Parallel Monte Carlo Radiosity

Marijn Koetzier

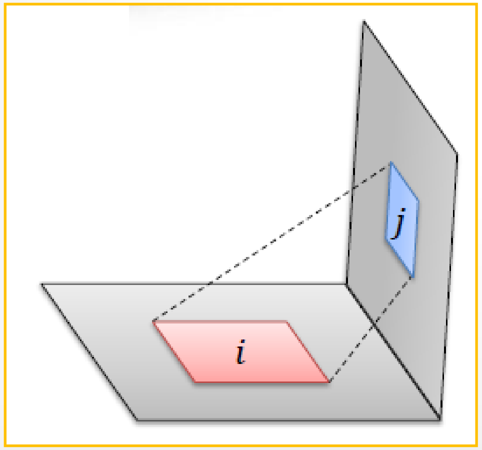
Table of Contents

Introduction

- Paper is state of the art (2012)
- Incremental Stochastic Radiosity
- Focuses on high performance
- Single scene, multiple processors
- Static scenes

Monte Carlo Radiosity

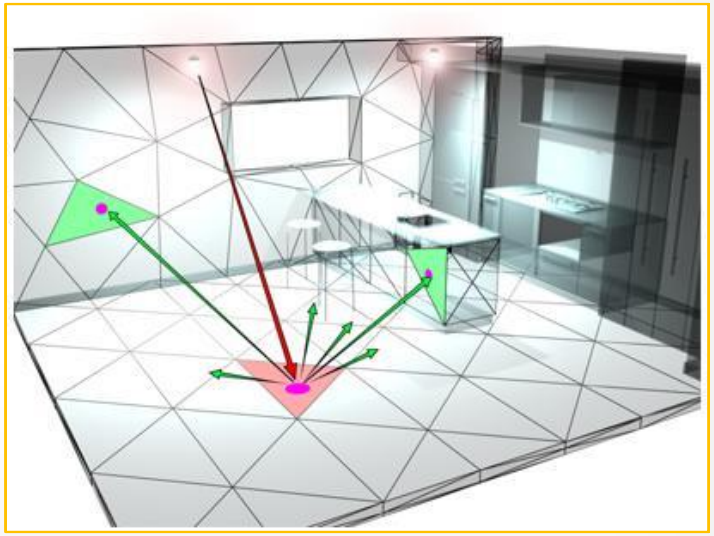
- Approximization by sampling
- Cast Nr_i rays from source surface S_i
- Random origins, direction cosine distribution
- Form factor for destination surface S_j



$$F_{ij} = \frac{Nr_{ij}}{Nr_i}$$

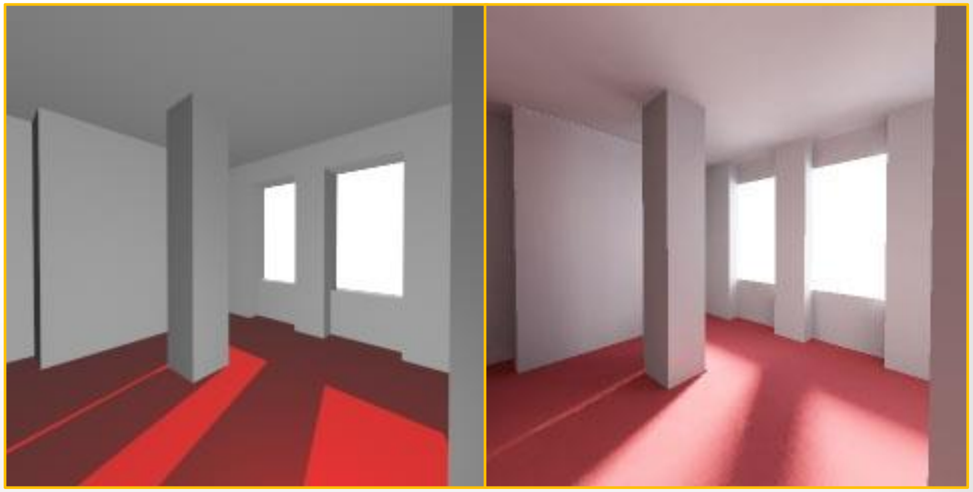
Incremental Stochastic Radiosity

- Jacobi iterative method
- Converged when energy from total scene in previous iteration is close to current iteration
- Iterate until converged



Effects of Radiosity

- Rays of light from light source to surface
- Rays of light from surface to surface
- Did I mention rays?
- Results more realistic than direct illumination techniques
- Rays, rays, rays…



Parallel Approach

Basics:

- Each processor dedicated to a sub-scene
- Calculates radiosity for local surface
- Uses local rays and exchanged rays
- Exchange when ray pierces next sub-scene

Scene Partitioning

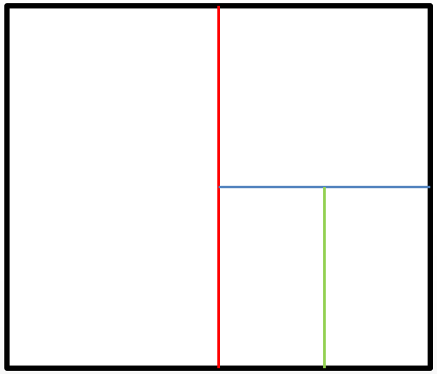
Goals:

- Distribute workload over processors
- Prevent data duplication in memory
- Minimizing communication among processors

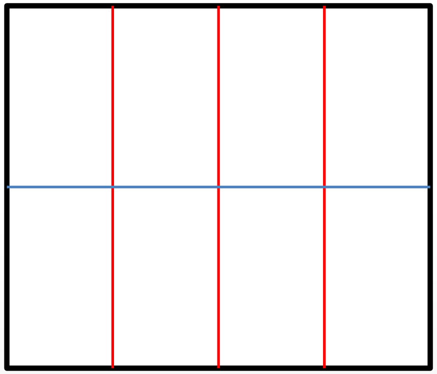
Scene Partitioning

Uniform Convex Partitioning

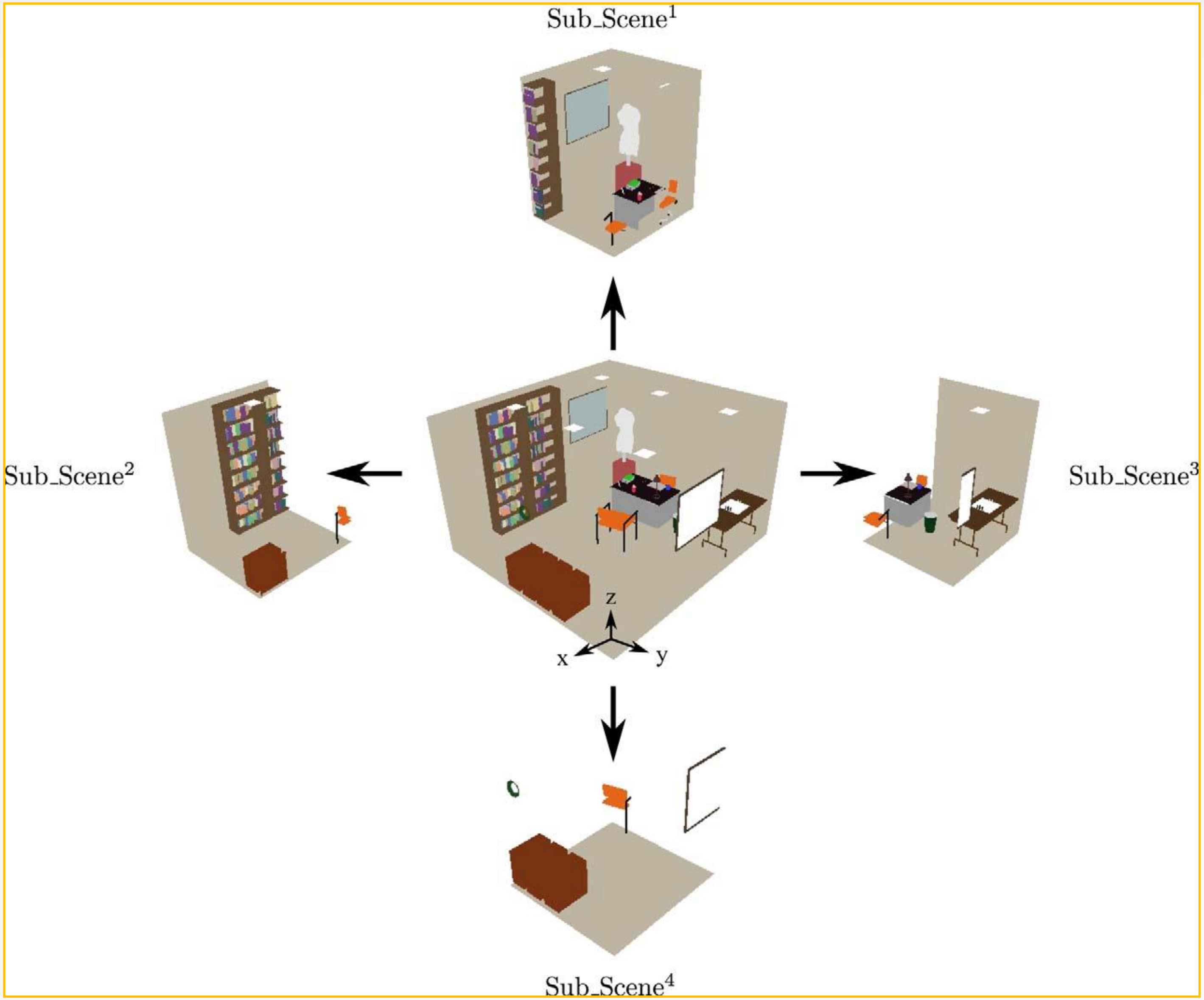
- Split scene into sub-scenes
- Sub-scenes equally large
- Recursive or *a priori*

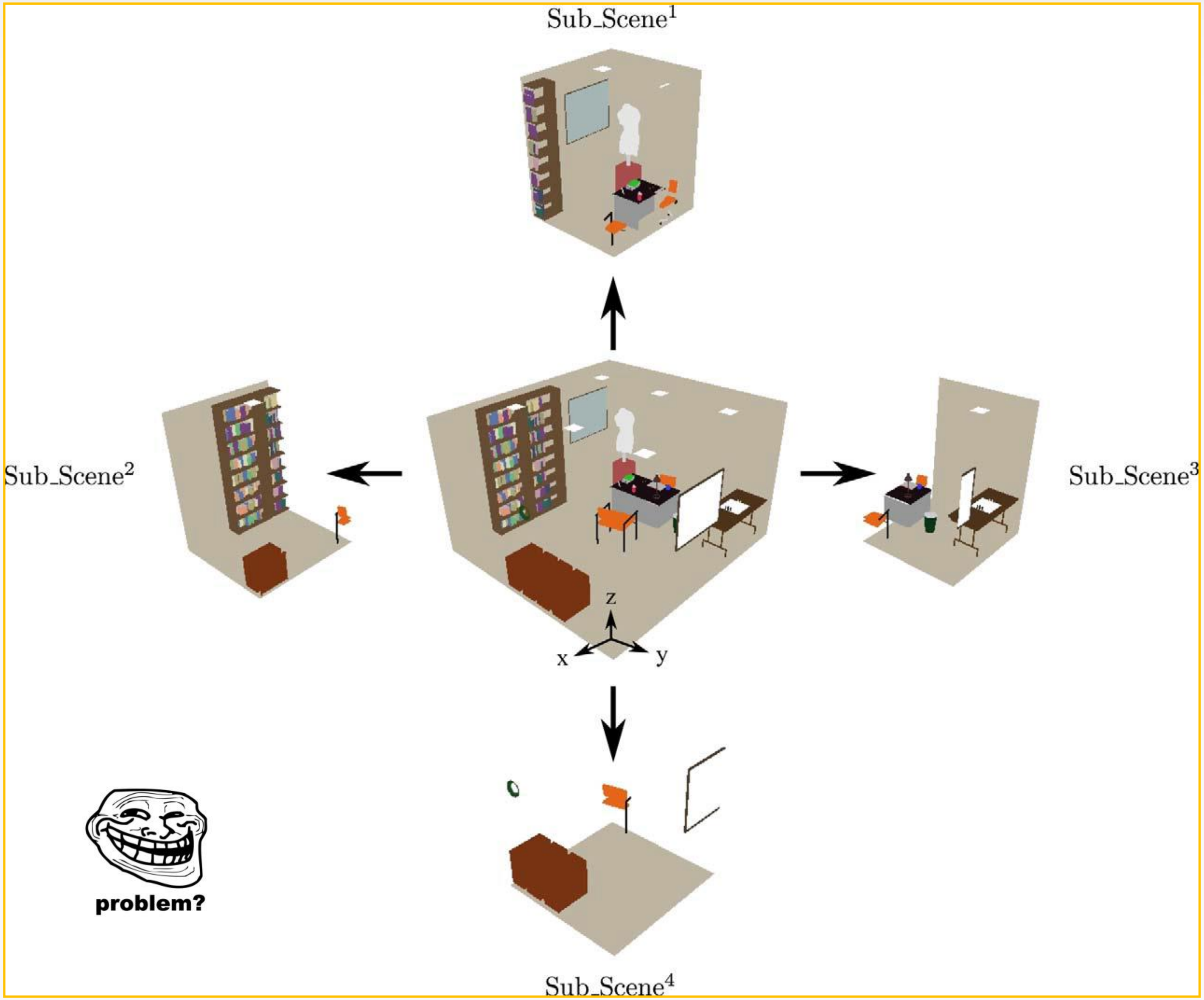


Recursive



A priori


Sub_Scene1
Sub_Scene2
Sub_Scene3
Sub_Scene4
z
x
y



Load balancing?

Scene Partitioning

Non-Uniform Convex Partitioning

- Load imbalance correction
- In terms of surface area
- Select partition which minimizes load variance

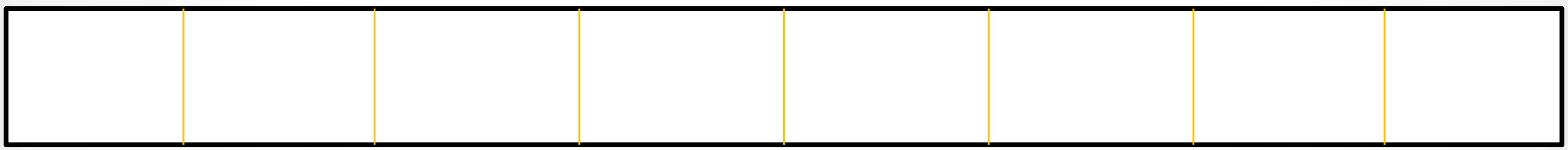
Scene Partitioning

Partition Strategy

- Start at the root of the scene
- Determine $2^C - 1$ candidate splits
- Uniformly distributed candidate splits
- Binary search tree
- For each split, check workload distribution
- Take partition that minimizes workload variance among processors

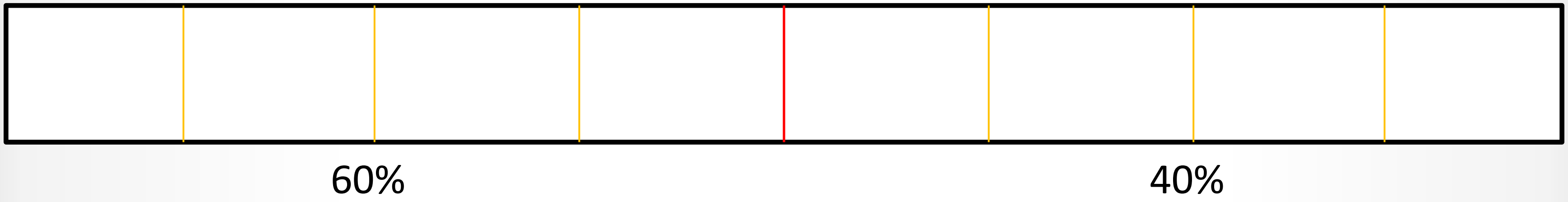
Scene Partitioning

- Example (1 dimension, single split, C = 3):



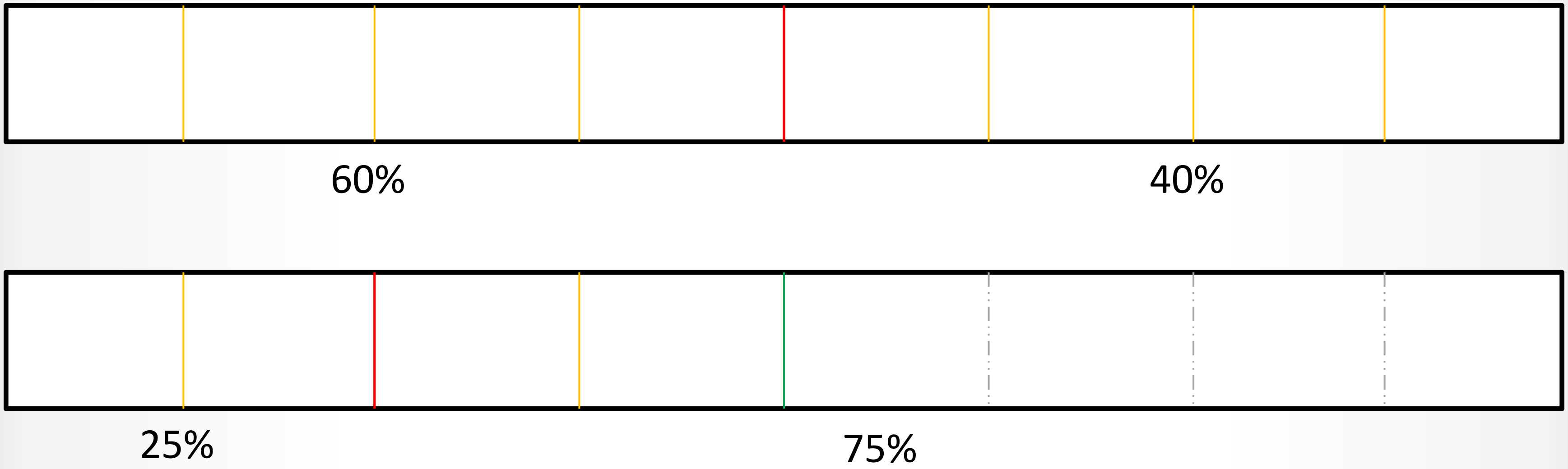
Scene Partitioning

- Example (1 dimension, single split, C = 3):



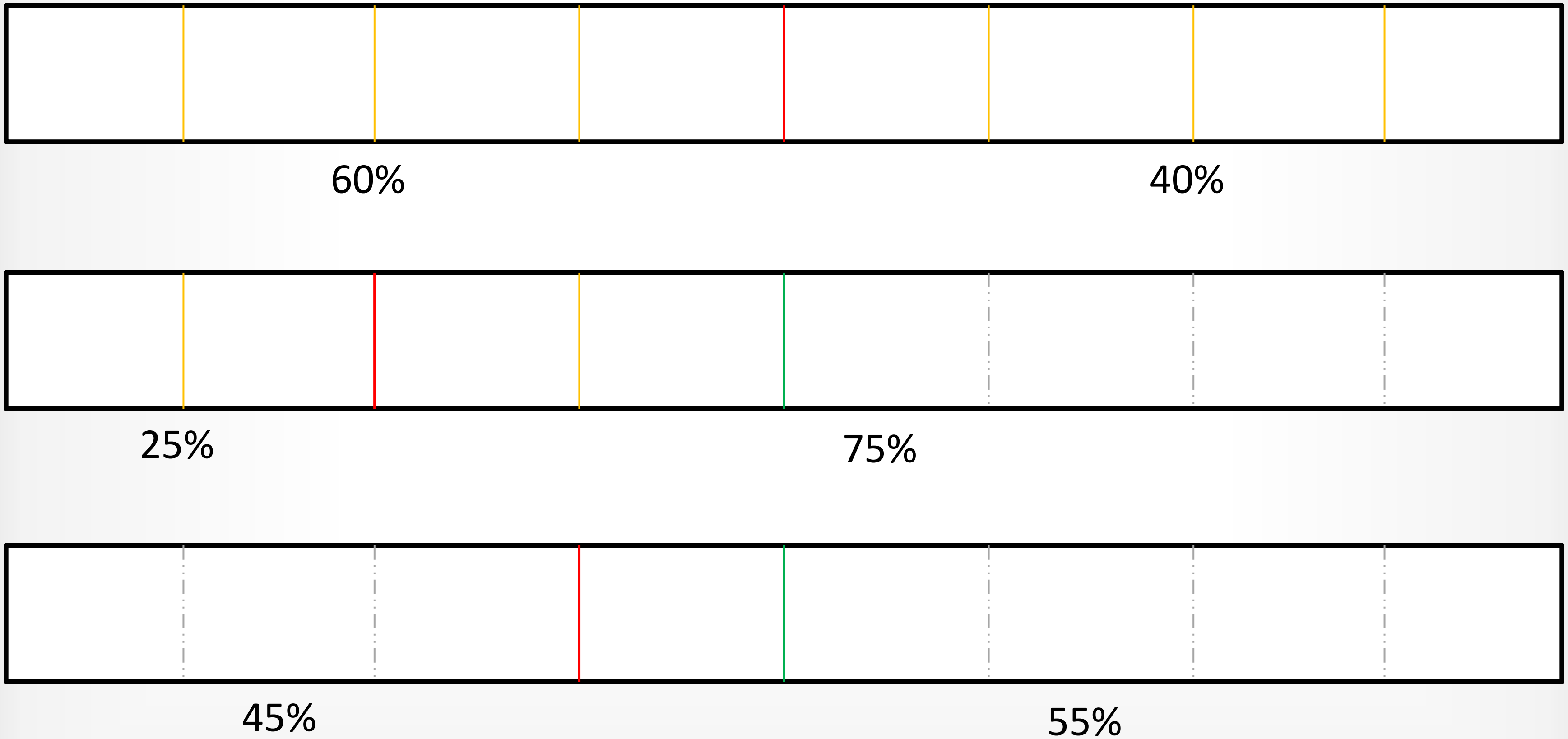
Scene Partitioning

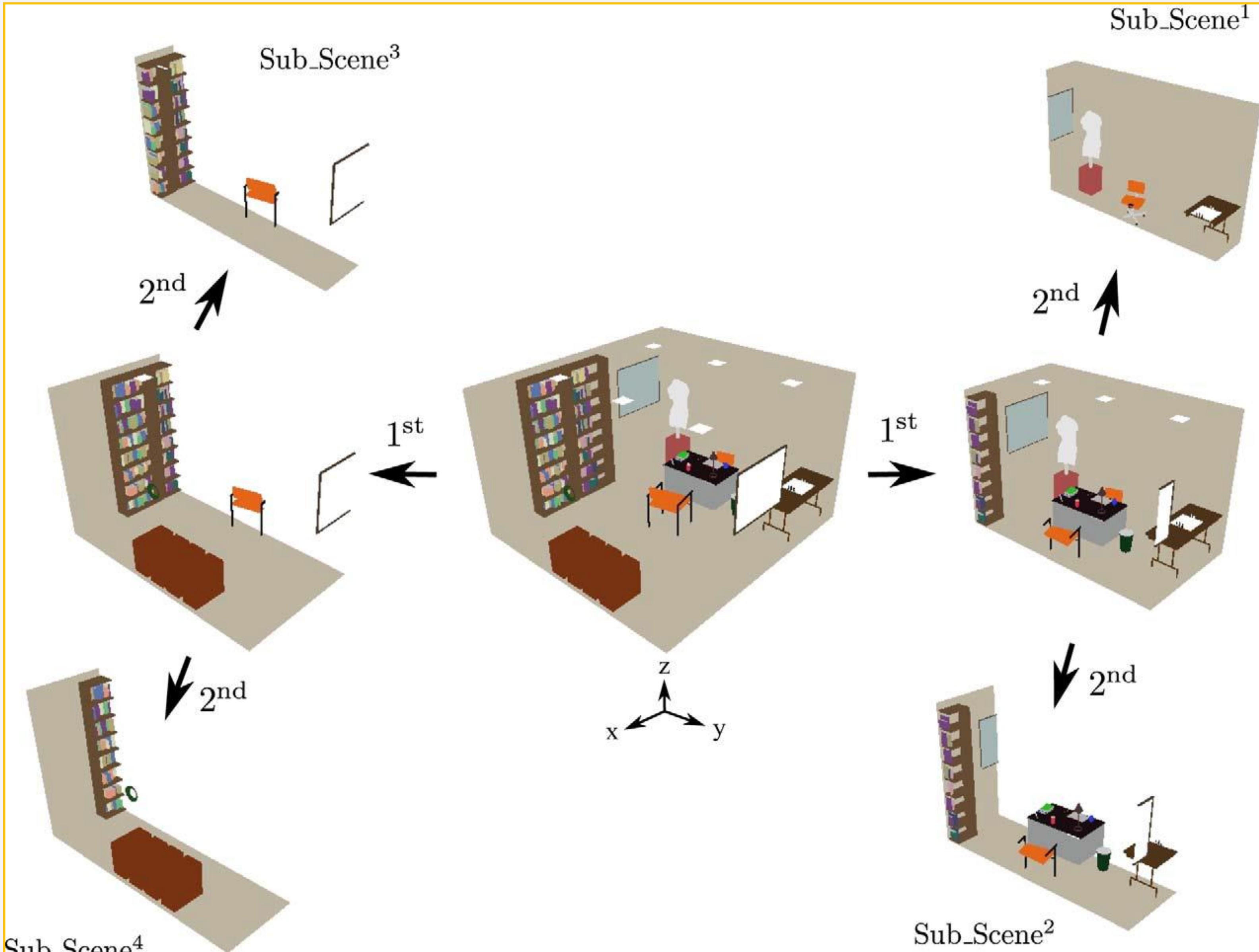
- Example (1 dimension, single split, C = 3):



Scene Partitioning

- Example (1 dimension, single split, C = 3):




Sub_Scene1
Sub_Scene3
2nd
2nd
1st
1st
2nd
2nd
z
x
y
Sub_Scene4
Sub_Scene2

Ray Packing

- Sending rays from/to adjacent sub-scenes
- Large amount of rays being exchanged
- Bundle rays by origin and similar direction
- Exploitation of SIMD
- 2 Packet Buffers for each sub-scene edge
- Fill buffer with propagating rays
- Fixed buffer size, send when buffer is full
- Incomplete packets send at final stage

Ray Packing

- Packet buffer size important
- Small packets: frequent communication
- Large packets: more incomplete packets

Conclusion

- Convex Partitioning
  - No overlap in memory
  - Quick results
  - Might not be as effective as e.g. Decision Forests

- Ray Packing
  - Grouping rays is a hot topic
  - Tackle packet buffer size