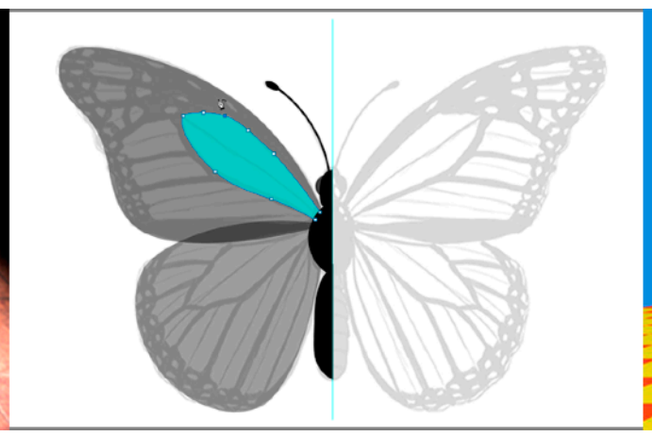
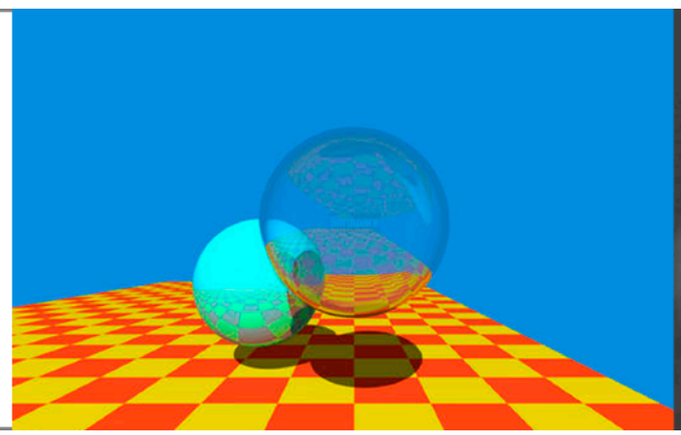
Introduction to Computer Graphics

GAMES101, Lingqi Yan, UC Santa Barbara

Lecture 14:
Ray Tracing 2
(Acceleration & Radiometry)


http://www.cs.ucsb.edu/~lingqi/teaching/games101.html

Announcements

- Grading of resubmissions — we're working on that
- GTC news: DLSS 2.0
  - https://zhuanlan.zhihu.com/p/116211994
- GTC news: RTXGI
  - https://developer.nvidia.com/rtxgi
- Personal feeling
  - Offline rendering techniques will soon become real-time
  - Current real-time rendering techniques will still be useful
- Next lectures won't be easy

Last Lecture

- Why ray tracing?
- Whitted-style ray tracing
- Ray-object intersections
  - Implicit surfaces
  - Triangles
- Axis-Aligned Bounding Boxes (AABBs)
  - Understanding — pairs of slabs
  - Ray-AABB intersection

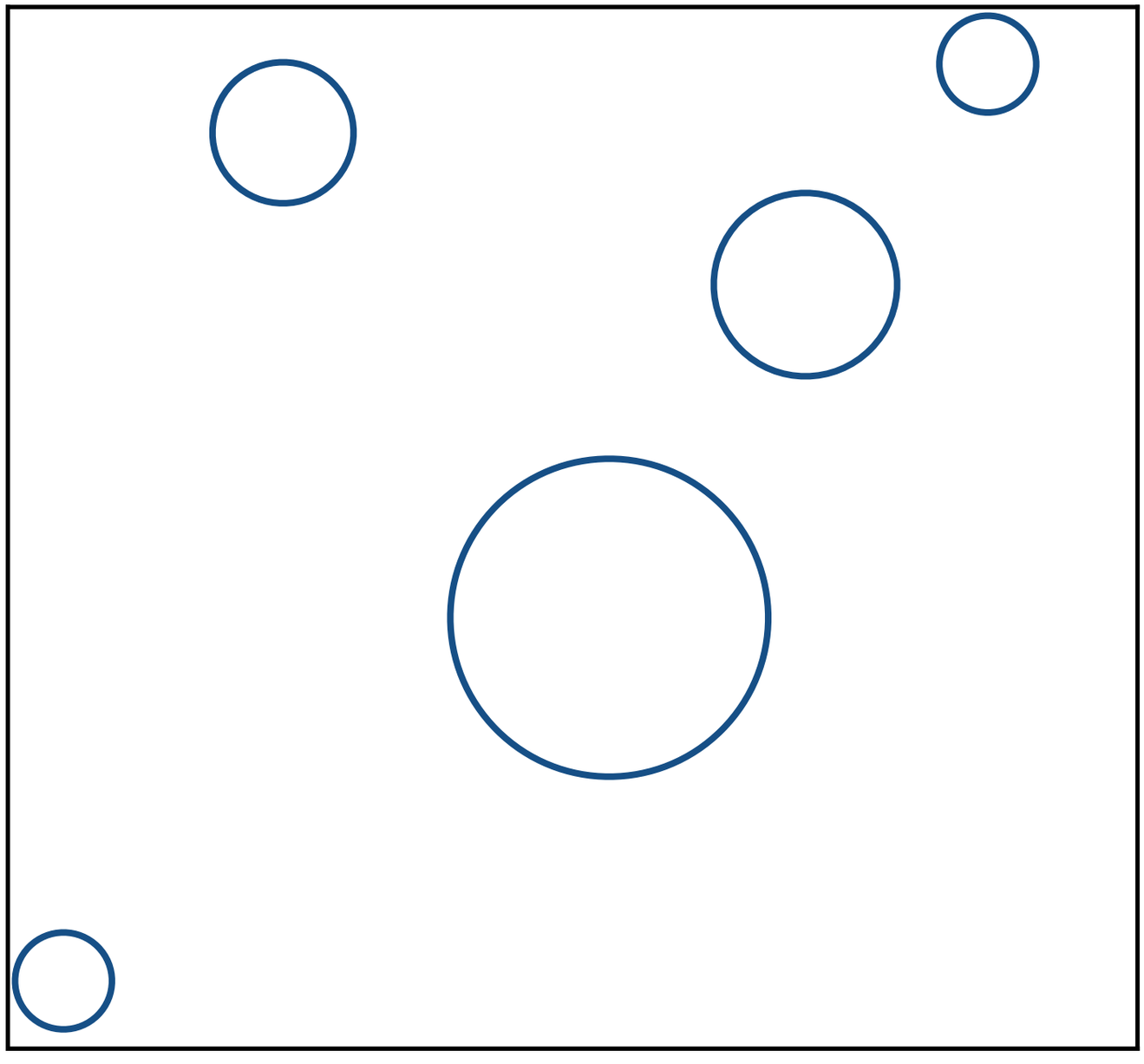
Today

- Using AABBs to accelerate ray tracing
  - Uniform grids
  - Spatial partitions

- Basic radiometry (辐射度量学)

Uniform Spatial Partitions (Grids)

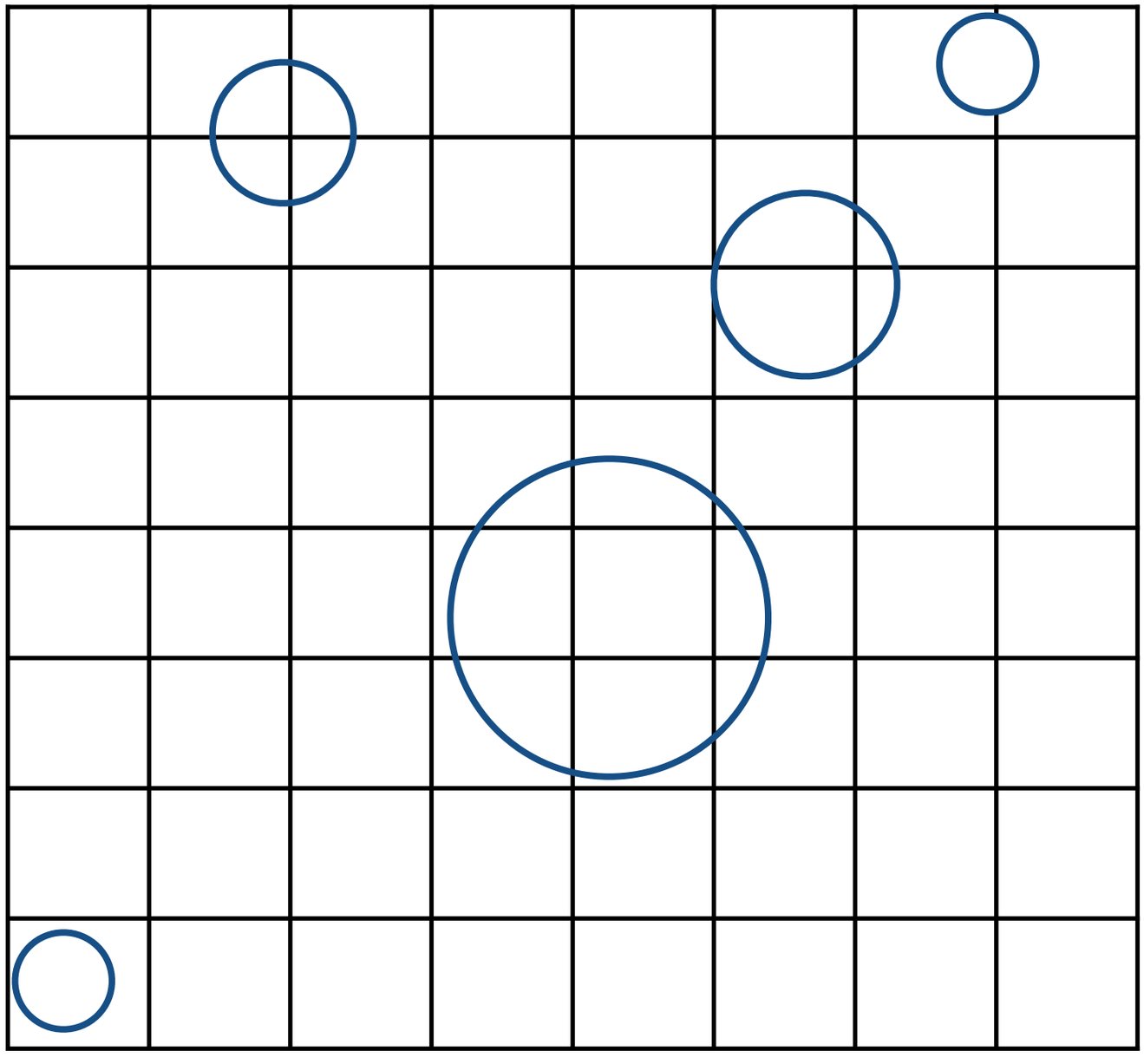
Preprocess – Build Acceleration Grid



1. Find bounding box

   构建一个 bbox 包含所有物体

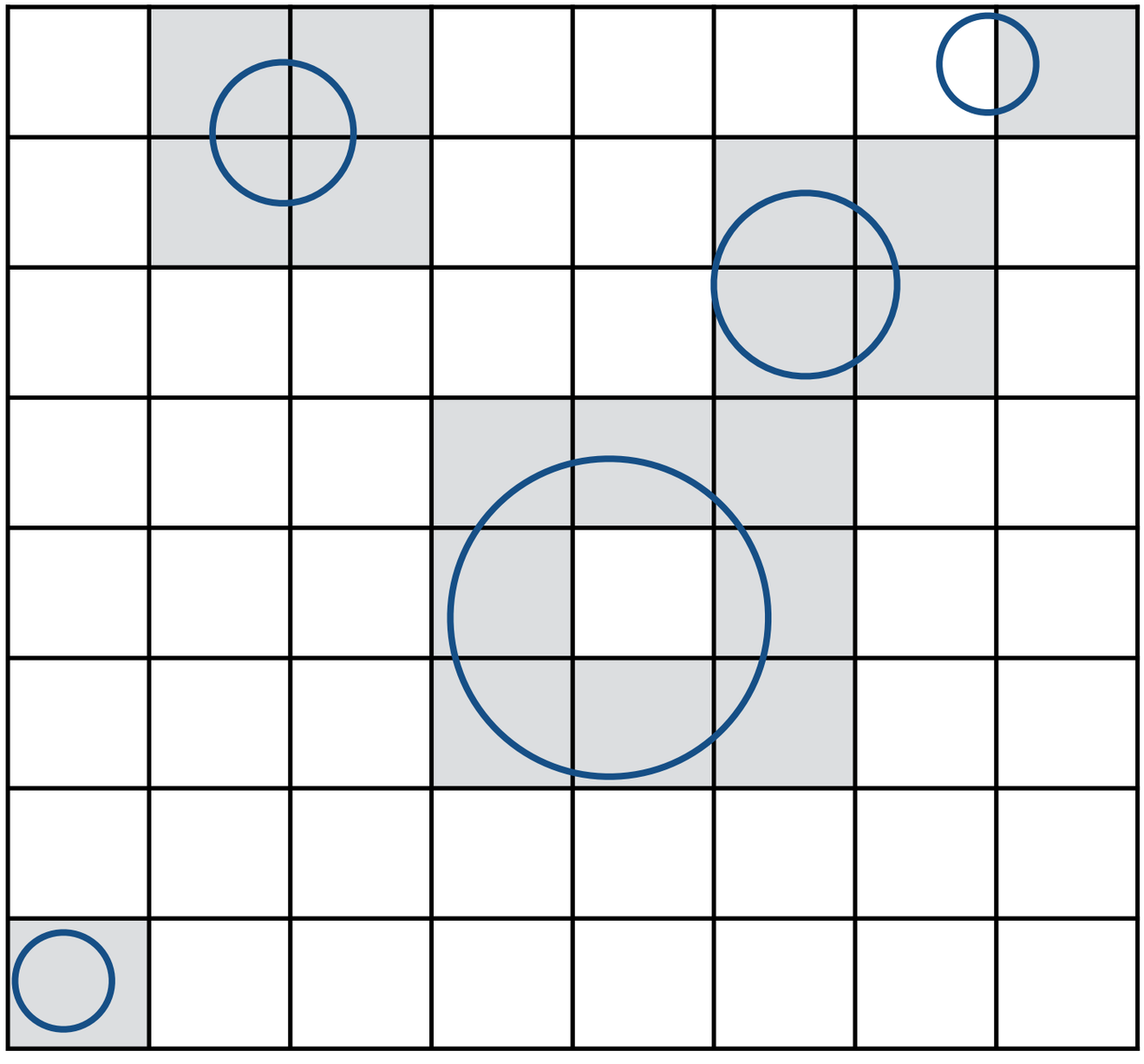
Preprocess – Build Acceleration Grid



1. Find bounding box
2. Create grid

在 bbox 区域内进行网格化分

Preprocess – Build Acceleration Grid



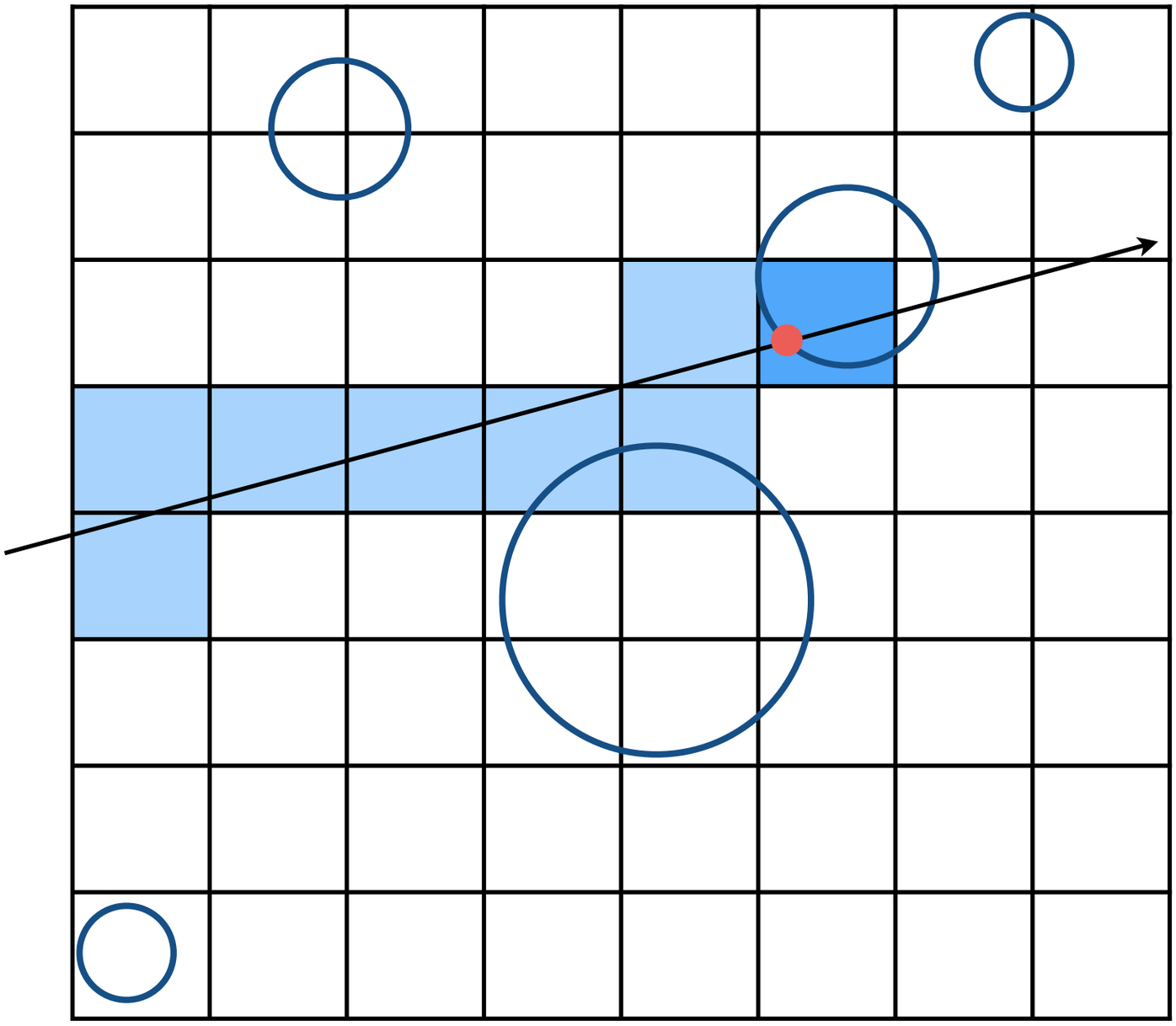
1. Find bounding box
2. Create grid
3. Store each object in overlapping cells

储存 有物体覆盖的 网格

Ray-Scene Intersection

均匀划分的步骤如下：

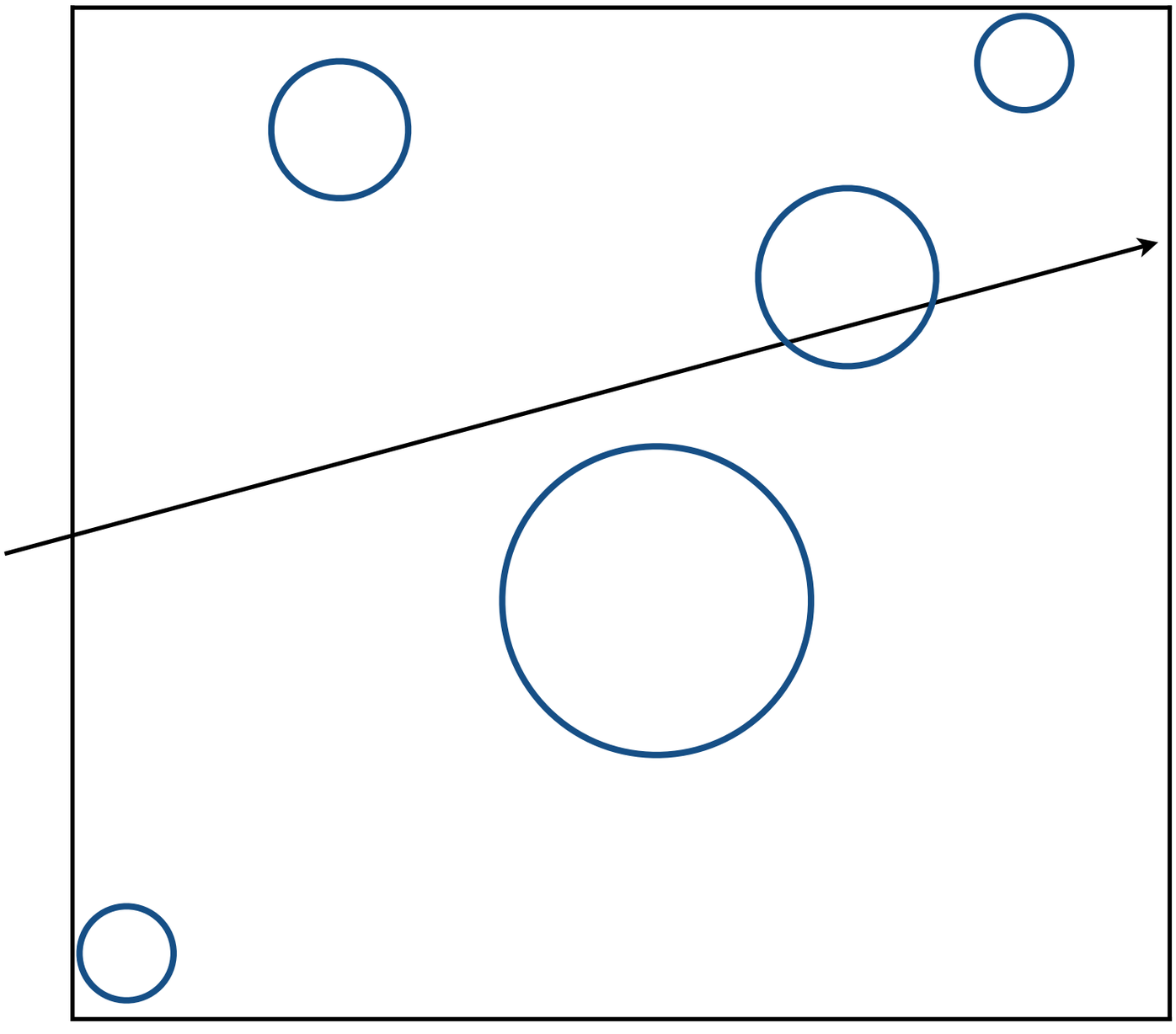
找到场景包围盒 –> 均匀划分该包围盒 –> 判定与物体相交的子包围盒 –> 与物体求交



Step through grid in ray traversal order

For each grid cell
Test intersection
with all objects
stored at that cell

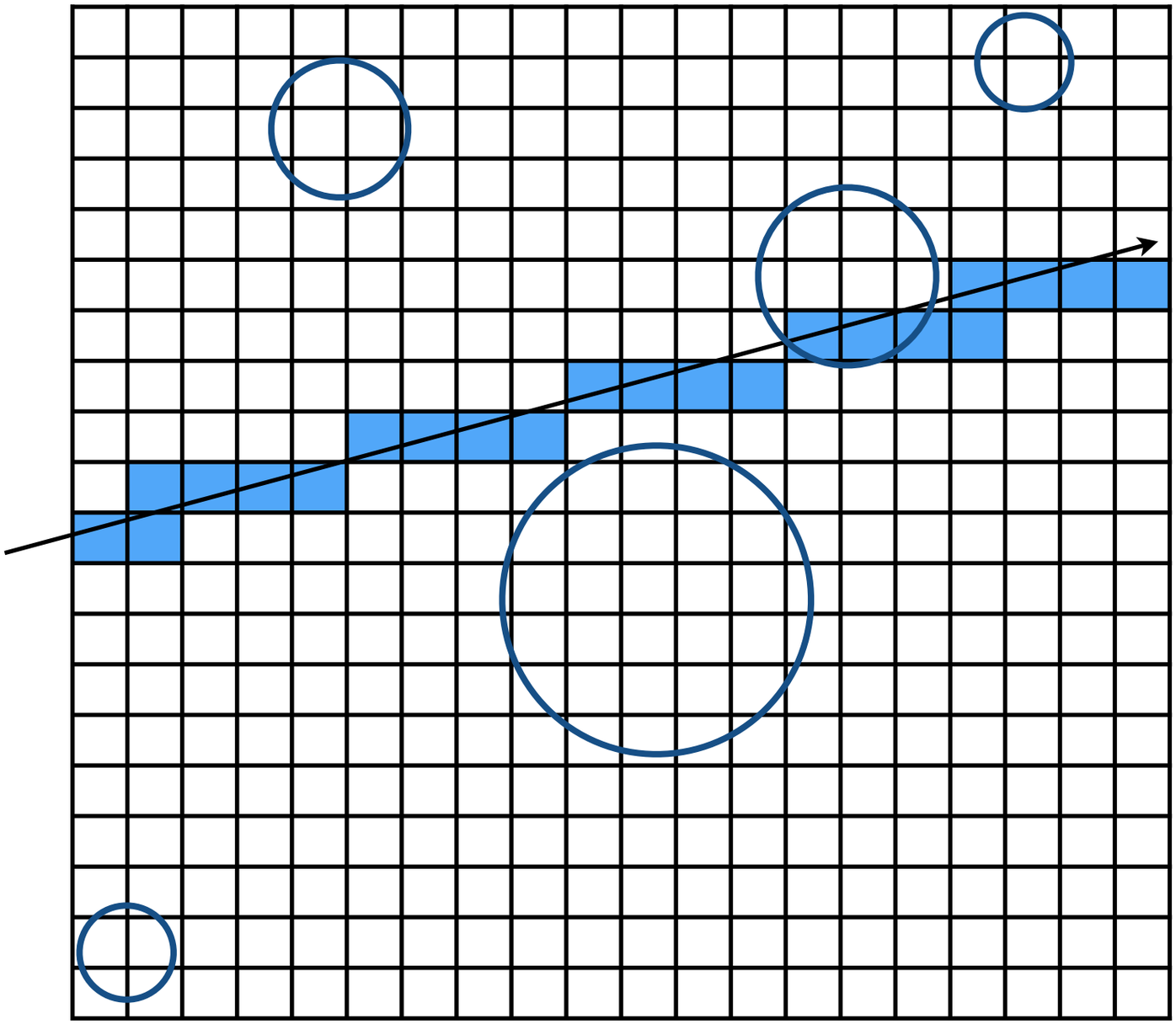
Grid Resolution?



One cell

- No speedup

Grid Resolution?

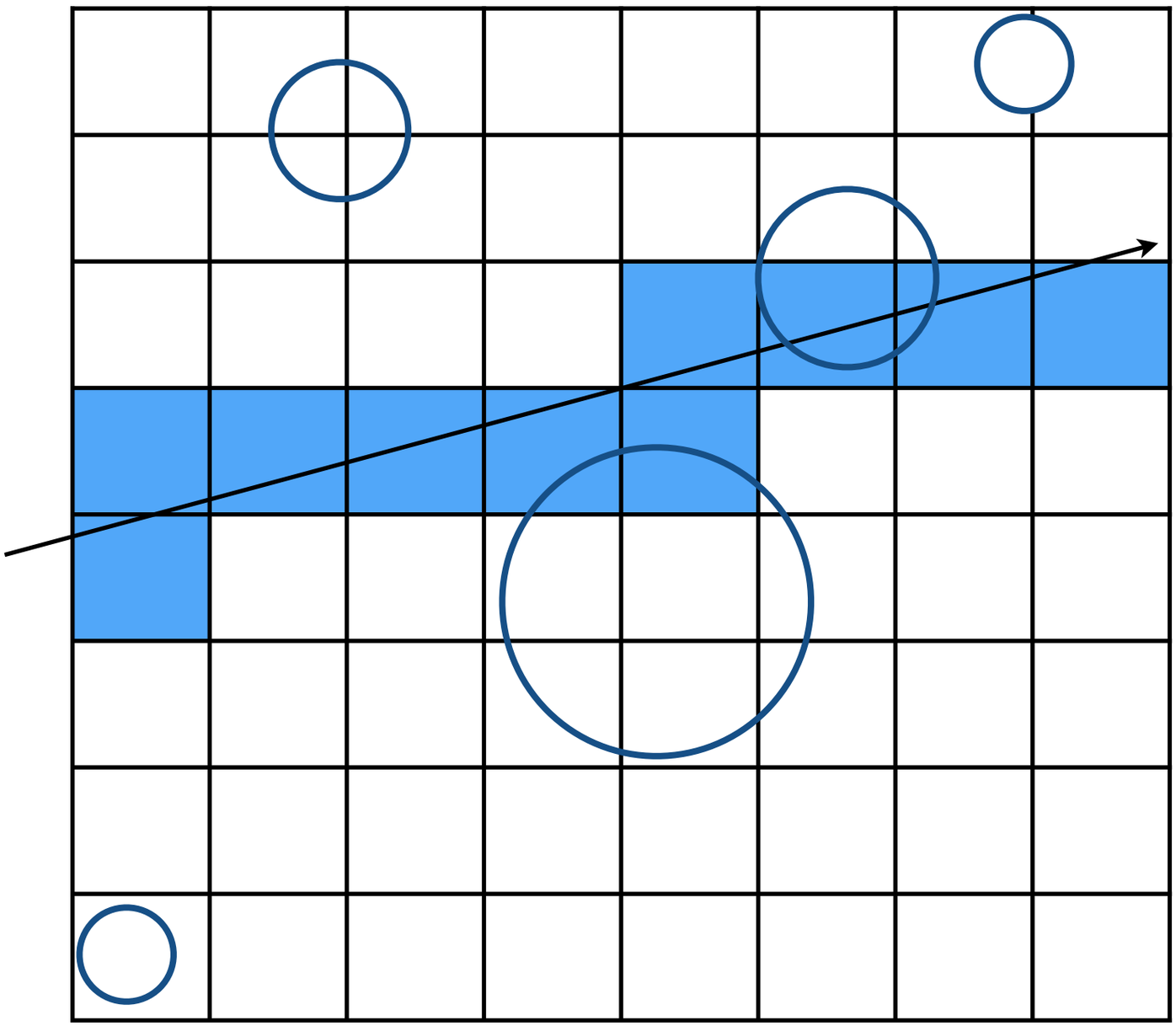


Too many cells

- Inefficiency due to extraneous grid traversal

如果一条光线向右上投射出去，为了确定这条光线与哪些子包围盒相交，就是光栅化一条光线，确定这些网格是否被保存，如果保存了，就计算在网格内的交点

Grid Resolution?



Heuristic:

- #cells = C * #objs
- C ≈ 27 in 3D

加速就是多做光线与网格求交，少做光线与物体求交

Uniform Grids – When They Work Well

Grids work well on large collections of objects

that are distributed evenly in size and space

Uniform Grids – When They Fail

“Teapot in a stadium” problem

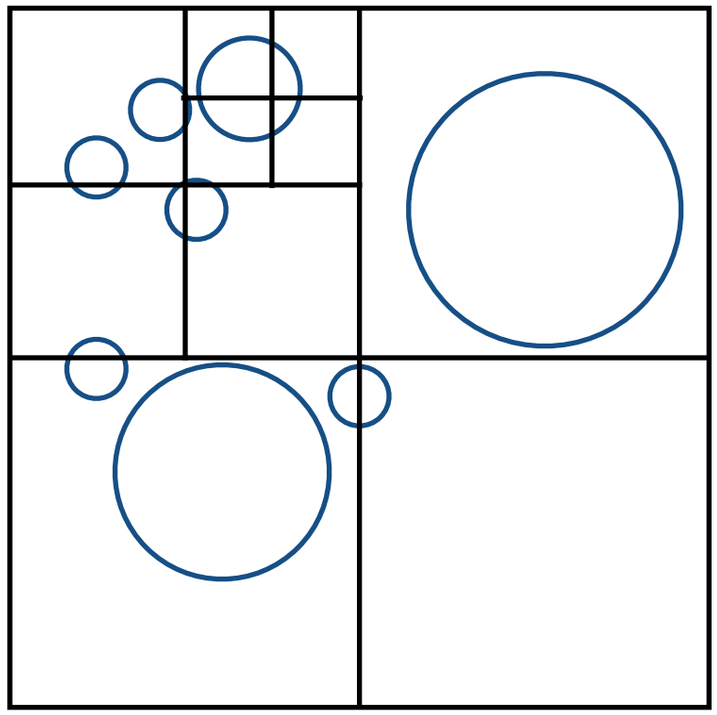
稀疏物体场景表现很差

Spatial Partitions

在网格均匀划分中划分出来的都是大小相同的格子，但在有些空旷的地方不需要这样划分，太浪费了，我们想在没物体的地方用大盒子，有物体的地方用密集的盒子，这也就引出了空间划分的方法

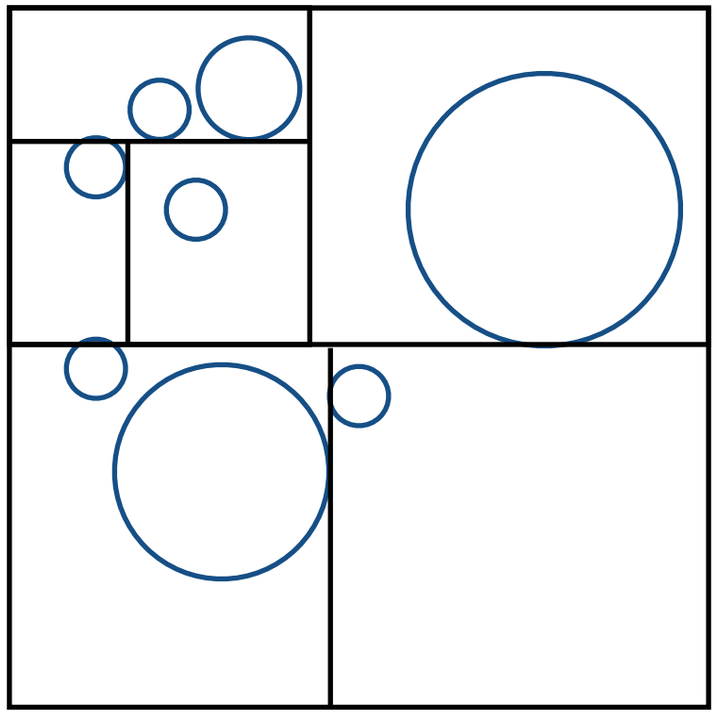
Spatial Partitioning Examples

每一次把空间划分成八份，直到满足一定的停止规则（比如某一次划分8个子空间中7个为空），缺点：维数越高越复杂



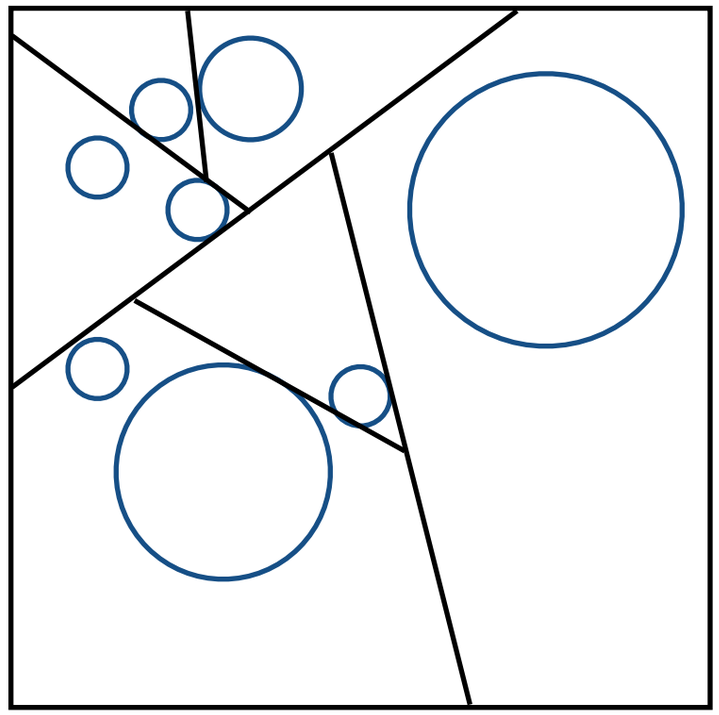
Oct-Tree

一种对空间二分的划分方法，每次选一个方向进行划分，与KD树的区别在于它不是横平竖直地切，且它会有越高维越不好计算的问题（砍开二维用线，砍开三维用面，维度越高越复杂）



KD-Tree

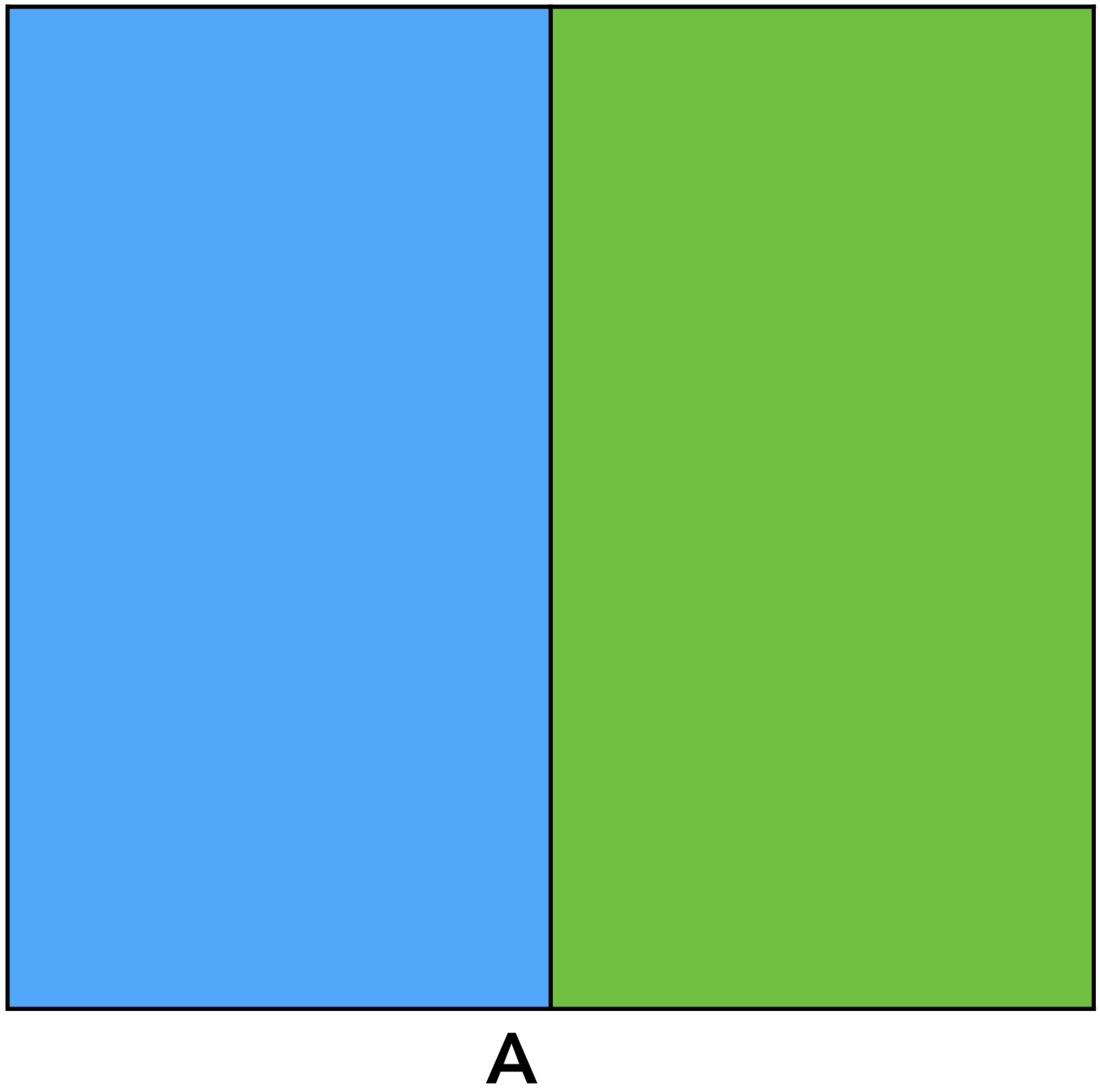
每次把空间划分为两份，x，y，z轴轮流切分，直到被切分节点中不存在物体则停止

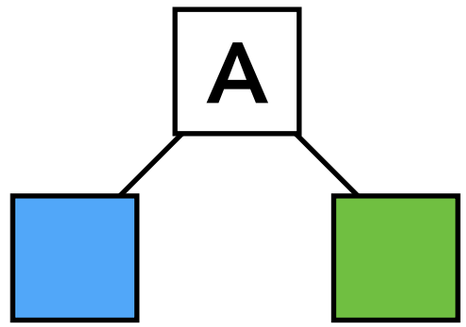


BSP-Tree

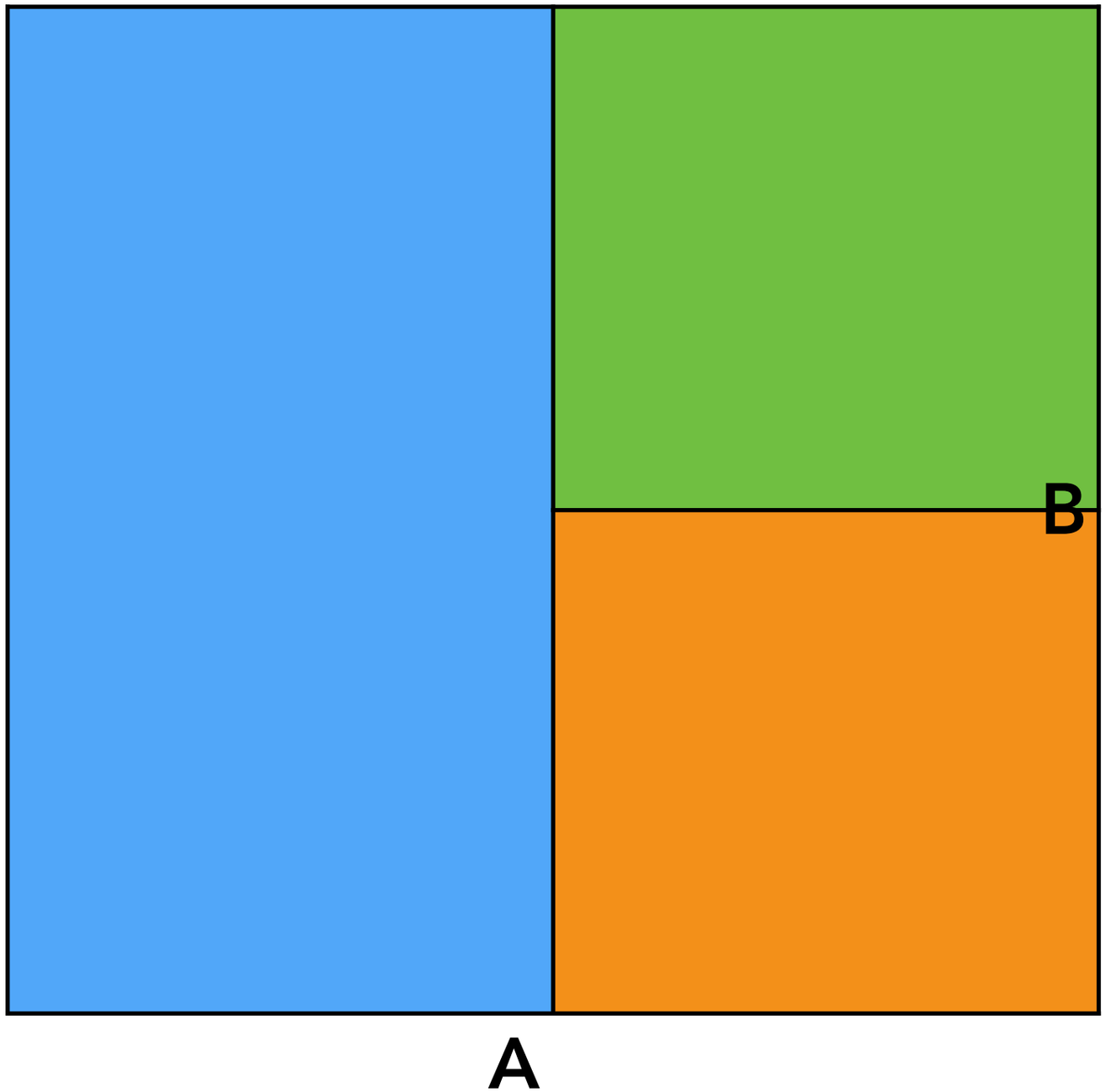
Note: you could have these in both 2D and 3D. In lecture we will illustrate principles in 2D.

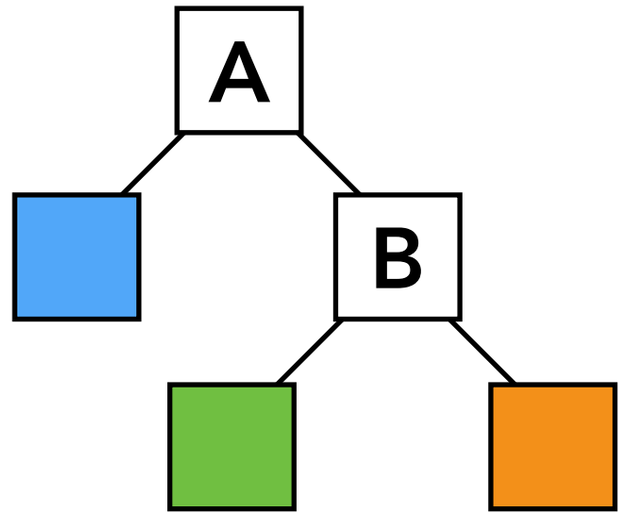
KD-Tree Pre-Processing



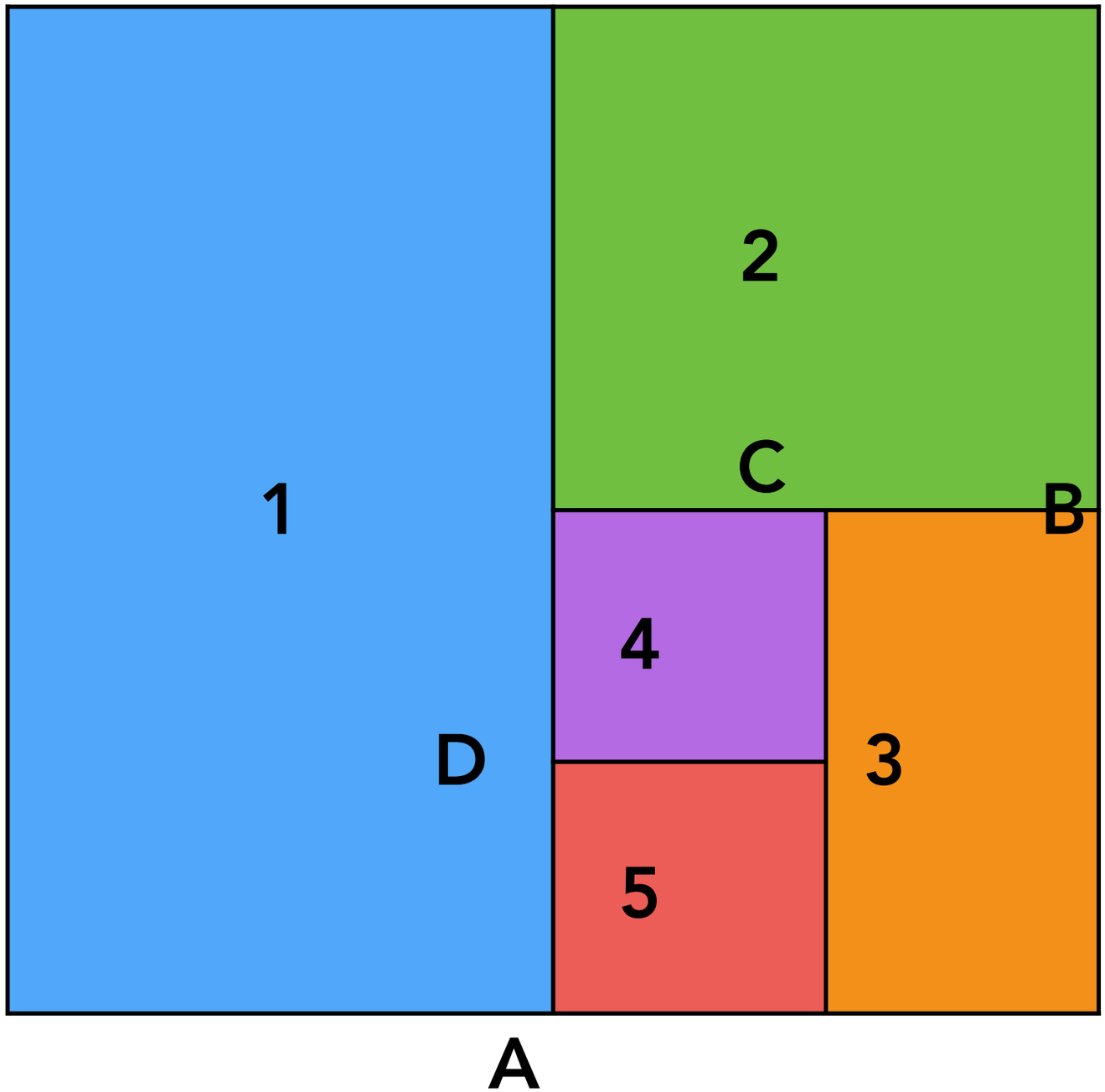


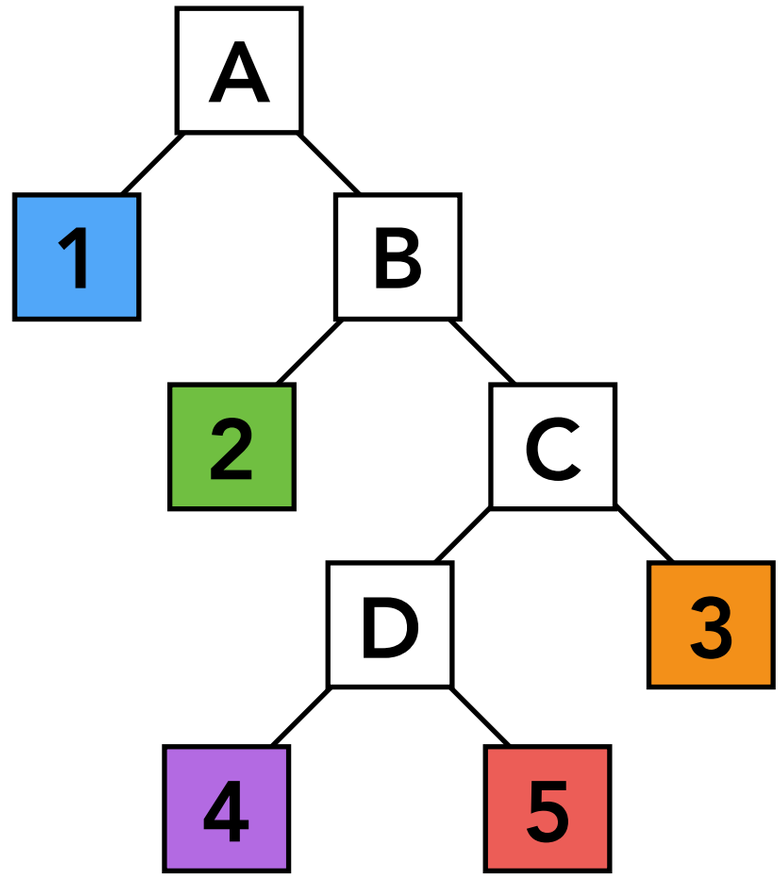
KD-Tree Pre-Processing





KD-Tree Pre-Processing





Note: also subdivide nodes 1 and 2, etc.

Data Structure for KD-Trees

Internal nodes store

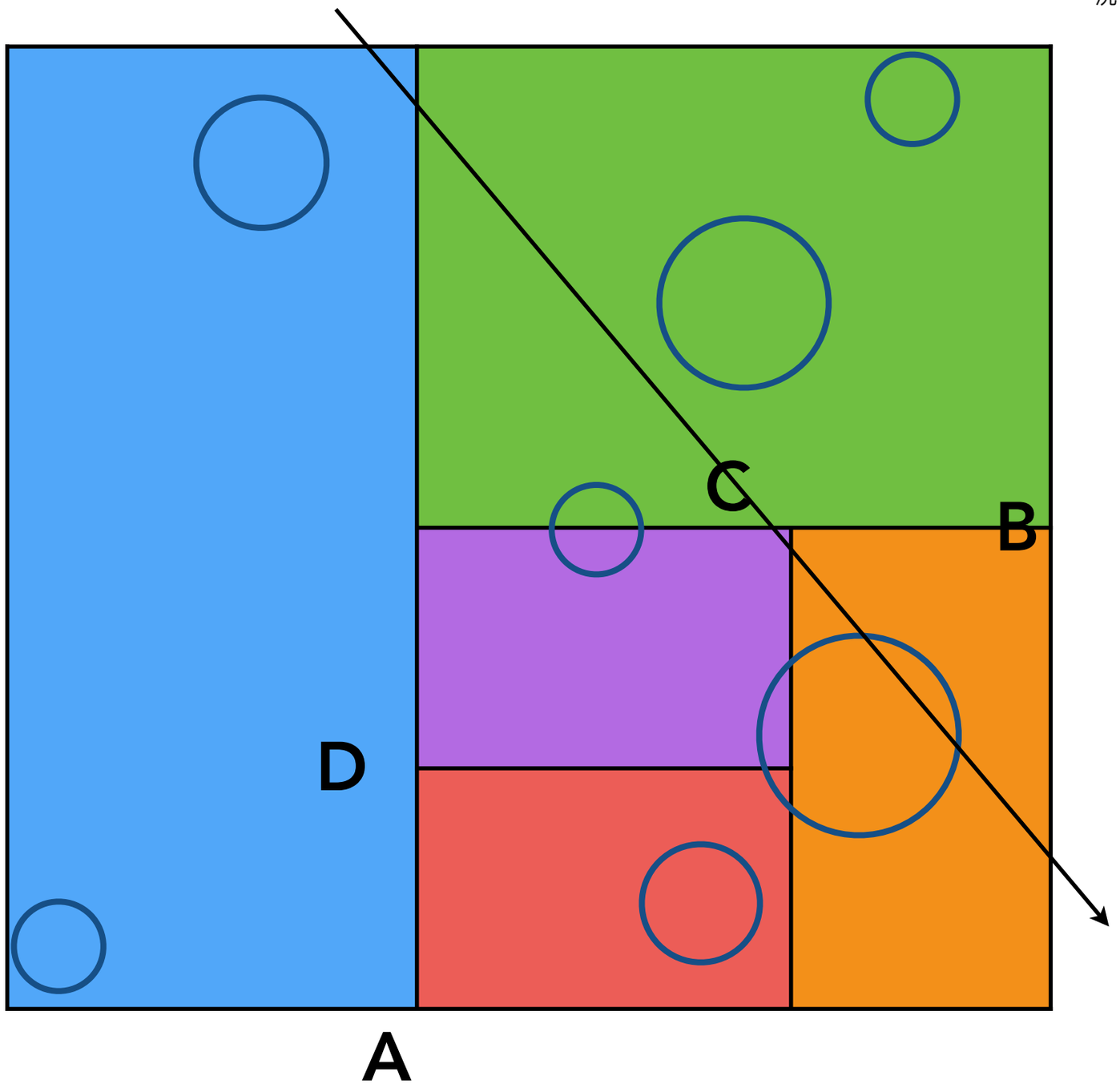
- split axis: x-, y-, or z-axis
- split position: coordinate of split plane along axis
- children: pointers to child nodes
- **No objects are stored in internal nodes**

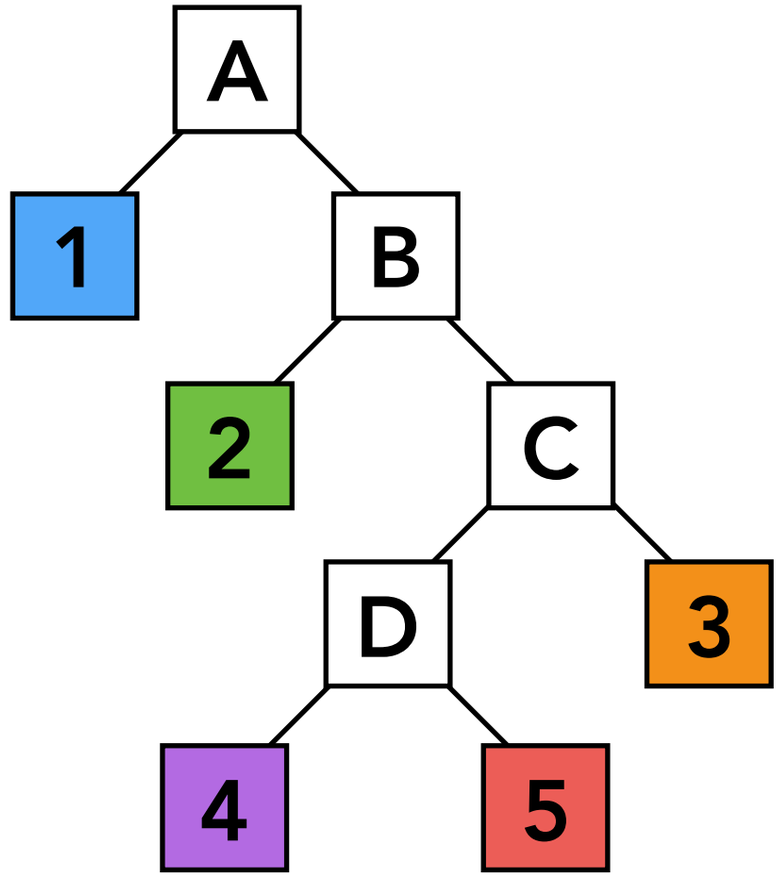
Leaf nodes store

- list of objects

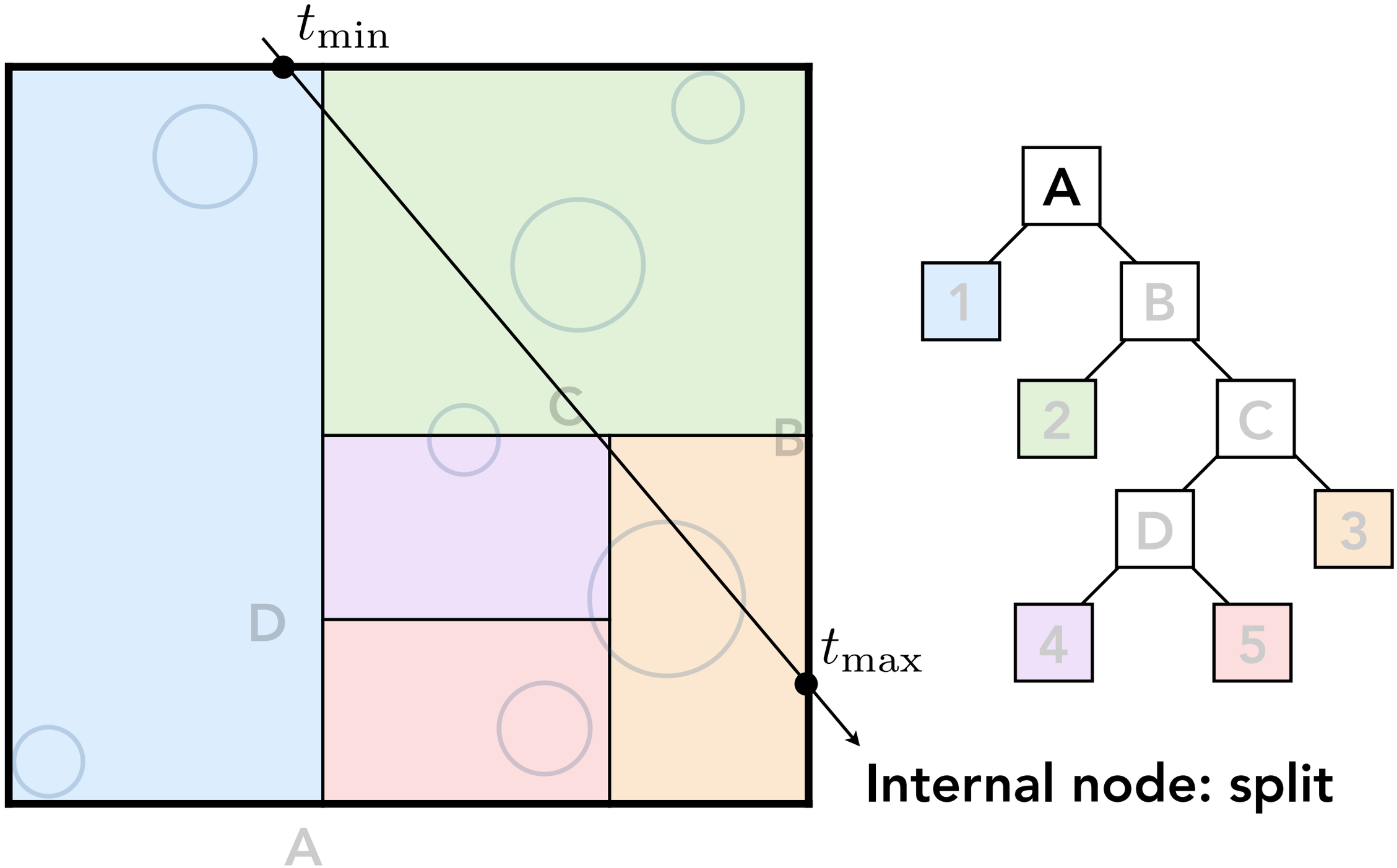
Traversing a KD-Tree

kd-tree 看着复杂，其实简而言之就是一个只在叶子节点存储物体信息，根节点只用来判断的树形结构。判断的方式也很树，比如图中的光线相交的判断方式：先计算最上层根节点（即最大包围盒）的相交情况，然后往下递归查找，直到找到了碰到物体的最小t为止。

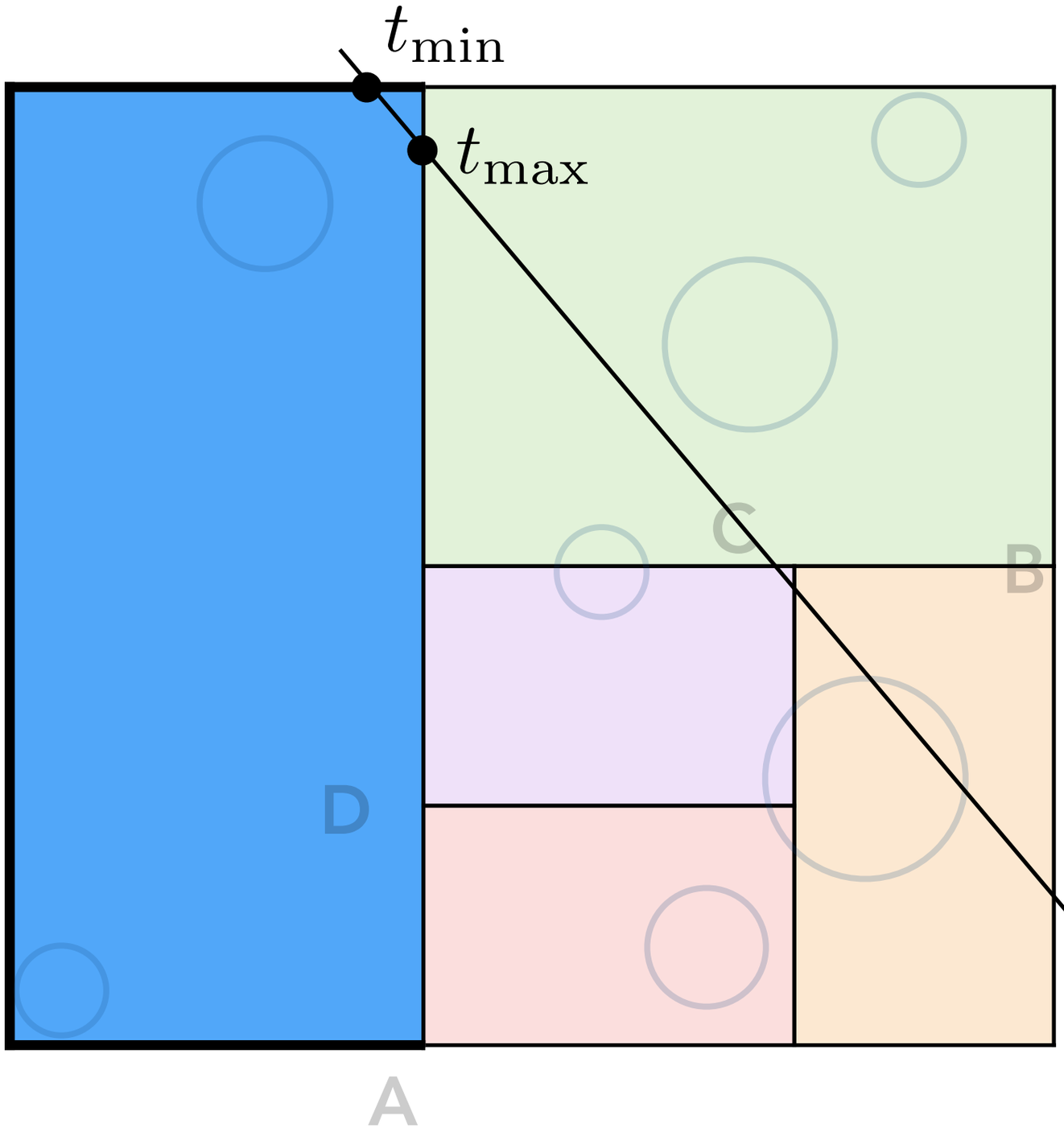


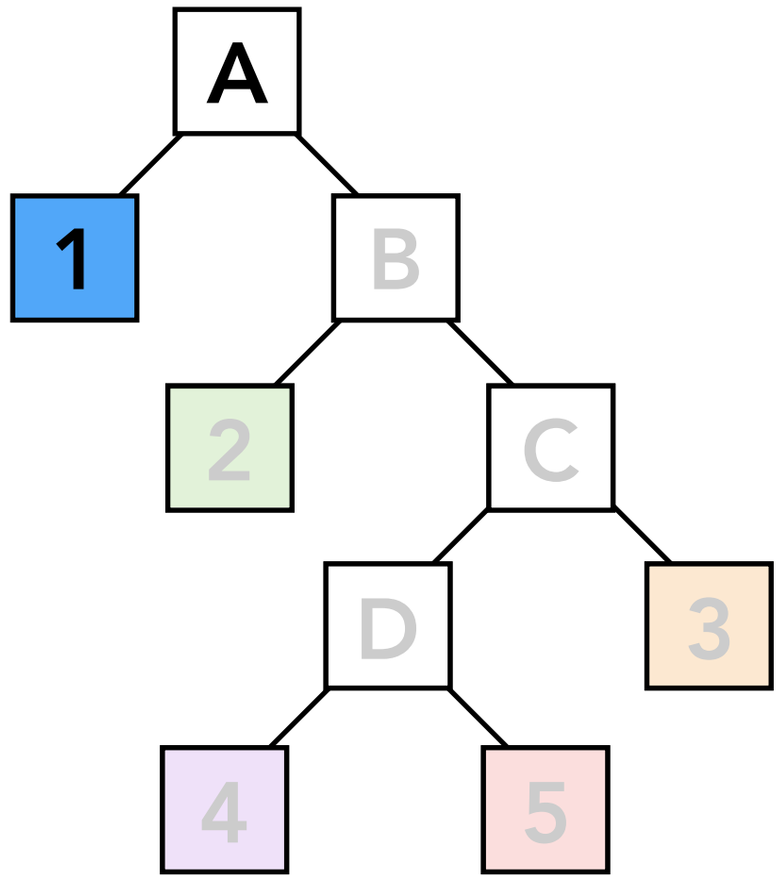


Traversing a KD-Tree



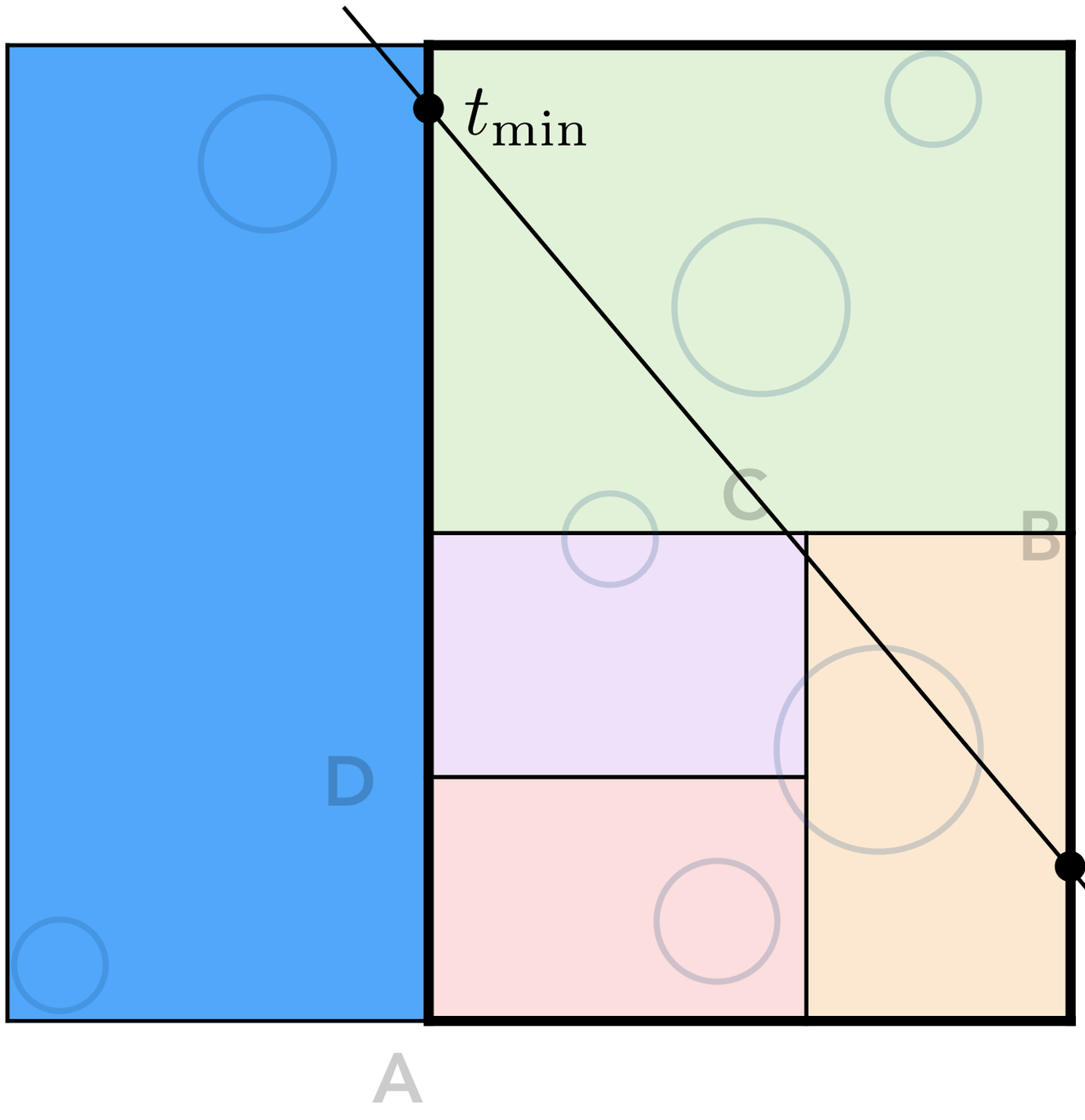
Traversing a KD-Tree

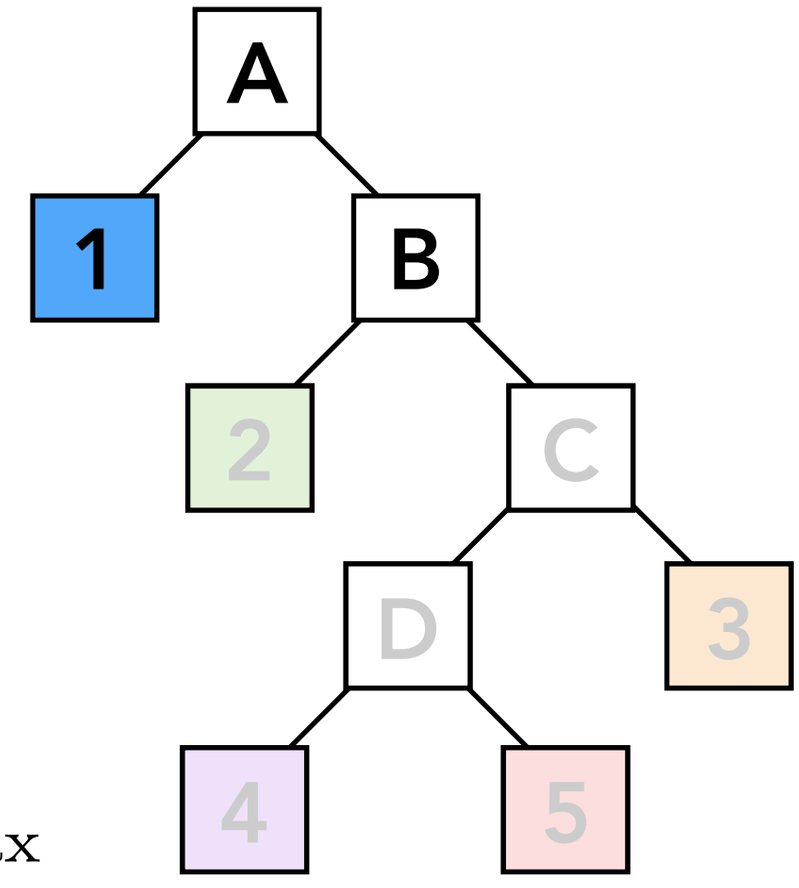




Assume it's leaf node: intersect all objects

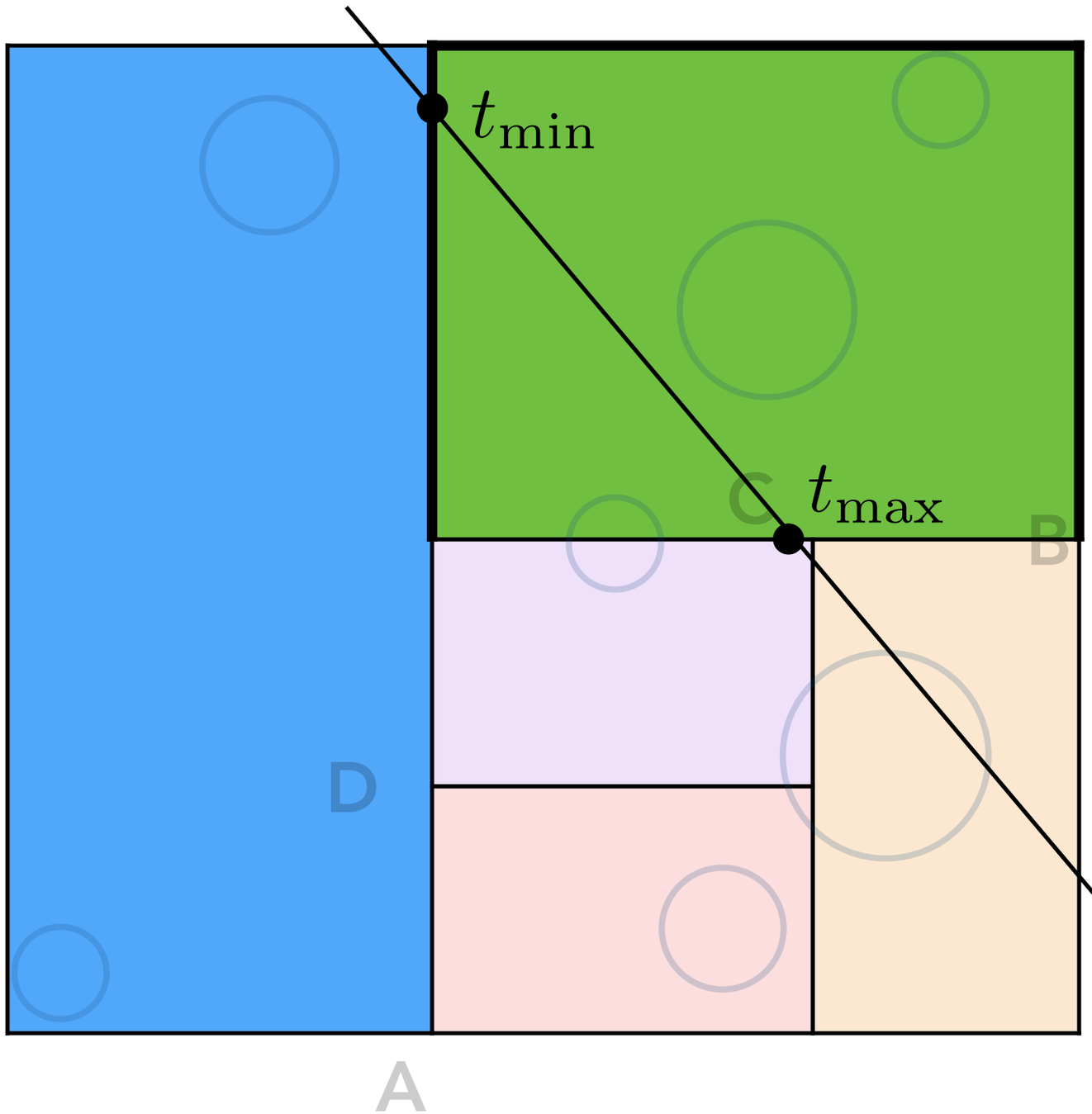
Traversing a KD-Tree

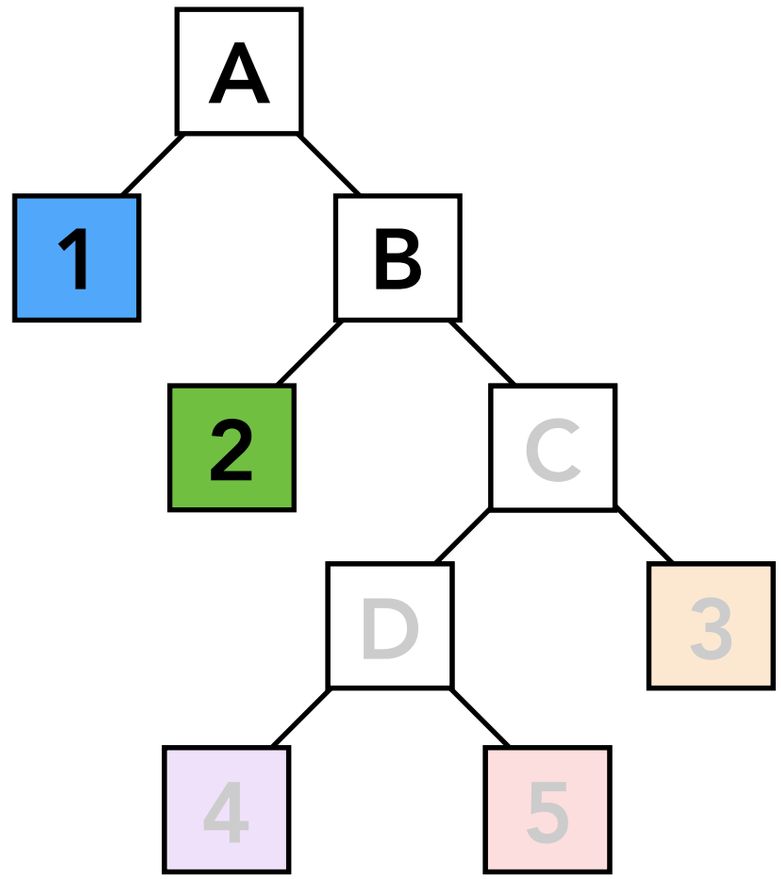




Internal node: split

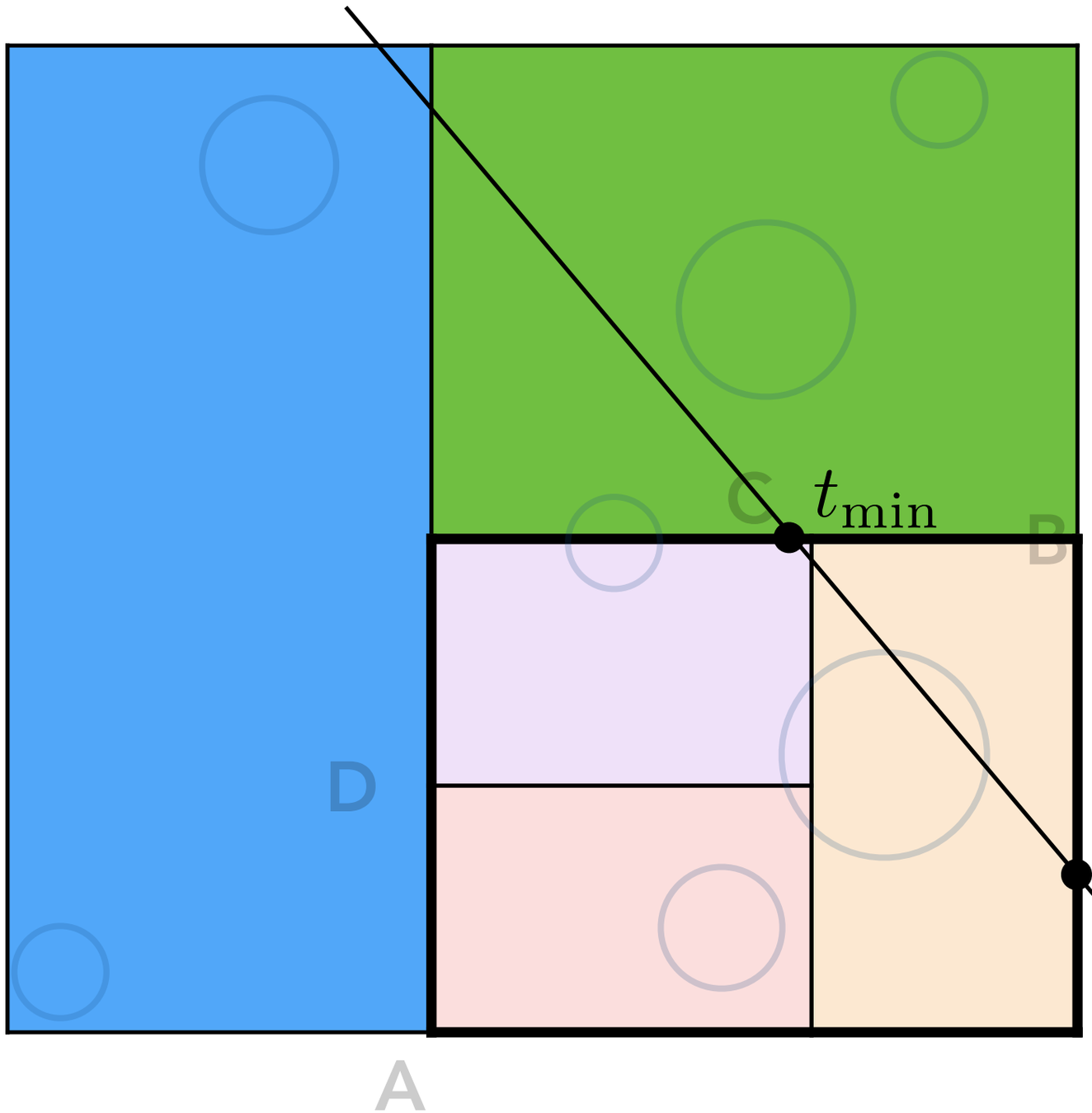
Traversing a KD-Tree

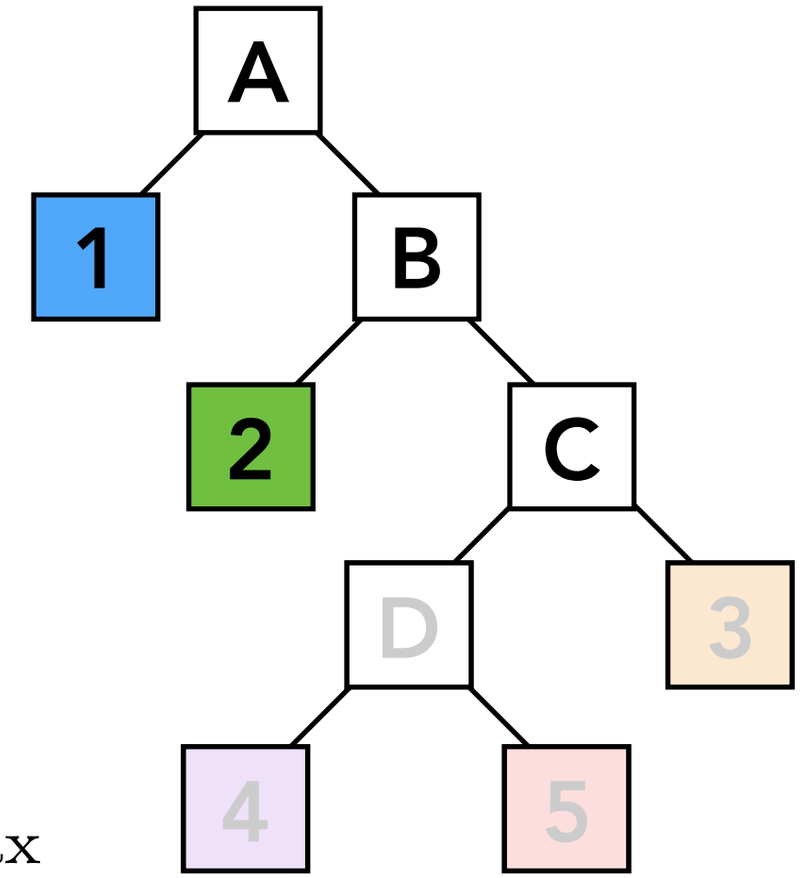




Leaf node: intersect all objects

Traversing a KD-Tree

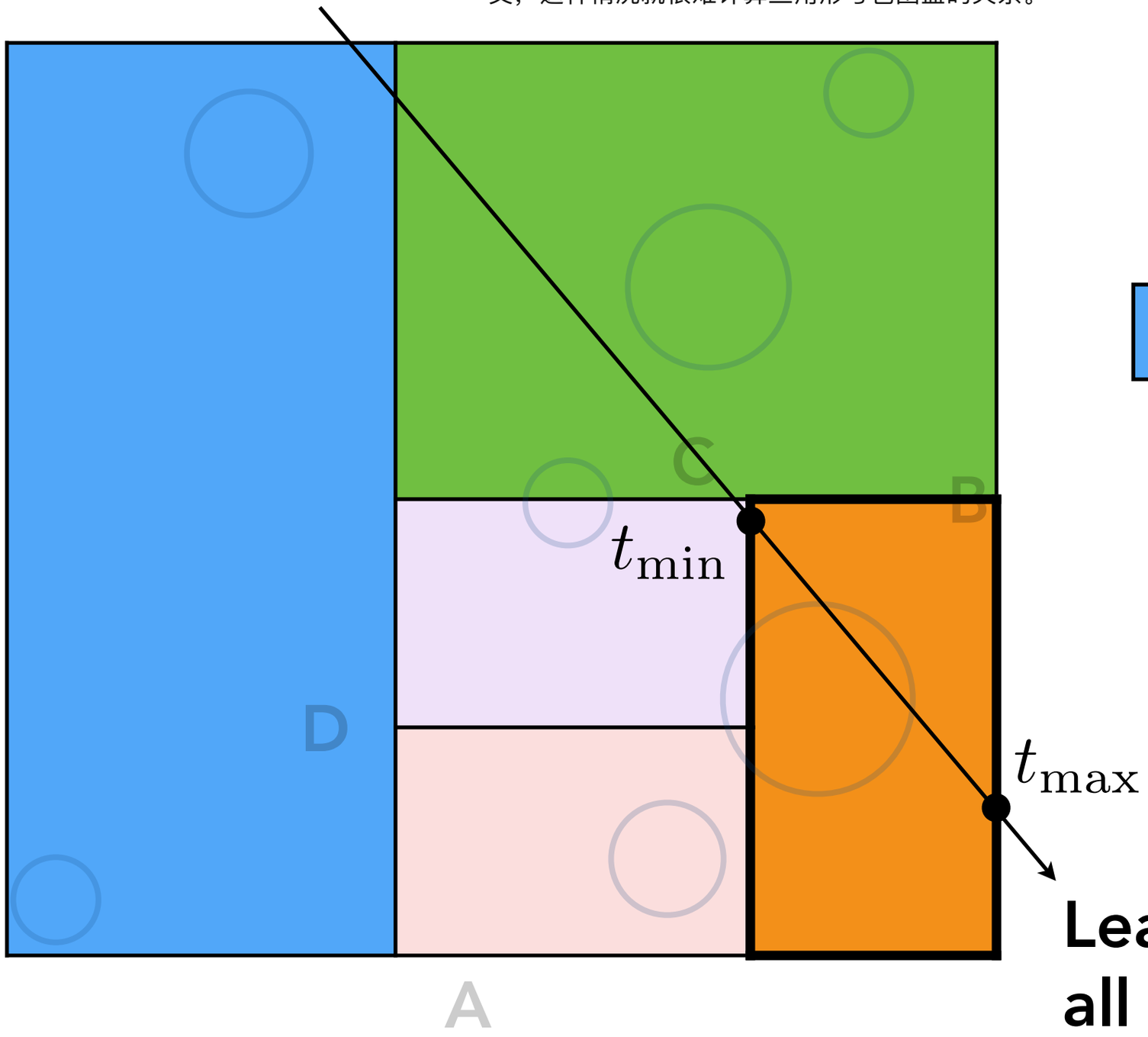


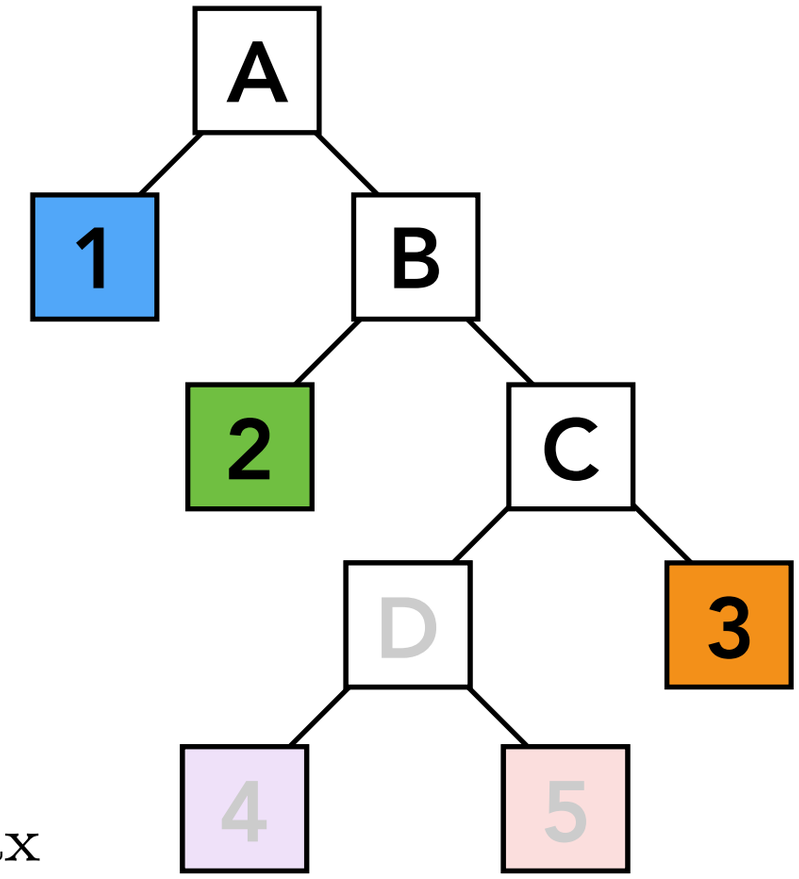


Internal node: split

Traversing a KD-Tree

这里简化了一些包围盒的计算，比如D子树的5节点，以及1子树将要往下划分的节点等。但有个致命问题：若一个物体同时与很多个包围盒相交怎么办？此时不仅要很多叶子节点来存储这个物体，并且KD-Tree的建立并不简单，还需要考虑物体（三角形）与包围盒的相交，比如一种情况是三角形的顶点全在包围盒之外但却有相交，这种情况就很难计算三角形与包围盒的关系。

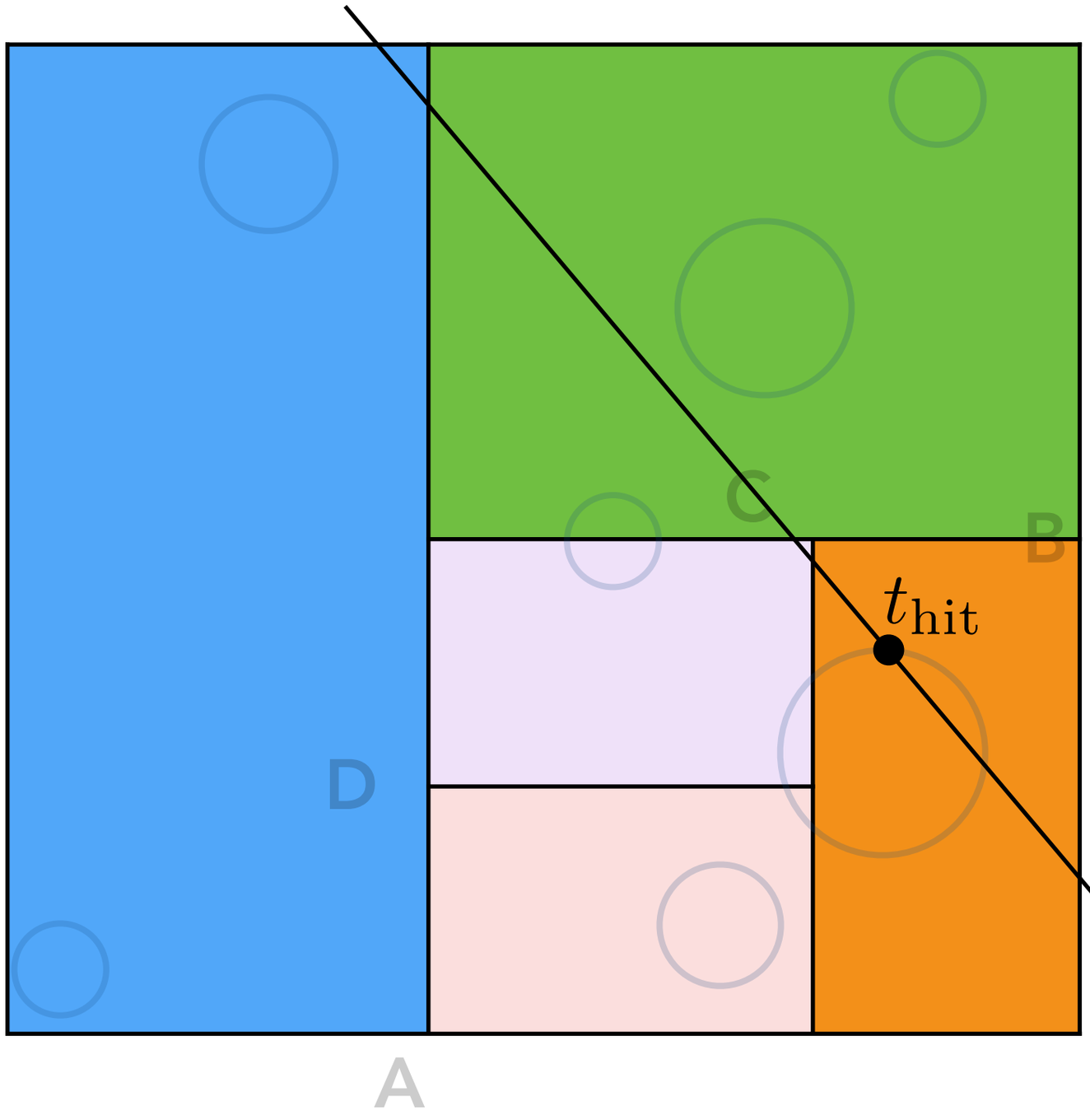


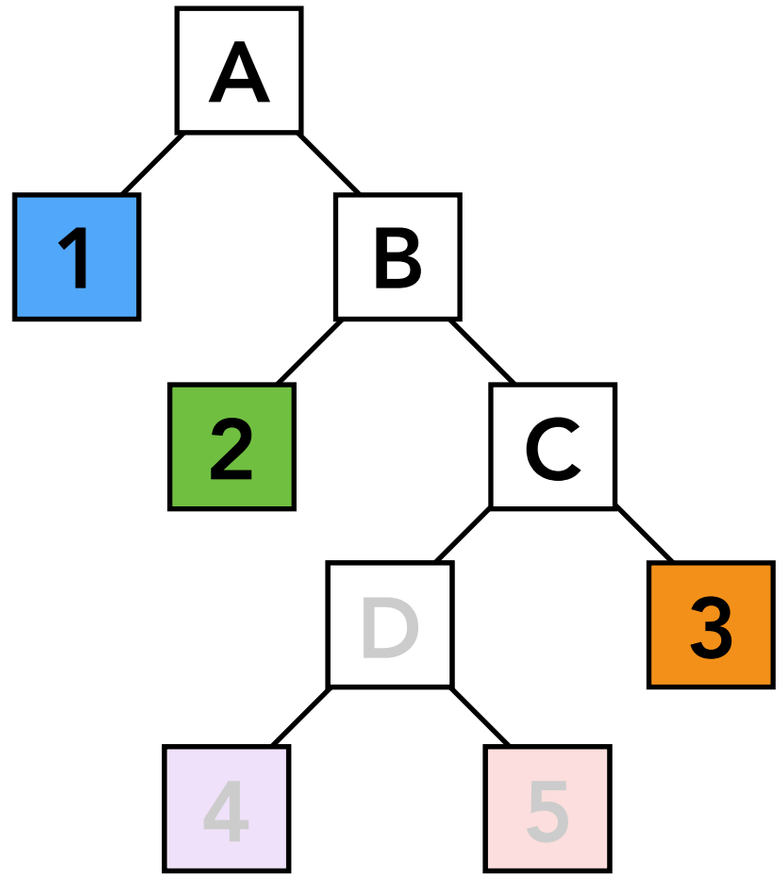


Leaf node: intersect all objects

Traversing a KD-Tree

这里不断递归，最后找到叶子节点3号中有物体碰撞，同理这里也会寻找到D子树的4节点去判断是否和物体有相交。右图树其实没有完整写出来，大概就是每个盒都会往下划分。





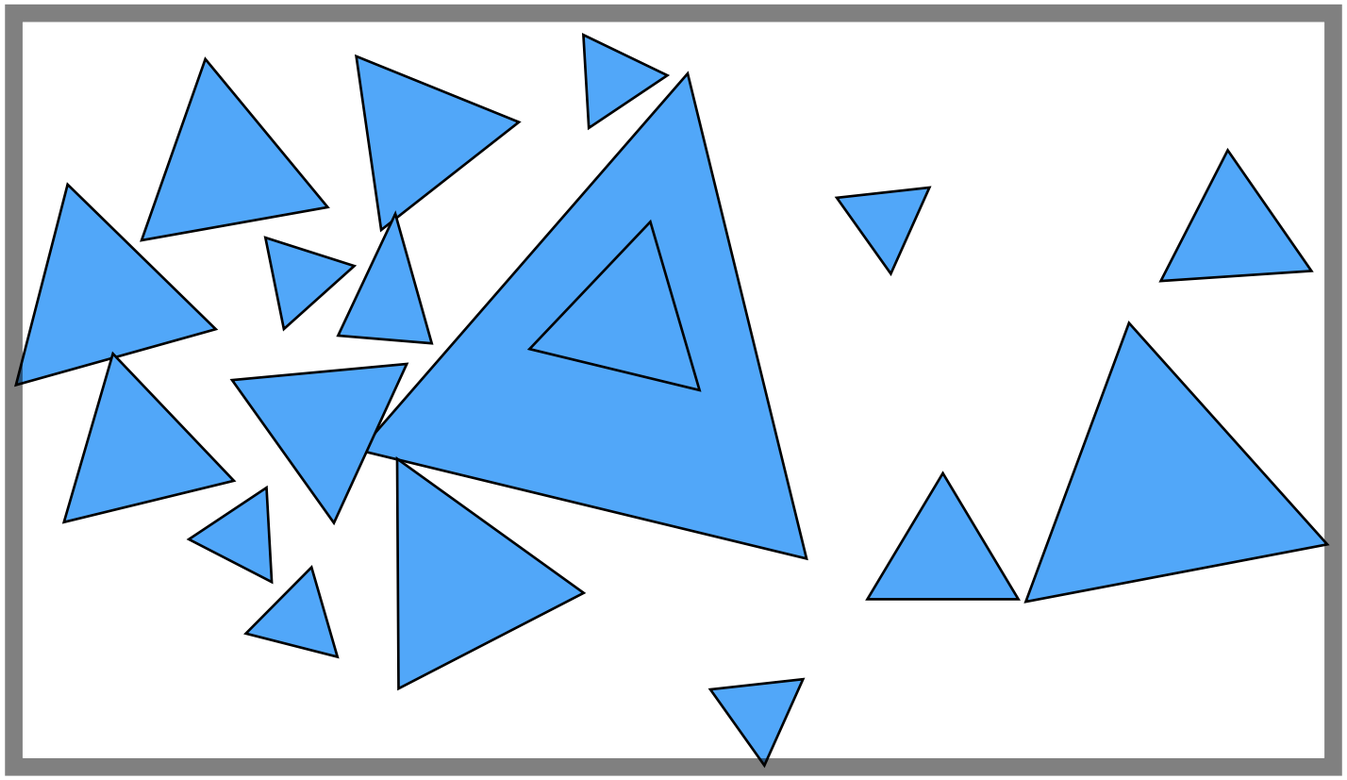
Intersection found

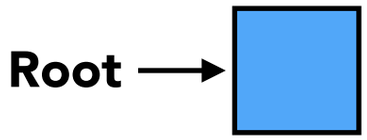
Object Partitions &

Bounding Volume Hierarchy (BVH)

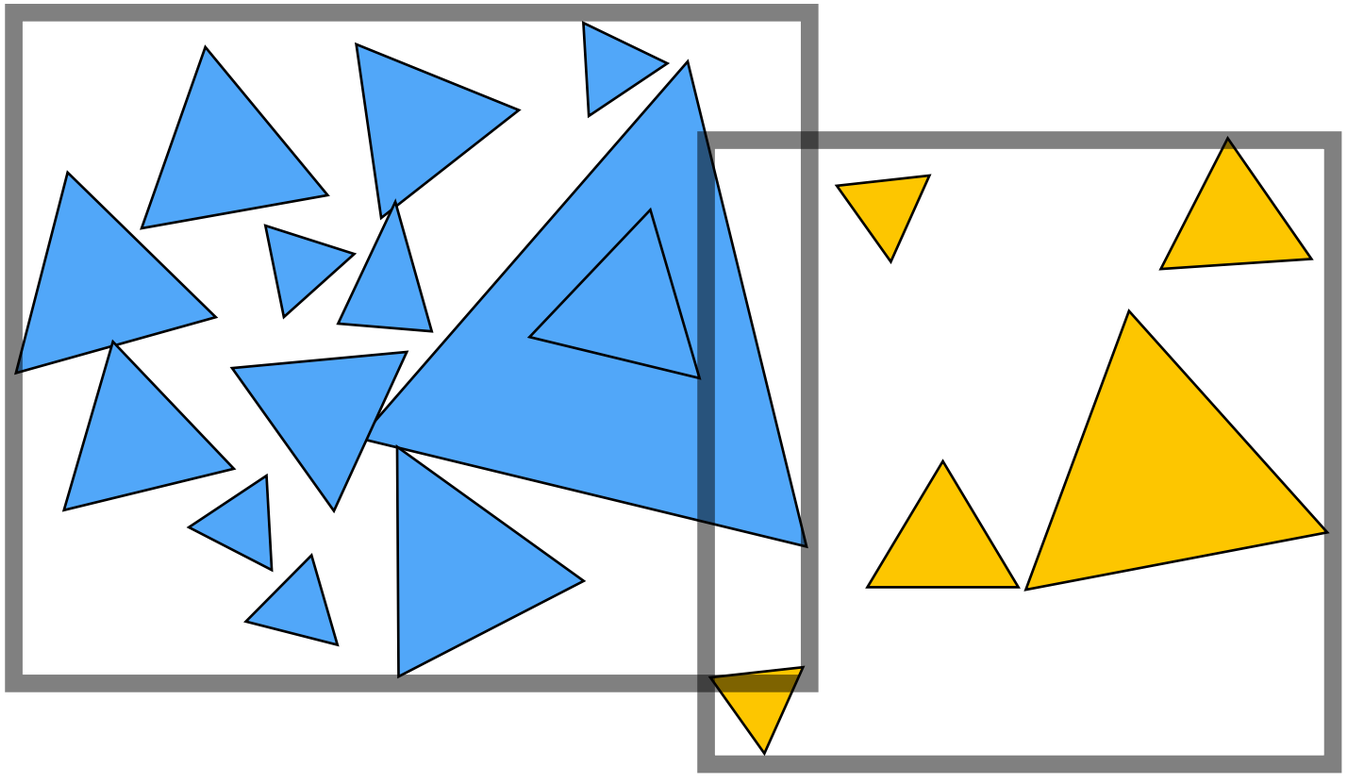
BVH的思路和KD树有着本质区别，BVH不是将空间划分为几个几个包围盒的形式，BVH是将空间中的物体划分为包围盒然后对于划分完的包围盒继续递归的划分下去直到一个包围盒里的物体（三角形）的数量较少为止。这就解决了KD树的物体与包围盒相交的问题，因为按照这样划分，一个物体最后只能出现在一个包围盒里

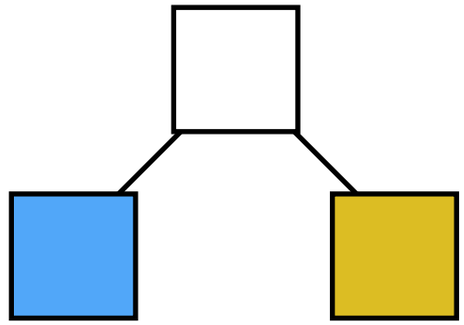
Bounding Volume Hierarchy (BVH)



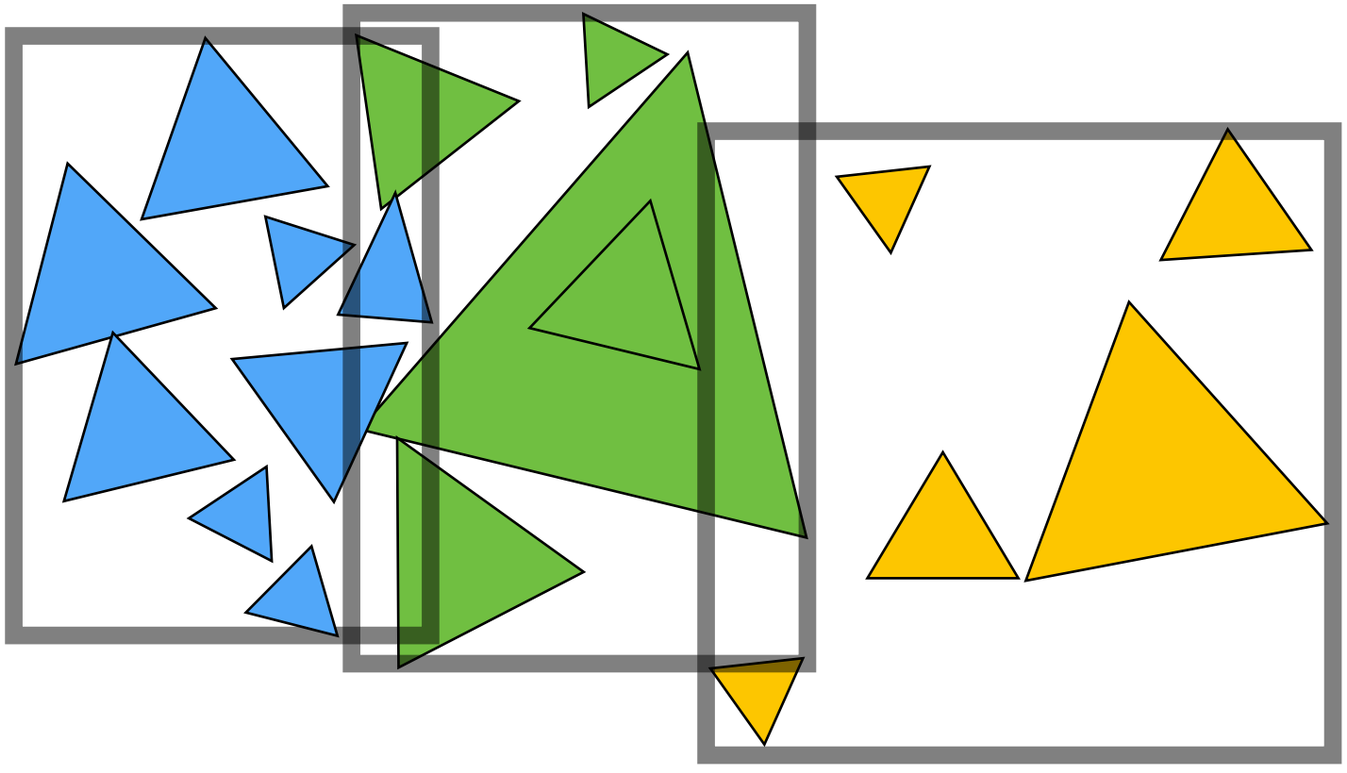


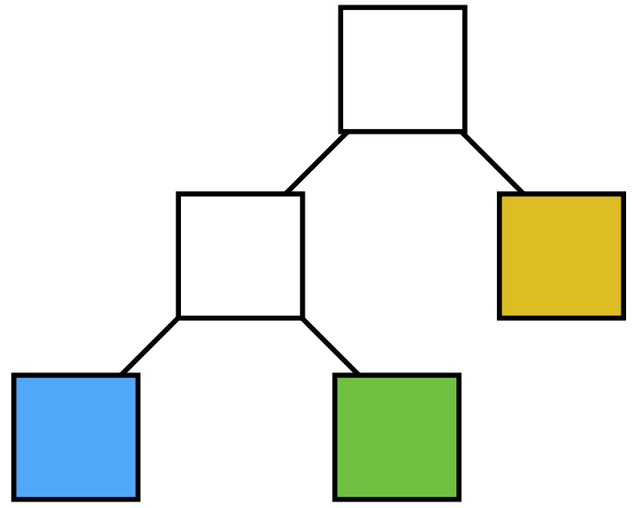
Bounding Volume Hierarchy (BVH)





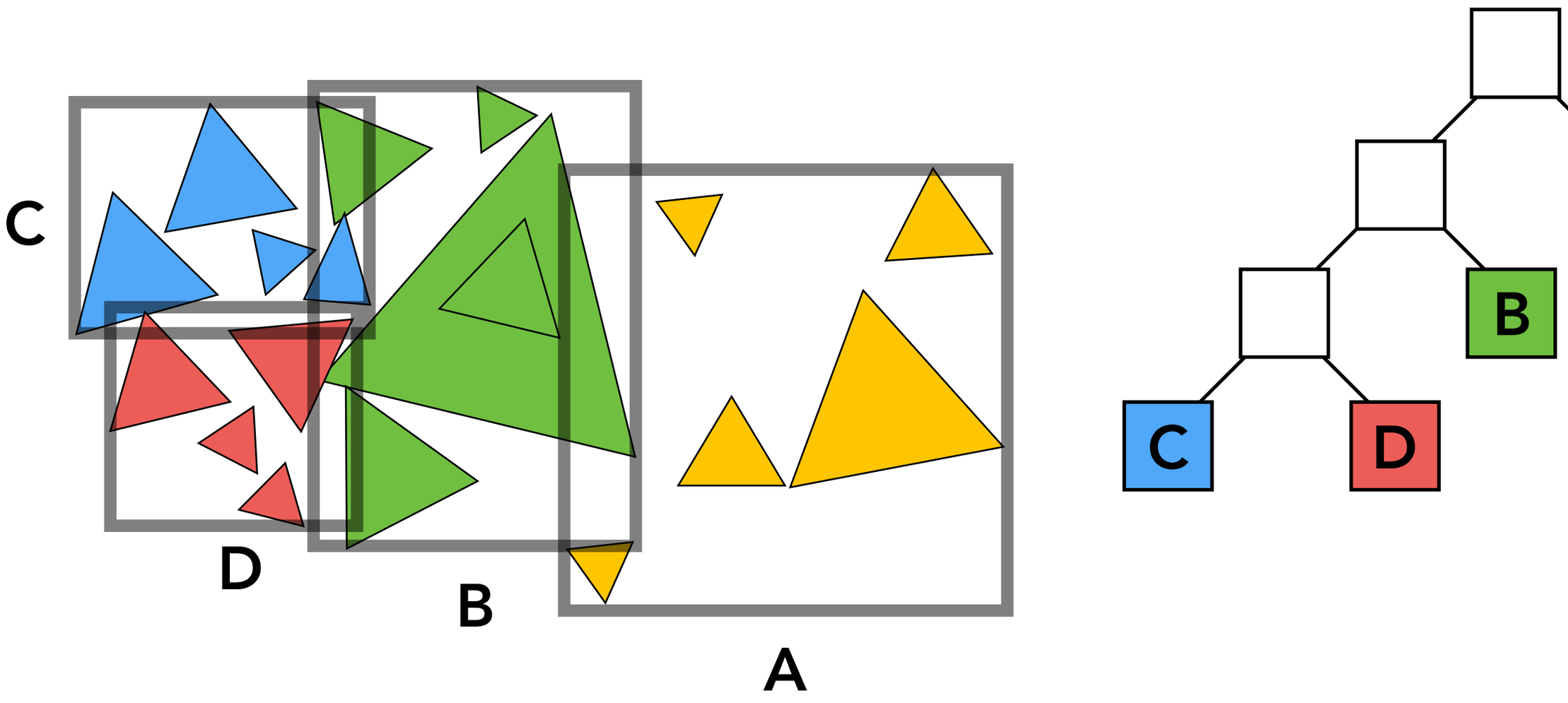
Bounding Volume Hierarchy (BVH)



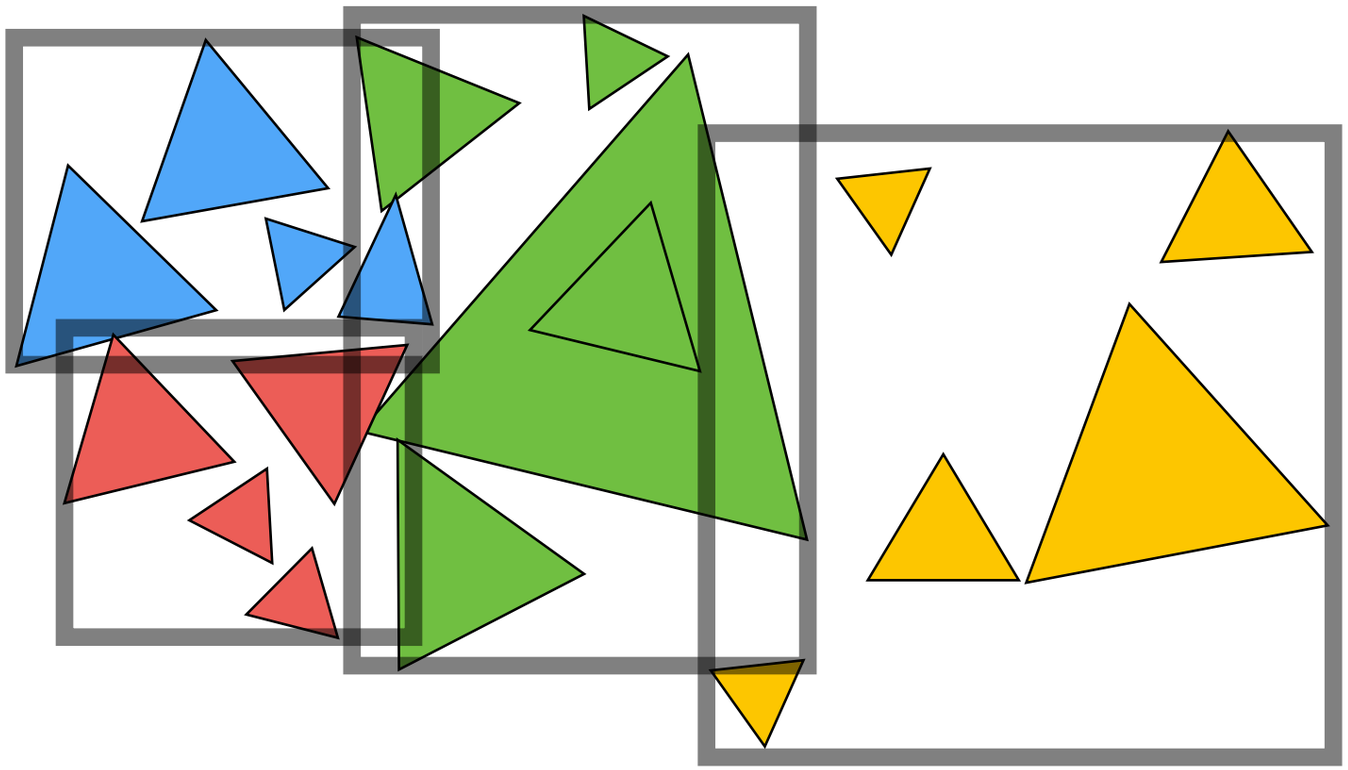


Bounding Volume Hierarchy (BVH)

BVH这种储存结构，中间结点储存包围盒和子节点的指针，叶子结点储存包围盒和物体（的集合）



Summary: Building BVHs



BVH 划分的基本步骤

- Find bounding box
- Recursively split set of objects in two subsets
- **Recompute** the bounding box of the subsets
- Stop when necessary
- Store objects in each leaf node

找到场景包围盒 –> 每次将物体分为两堆 –> 对两堆物体重新计算包围盒 –> 直到一堆中物体少到一定程度

Building BVHs

How to subdivide a node? 空间划分方式

- Choose a dimension to split
- Heuristic #1: Always choose the longest axis in node
- Heuristic #2: Split node at location of **median** object

这里有两个Heuristic（关于这个词我印象深刻，在上深度学习课程的时候老师特别提过，意思就是一拍脑袋的想法）第一是永远沿着最长的轴划分。第二是取三角形数量的中位数的那个三角形的位置来划分，这样能保证树形结构平衡

Termination criteria? 递归终止条件

- Heuristic: stop when node contains few elements (e.g. 5)

Data Structure for BVHs

Internal nodes store

- Bounding box
- Children: pointers to child nodes

Leaf nodes store

- Bounding box
- List of objects

Nodes represent subset of primitives in scene

- All objects in subtree

BVH Traversal

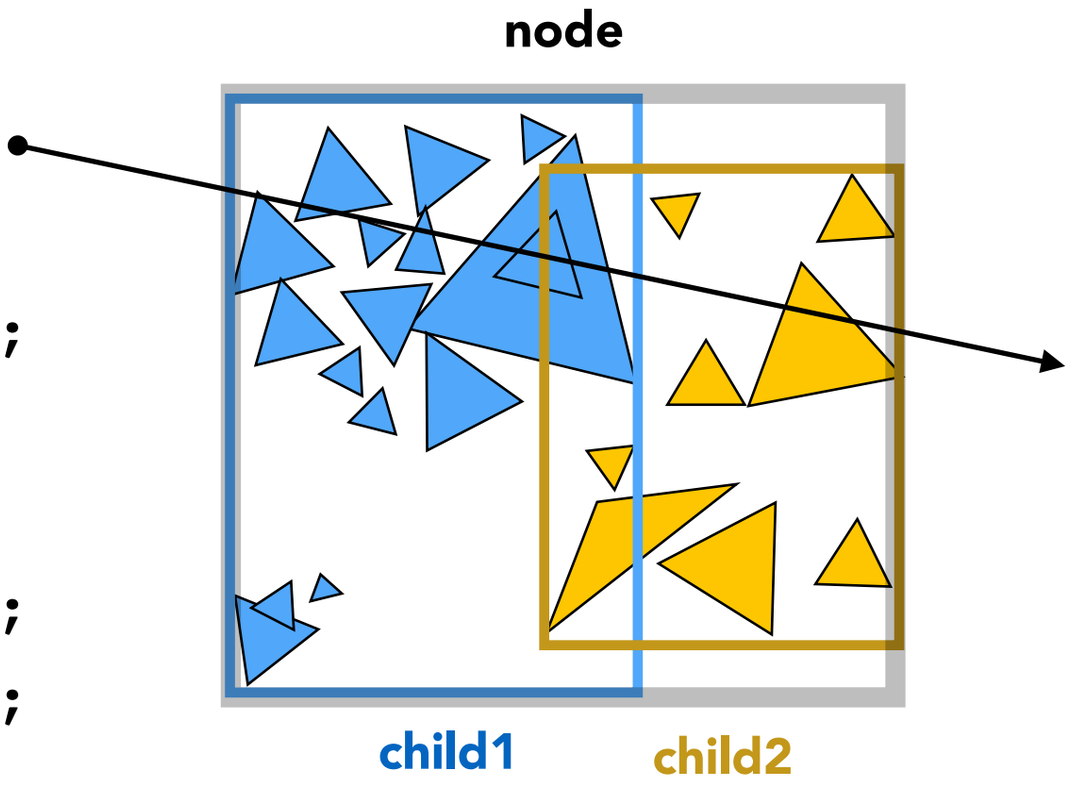
BVH的伪代码思路，就是先判断光线是否进入包围盒，若是就继续判断光线进入的包围盒是否是叶子节点，若是就与盒内物体都求交点然后返回最近的那个交点，若不是叶子节点，那就递归访问这个根节点的子节点。

```
Intersect(Ray ray, BVH node) {
  if (ray misses node.bbox) return;

  if (node is a leaf node)
     test intersection with all objs;
     return closest intersection;

  hit1 = Intersect(ray, node.child1);
  hit2 = Intersect(ray, node.child2);

  return the closer of hit1, hit2;
}
```

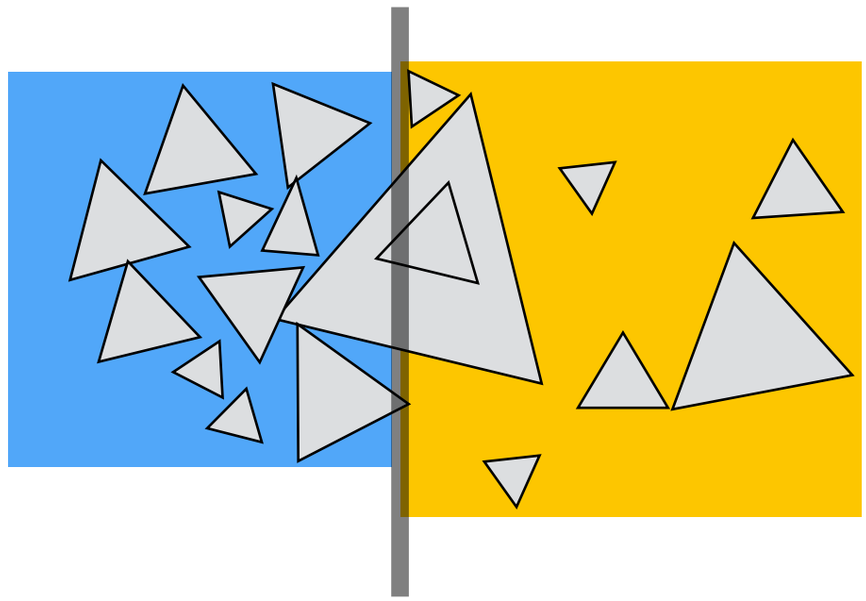


Spatial vs Object Partitions

一个是对空间的划分，一个是基于物体的划分；KD树的包围盒不会发生重合，而BVHs会发生相交

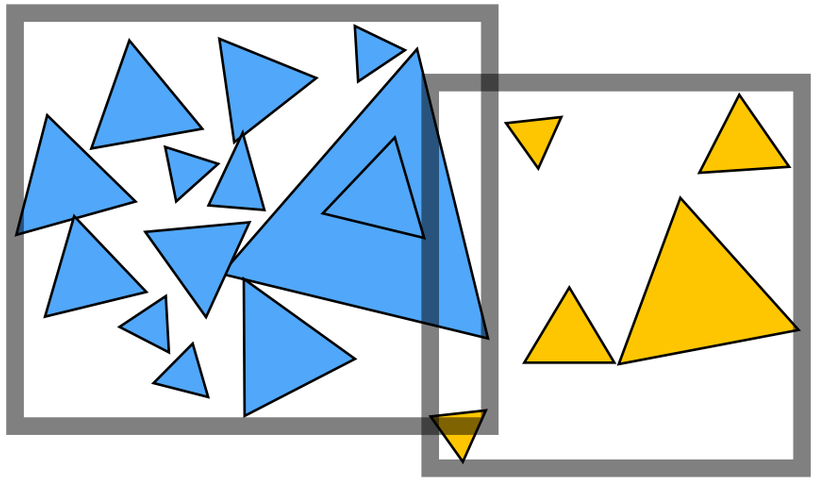
Spatial partition (e.g.KD-tree)

- Partition space into non-overlapping regions
- An object can be contained in multiple regions



Object partition (e.g. BVH)

- Partition set of objects into disjoint subsets
- Bounding boxes for each set may overlap in space



Today

- Using AABBs to accelerate ray tracing
  - Uniform grids
  - Spatial partitions

- Basic radiometry (辐射度量学)
  - Advertisement: new topics from now on, scarcely covered in other graphics courses

Radiometry — Motivation

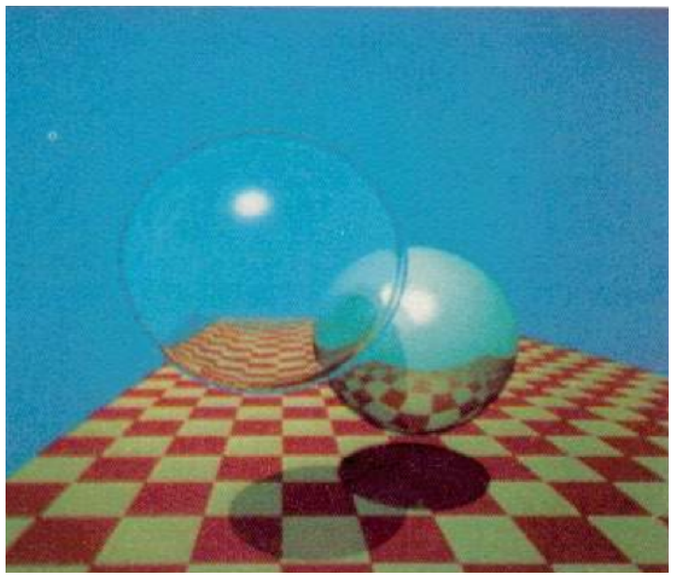
Observation

- In assignment 3, we implement the Blinn-Phong model
- Light intensity I is 10, for example
- But 10 what?

Do you think Whitted style ray tracing gives you CORRECT results?

All the answers can be found in radiometry

- Also the basics of “Path Tracing”



Radiometry

辐射度量学（Radiometry）就是用来定义光的一系列物理量的方法，即在物理上精确的定义光照。

Measurement system and units for illumination

Accurately measure the spatial properties of light

- New terms: Radiant flux, intensity, irradiance, radiance

Perform lighting calculations **in a physically correct manner**

My personal way of learning things:

- WHY, WHAT, then HOW

Radiant Energy and Flux (Power)

先从Photons（光子）来理解光线，所谓光线可看做一系列光子所传播而形成的，我们都知道光是有能量的，并且光的在介质中的传播速度取决于介质的折射率，折射率越大光速越小，所以光子的传播速度也取决于介质。

Radiant Energy and Flux (Power)

Definition: Radiant energy is the energy of electromagnetic radiation. It is measured in units of joules, and denoted by the symbol:

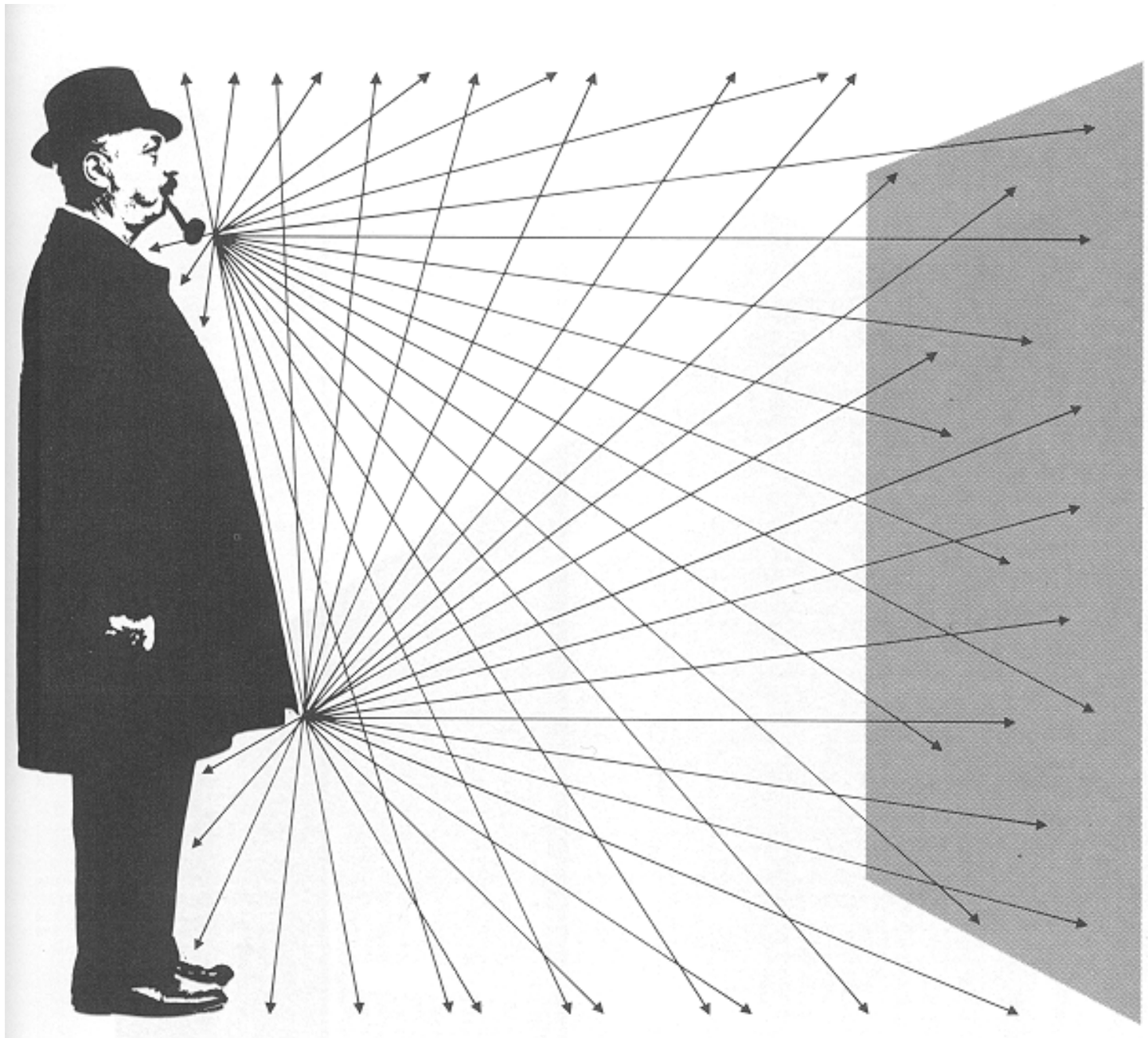
Radiant Energy 就是这一系列光子能量q的总和

$$Q \ [\mathrm{J} = \mathrm{Joule}]$$

Definition: Radiant flux (power) is the energy emitted, reflected, transmitted or received, per unit time.

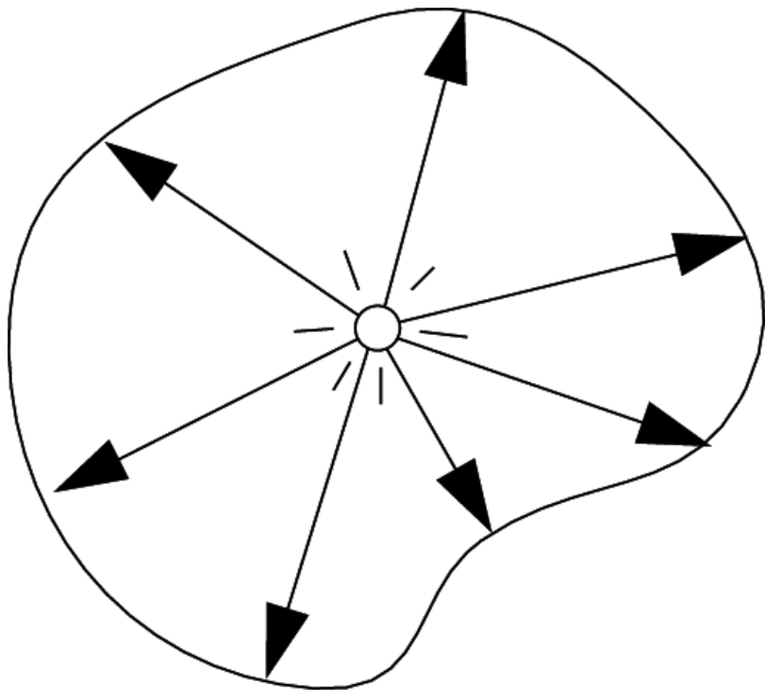
$$\Phi \equiv \frac{\mathrm{d}Q}{\mathrm{d}t} \ [\mathrm{W} = \mathrm{Watt}] \ [\mathrm{lm} = \mathrm{lumen}]^*$$

Flux – #photons flowing through a sensor in unit time



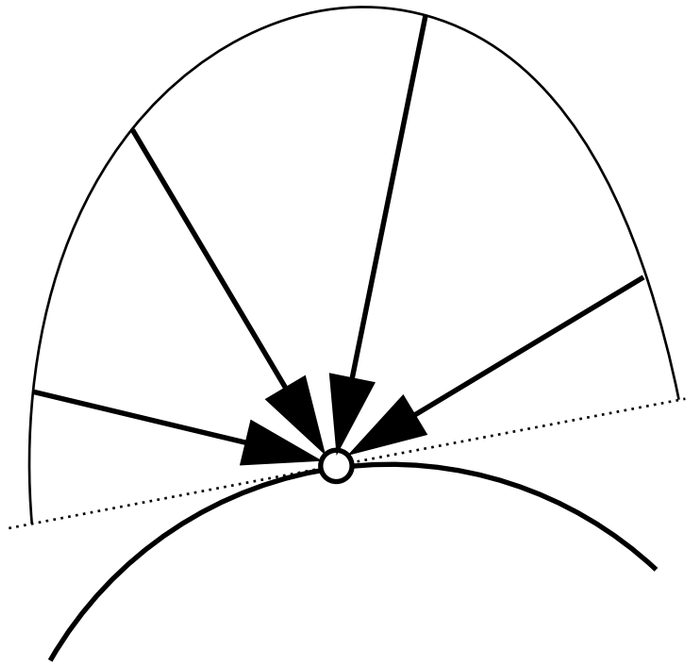
From London and Upton

Important Light Measurements of Interest



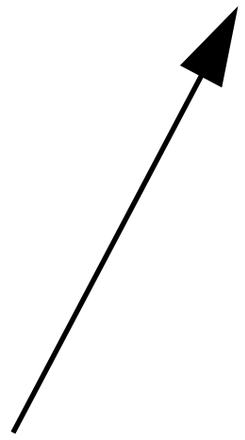
Light Emitted From A Source

"Radiant Intensity"



Light Falling On A Surface

"Irradiance"



Light Traveling Along A Ray

"Radiance"

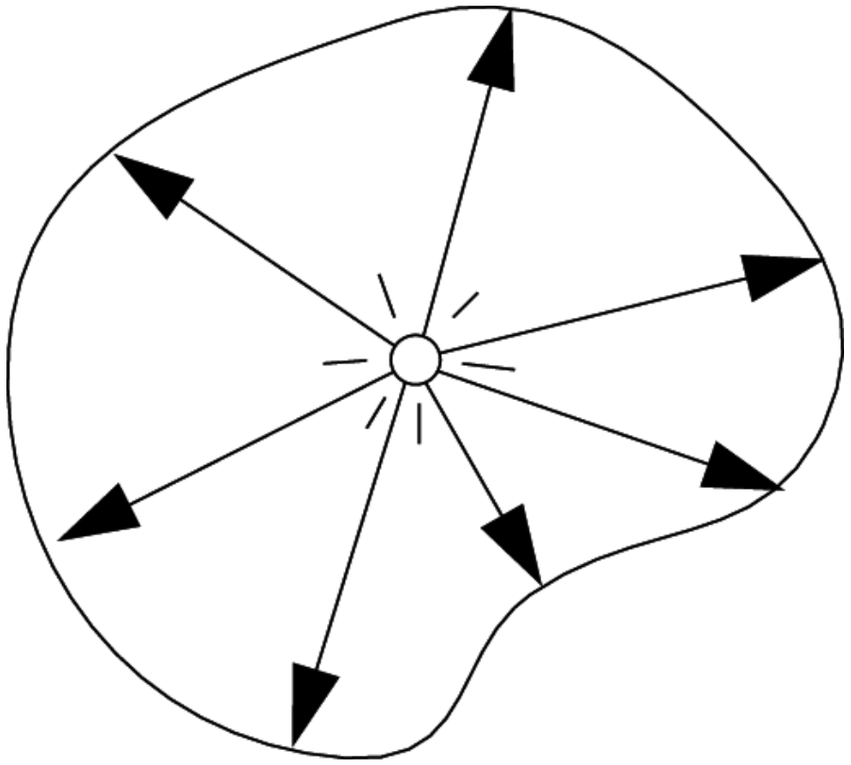
Radiant Intensity

Radiant Intensity

单位立体角上的Power

Definition: The radiant (luminous) intensity is the power per unit **solid angle** **(?)** emitted by a point light source.

(立体角)



$$I(\omega) \equiv \frac{\mathrm{d}\Phi}{\mathrm{d}\omega}$$

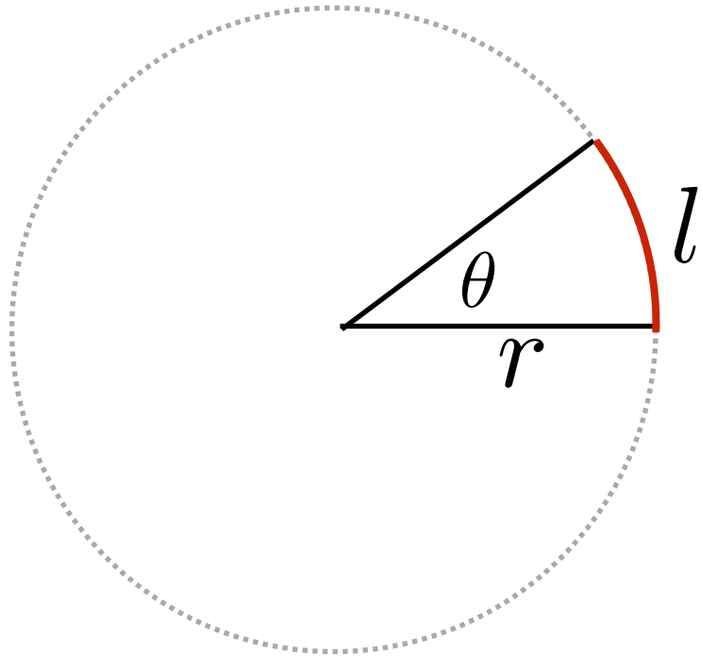
$$\left[\frac{\mathrm{W}}{\mathrm{sr}}\right] \left[\frac{\mathrm{lm}}{\mathrm{sr}} = \mathrm{cd} = \mathrm{candela}\right]$$

The candela is one of the seven SI base units.

Angles and Solid Angles

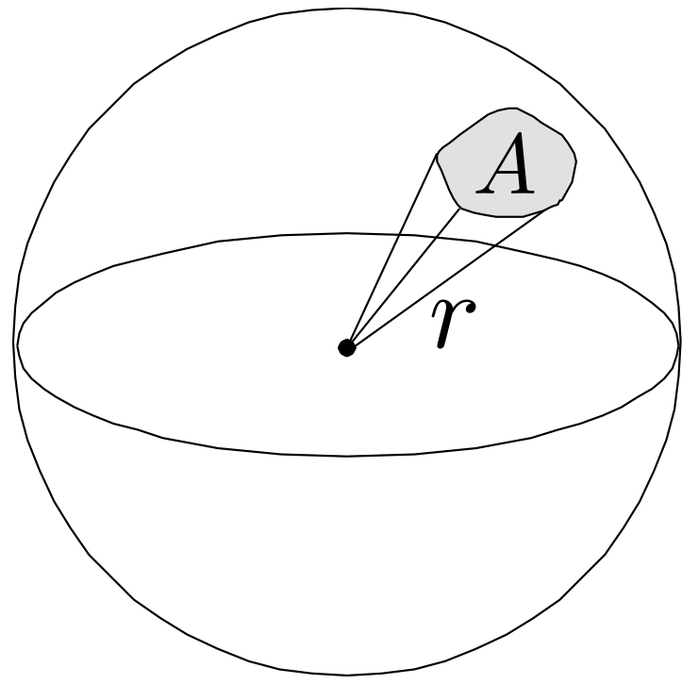
Angle: ratio of subtended arc length on circle to radius

- $\theta = \dfrac{l}{r}$
- **Circle has 2π radians**

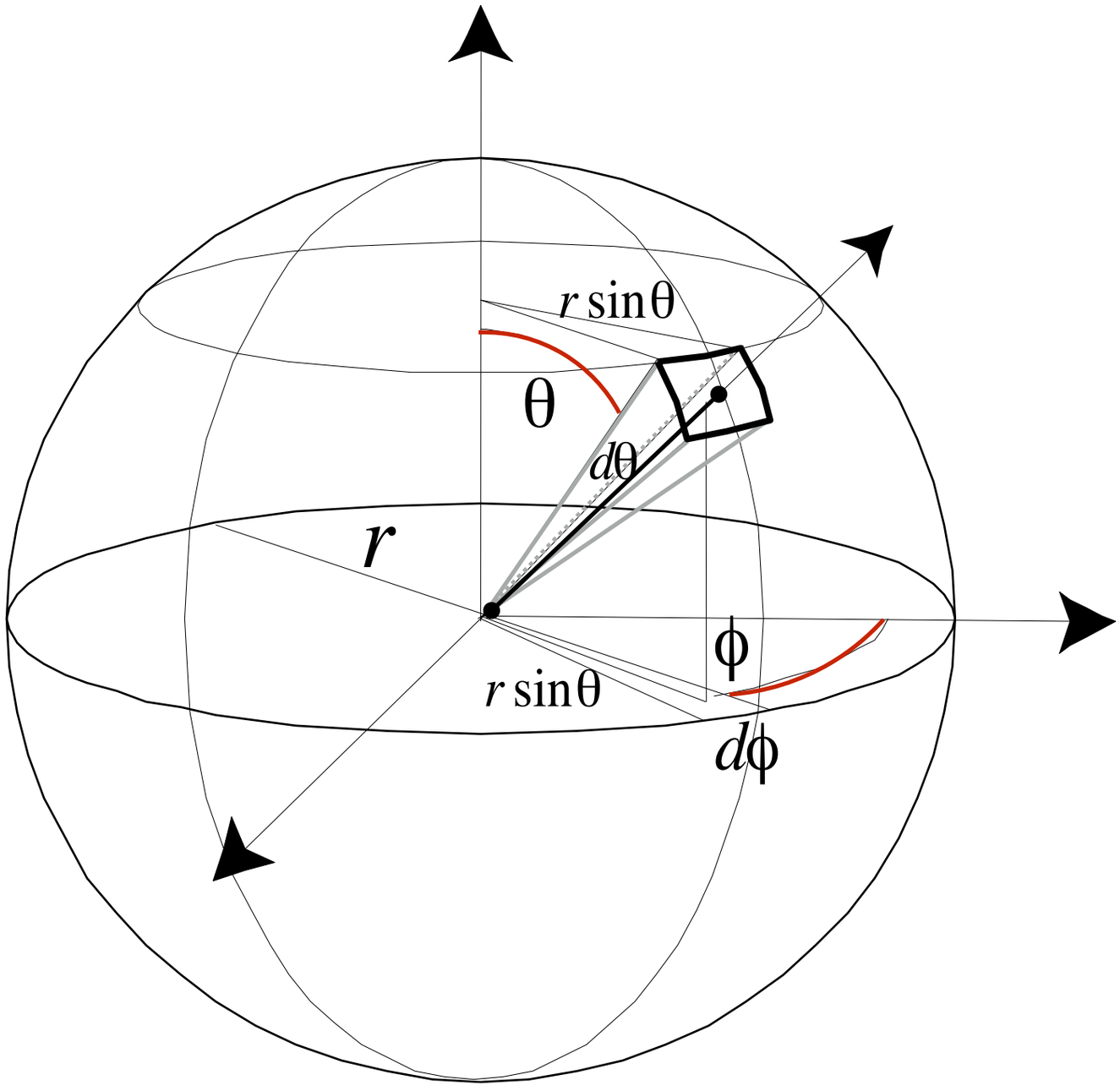


Solid angle: ratio of subtended area on sphere to radius squared

- $\Omega = \dfrac{A}{r^2}$
- **Sphere has 4π steradians**



Differential Solid Angles

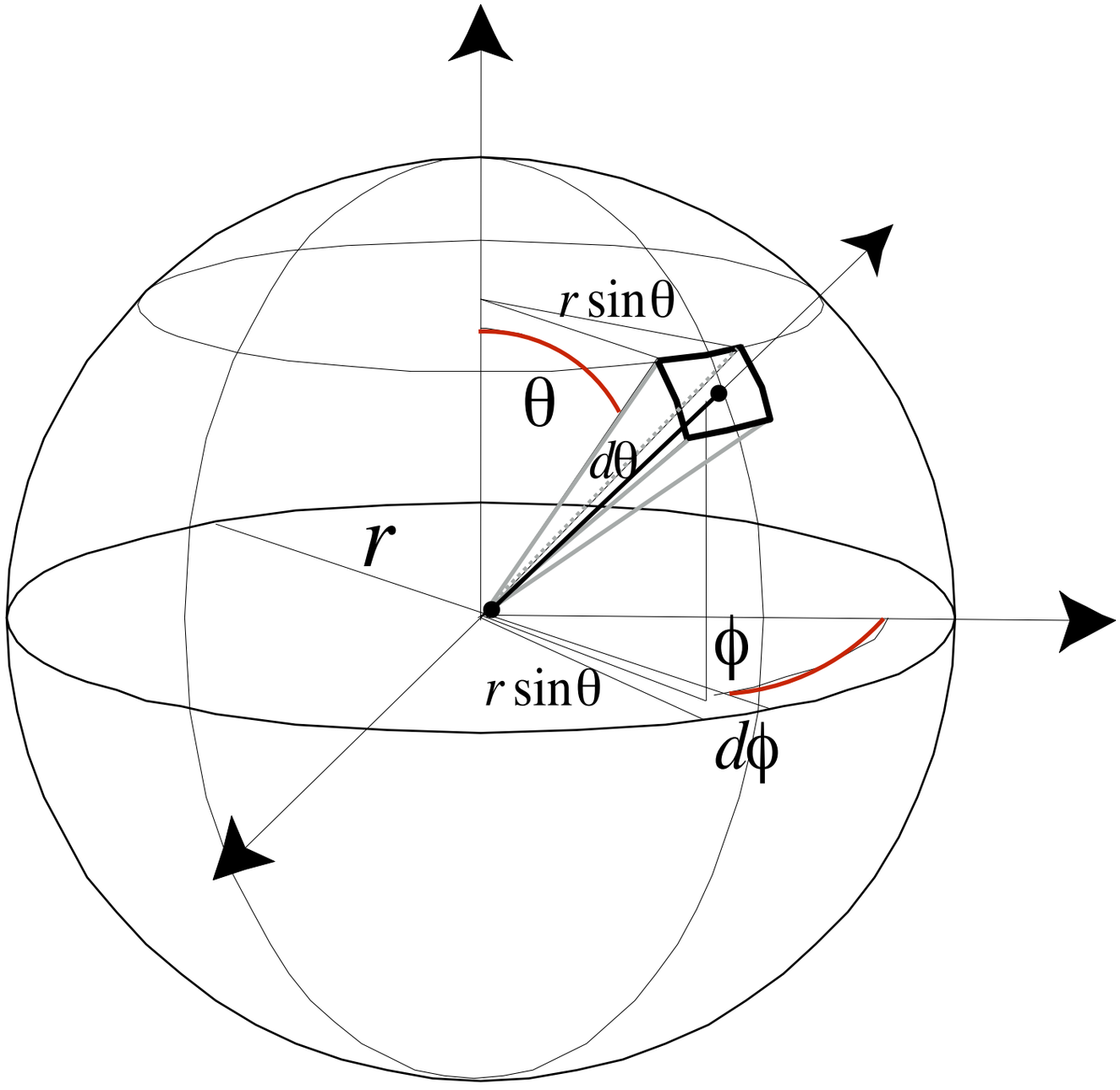


$$\mathrm{d}A = (r\,\mathrm{d}\theta)(r\sin\theta\,\mathrm{d}\phi)$$

$$= r^2\sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi$$

$$\mathrm{d}\omega = \frac{\mathrm{d}A}{r^2} = \sin\theta\,\mathrm{d}\theta\,\mathrm{d}\phi$$

Differential Solid Angles



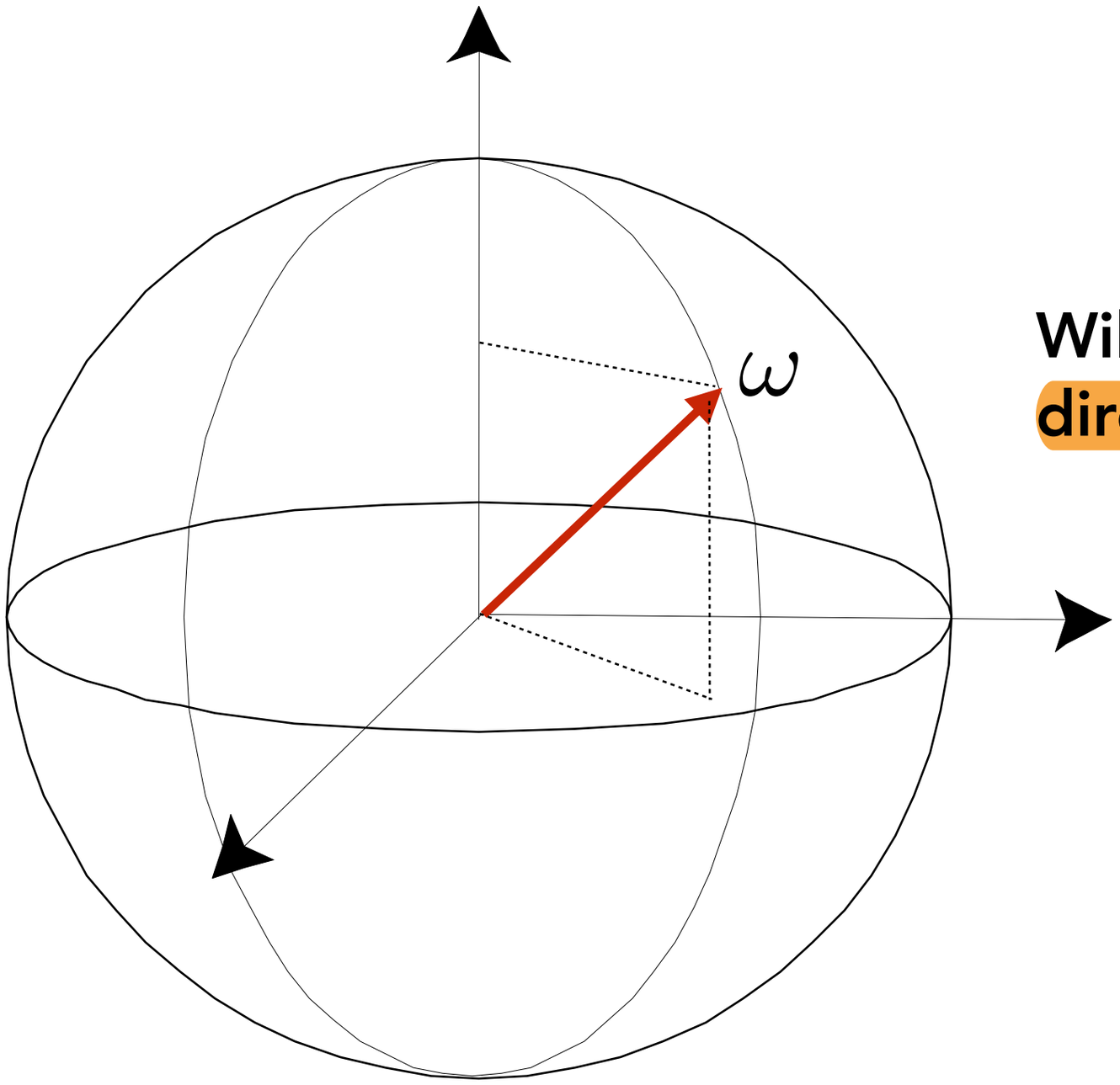
Sphere: S^2

$$\Omega = \int_{S^2} \mathrm{d}\omega$$

$$= \int_0^{2\pi} \int_0^{\pi} \sin\theta \, \mathrm{d}\theta \, \mathrm{d}\phi$$

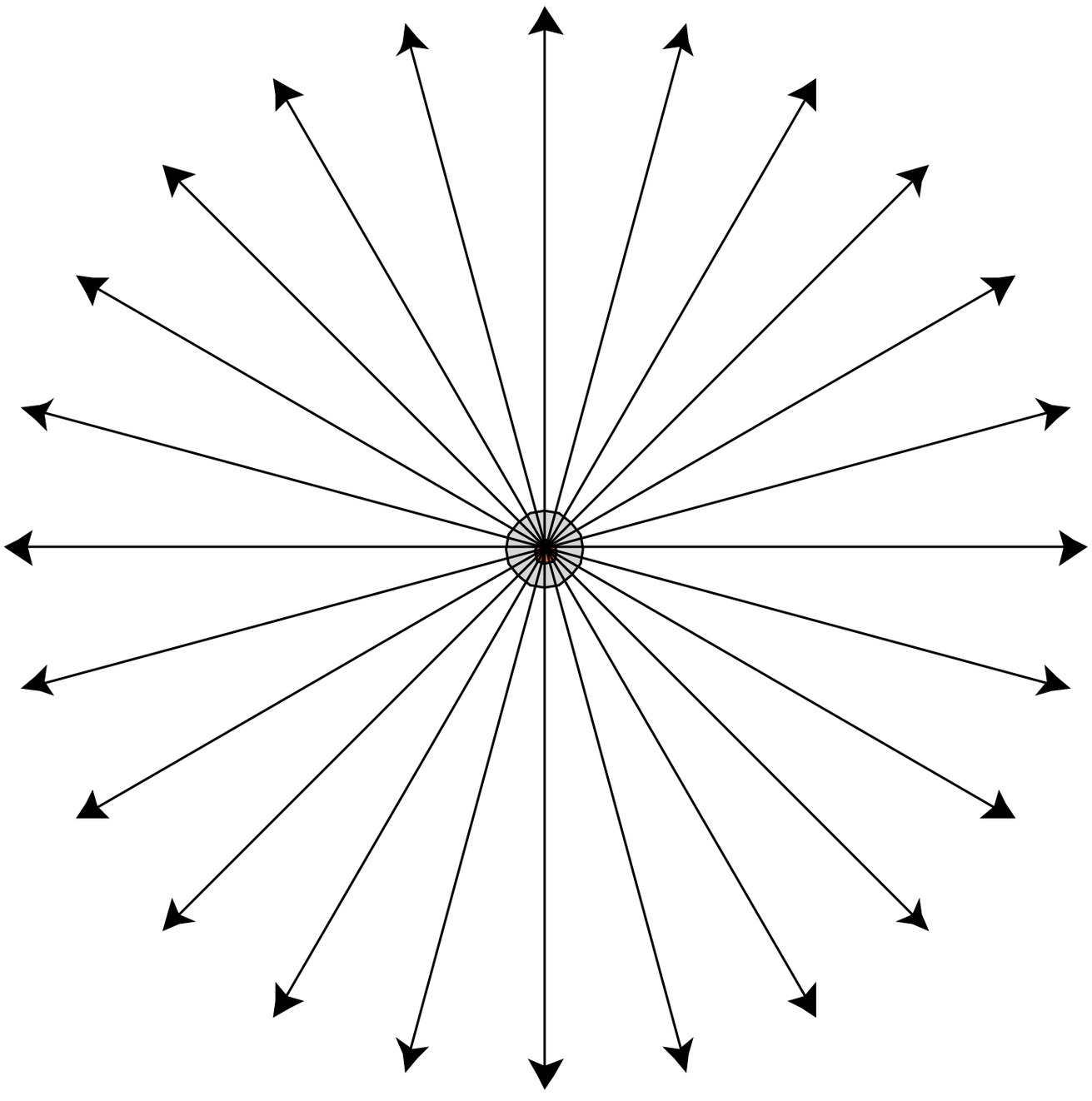
$$= 4\pi$$

ω as a direction vector



Will use ω to denote a direction vector (unit length)

Isotropic Point Source



$$\Phi = \int_{S^2} I \, \mathrm{d}\omega$$
$$= 4\pi I$$

$$I = \frac{\Phi}{4\pi}$$

Modern LED Light

Output: 815 lumens

(11W LED replacement
for 60W incandescent)

Radiant intensity?

Assume isotropic:
Intensity = 815 lumens / 4pi sr
= 65 candelas

Thank you!

(And thank Prof. Ravi Ramamoorthi and Prof. Ren Ng for many of the slides!)