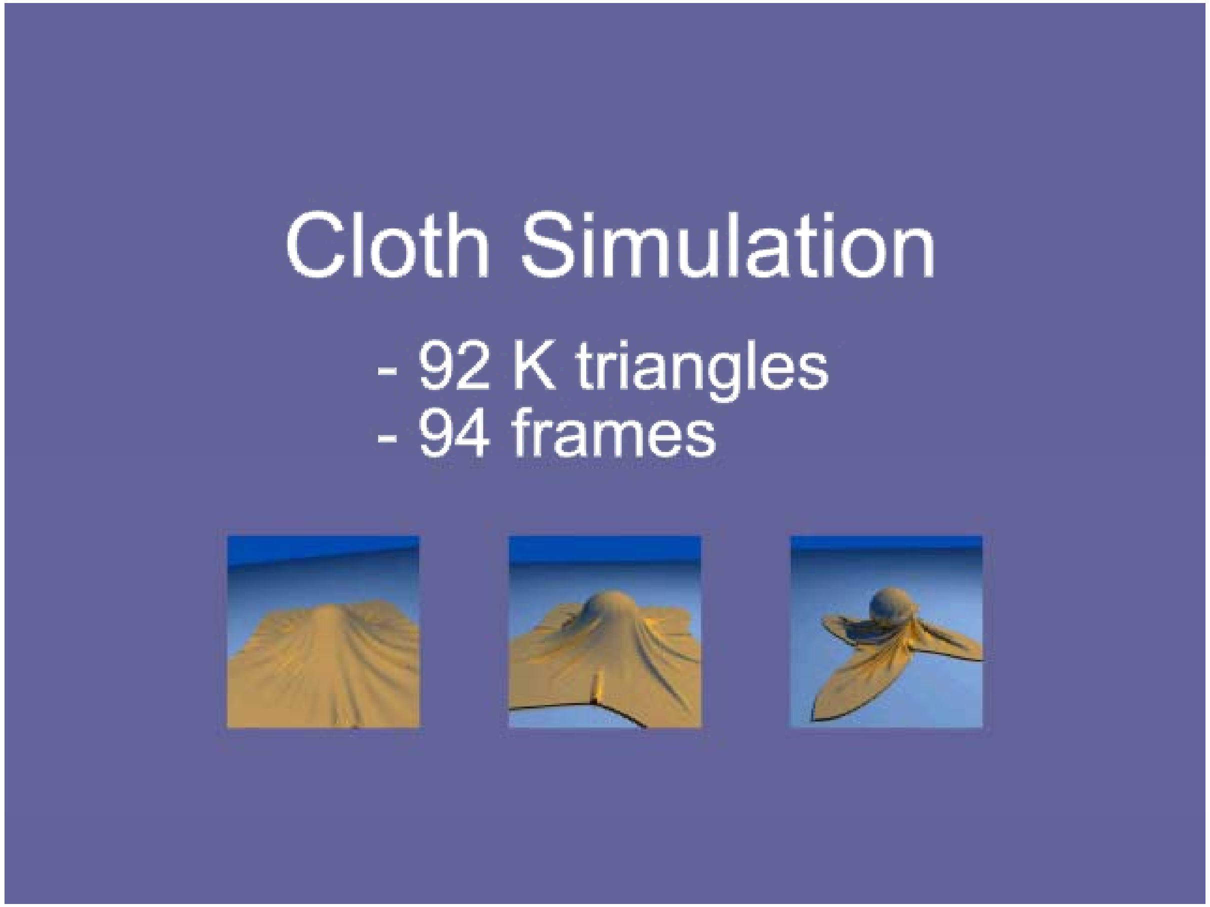


Evaluating Our Algorithm

- **Implement BVH-based ray tracer [Lauterbach et al. 06]**
  - **Tests with four dynamic scenes having different characteristics**

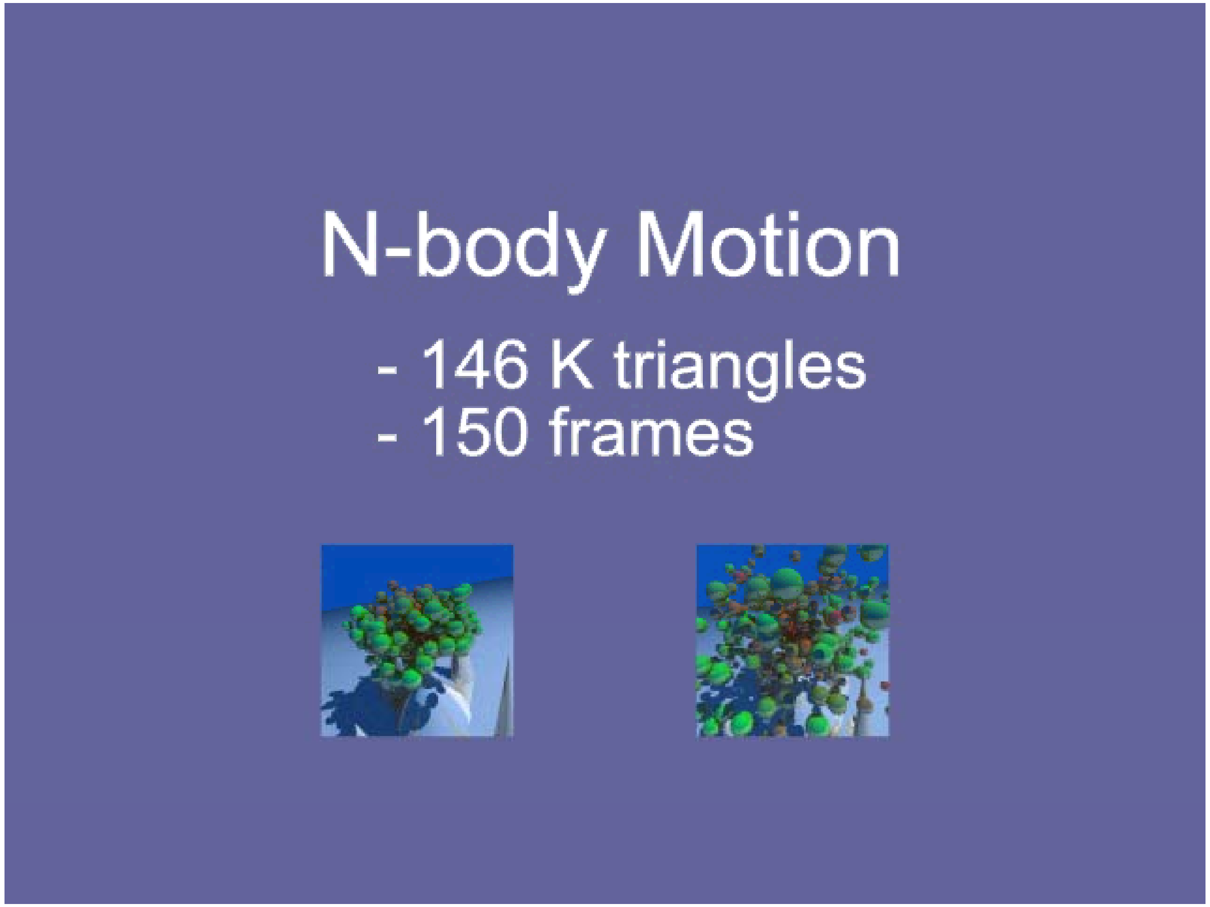
Dynamic Scenes

- **Cloth simulation (92K)**



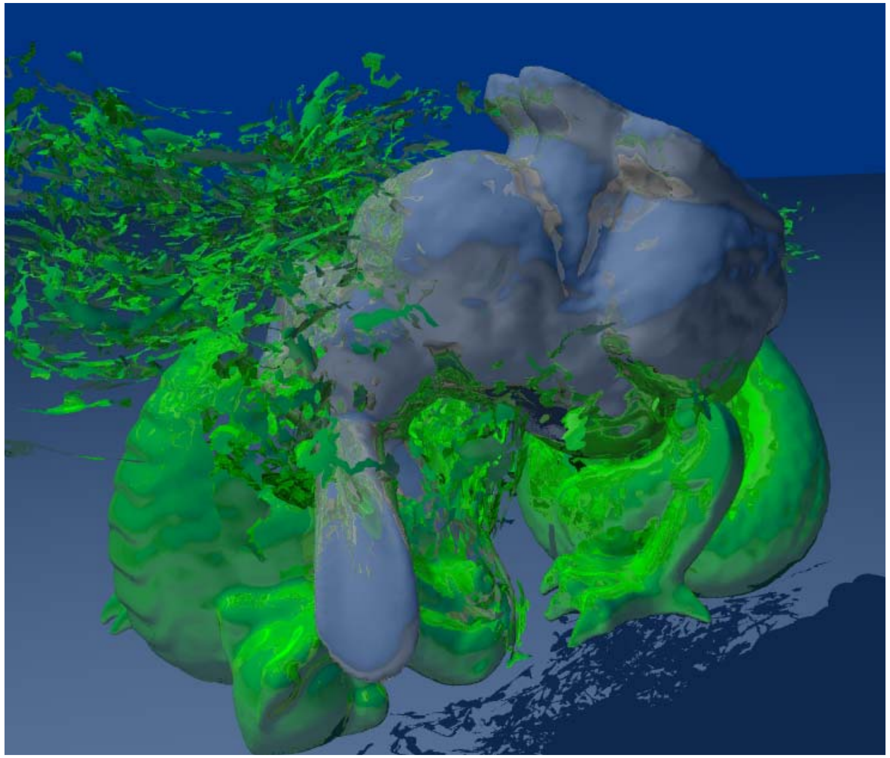
Dynamic Scenes

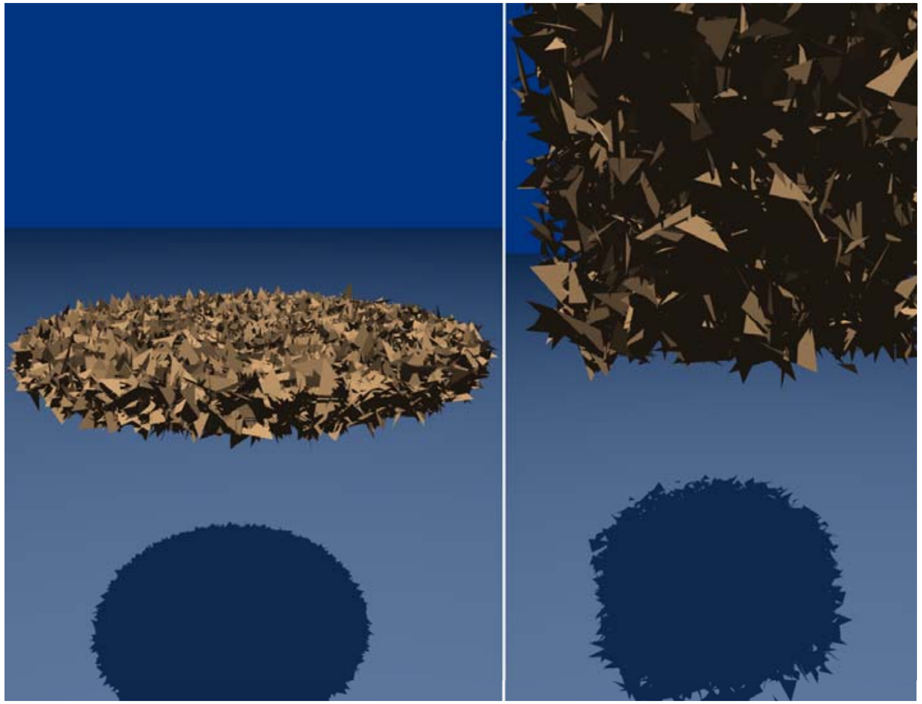
- **N-body simulation (146K)**



Dynamic Scenes

- **Exploding dragon (252K)**
- **BART (65K)**





Prior Works

- BV Refitting [Wald et al. 07, Bergen 97]
- Complete re-construction from scratch
- Other two hybrid methods
  - Based on a simple heuristic
  - RT-Deform [Lauterbach et al. 06]
  - LM method [Larsson and Akenine-Möller 06]

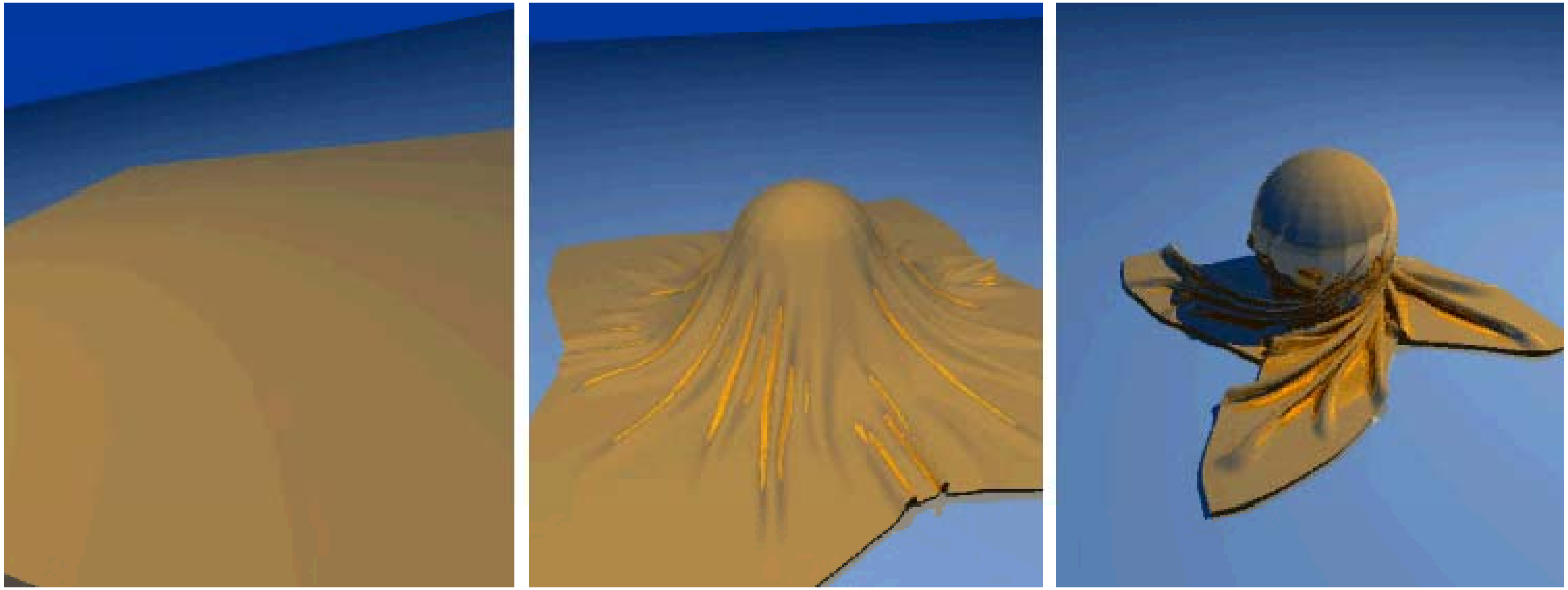
Performance Improvement Ratio

	Complete re-construction	Refitting only
Exploding dragon	8.5	11
N-body simulation	1.8	> 80
BART	1.1	28
Cloth simulation	4.7	0.96

KAIST

Image Shots from Cloth Simulation

Initial frame



Performance Improvement Ratio

Robust performance improvement across our benchmarks

	Complete const.	Refitting only	RT-Deform	LM method
Exploding dragon	8.5	11	**1.65**	**2.16**
N-body simulation	1.8	> 80	**1.25**	**1.36**
BART	1.1	28	**2.5**	**1.11**
Cloth simulation	4.7	0.96	**1.03**	**1.29**

Conclusions

- **Novel algorithm to selectively restructure BVHs**
  - **Based on selective restructuring operations and two BVH metrics**

- Dynamic scenes are available

At Next Class

- Will study collision detection