SPACE PARTITIONING ALGORITHMS

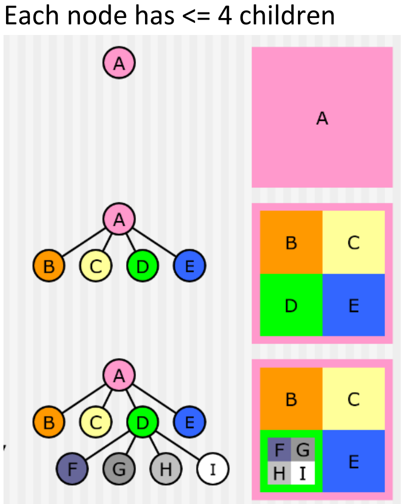
Two types:

1. **Space driven**
2. **Data driven**

SPACE-DRIVEN

1. **Fixed grid partitioning:** The space is partitioned into K equal sized grids, and each grid is a single partition.
2. **Binary space partitioning (BSP):** BSP recursively divides the space into two parts until the leaf partitions satisfy the constraints such as minimum number of spatial objects in a partition.
   - It is a binary tree.
   - Each nodes has <= 2 children.
   - Space is divided using hyperplanes.
   - Hyperplanes divides the space into two parts known as half-spaces.
3. **Variations of BSP:**

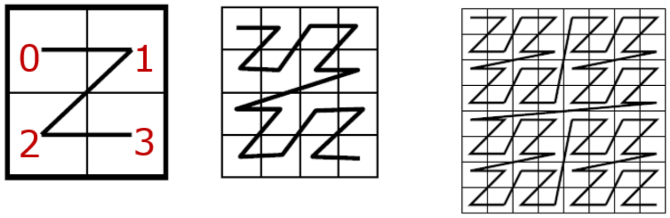
   a. **Quadtrees**: Partitions 2D space.
   - Each node has <= 4 children



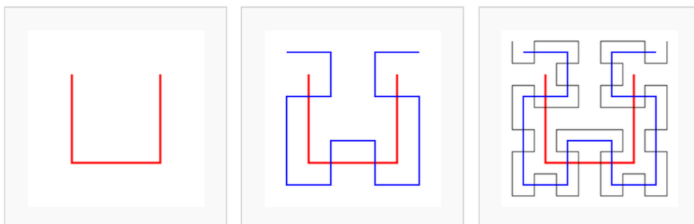
   b. **Octrees:** Partitions 3D space.
   - Each node has <= 8 children.

4. **Z-Ordering curve (Space Filling Curve):** This technique sorts the sample points by their order on the Z-curve and partitions the curve into n splits, each containing roughly ⌊k/n⌋ points. It uses

the distribution method to assign a record r to one cell by mapping the center point of its MBR to one of the n splits.



5. **Hilbert Curve (Space Filling Curve):** Same as Z-ordering but uses Hilbert curve.

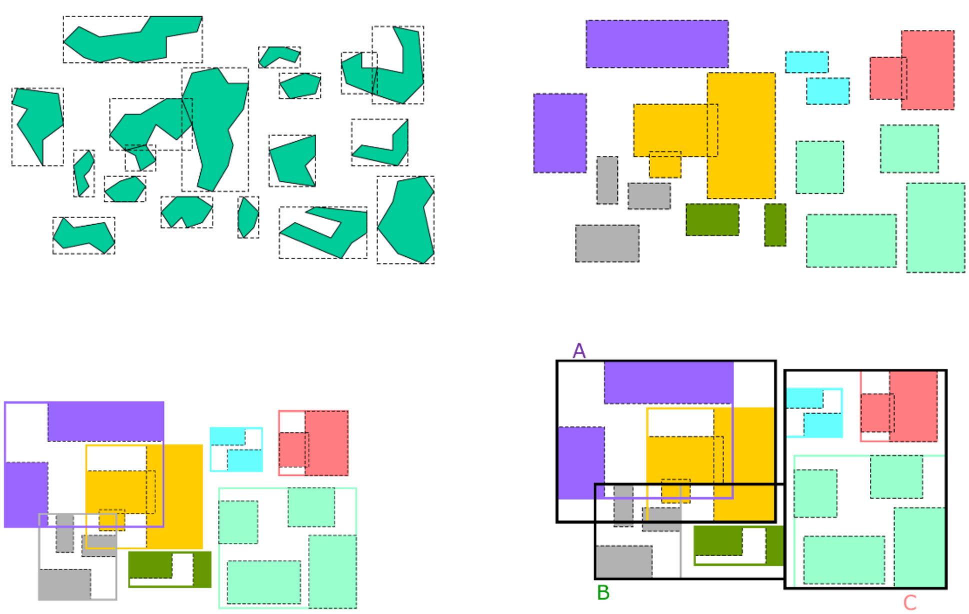


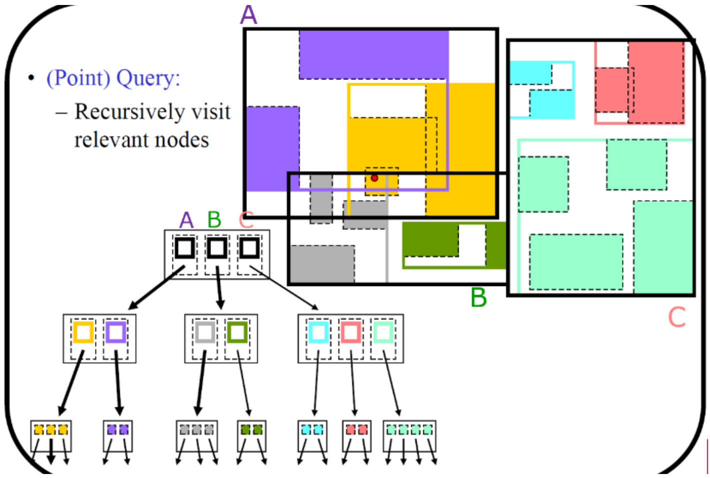
First order First and second order First to third order

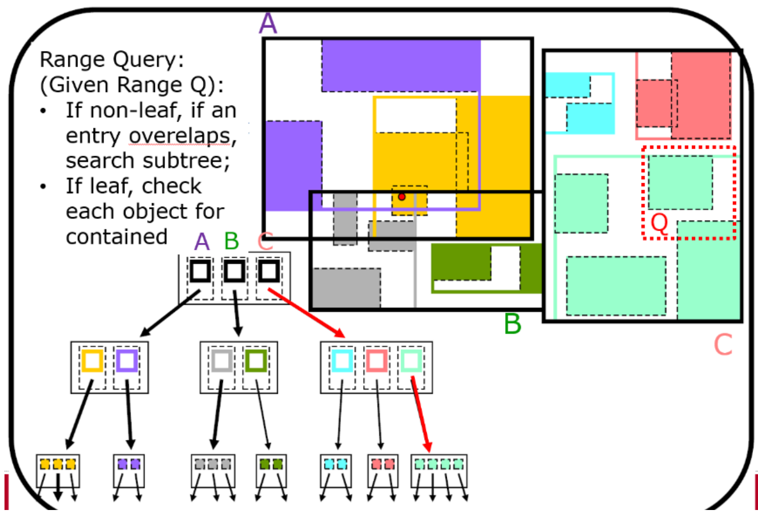
DATA-DRIVEN: Data structure adapts itself to the object distribution.

---

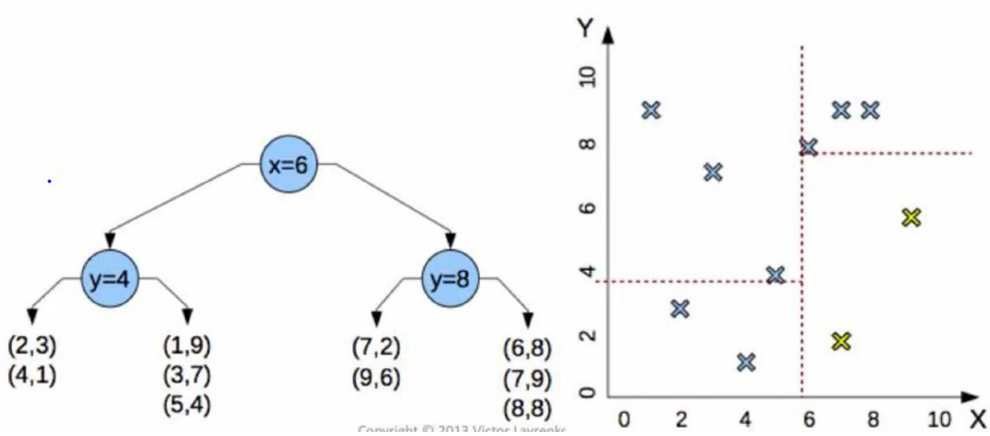
1. **R-tree:** Main idea is group nearby objects and represent them as minimum bounding rectangles (MBRs) in the next higher level of tree.



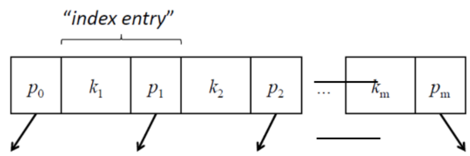


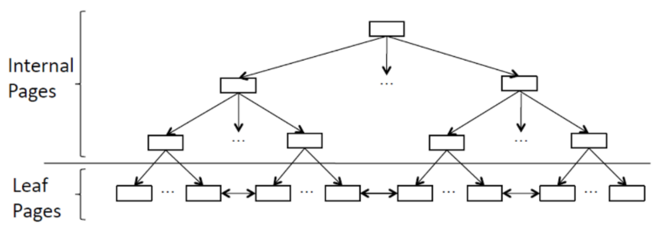


2. **Kd-tree:** Recursively split space into half on exactly one feature and rotate through features. In the example, it is partitioned through feature x's median and then partitioned through feature y's median.

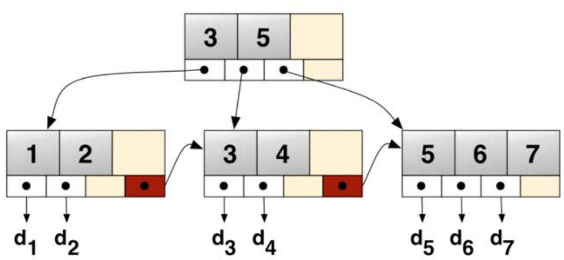


3. **B-tree:**
    - B-tree is a generalization of Binary Search tree in which a node can have more than two children.
    - Each tree node is mapped to a disk page.

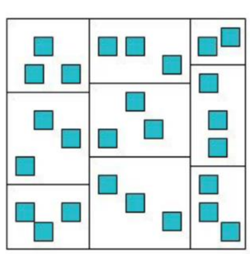




Sample example:



4. **Sort-Tile Recursive partitioning (STR):** STR first partitions the spatial universe into vertical strips then, each vertical strip is further partitioned in horizontal direction.
Approach: This technique bulk loads the random sample into an R-tree using the STR algorithm [8] and the capacity of each node is set to $\lfloor k/n \rfloor$. The MBRs of leaf nodes are used as cell boundaries. Boundary objects are handled using the distribution method where it assigns a record r to the cell with maximal overlap.
Example: Tile the data space of size = $\sqrt{\frac{r}{n}}$ using vertical slices.
Where r = number of rectangles.
n = blocking factor
P = leaf node page = $\left\lceil \sqrt{\frac{r}{n}} \right\rceil$
For r = 25, n = 3, n_{Tile} = 9, n_v=3, n_H=3



IMPLEMENTATION

Implemented the following two algorithms for 3D data and visualized the partitions in Matlab.

1. Binary Space Partition algorithm
2. Sort Tile Partitioning algorithm

RESULTS:

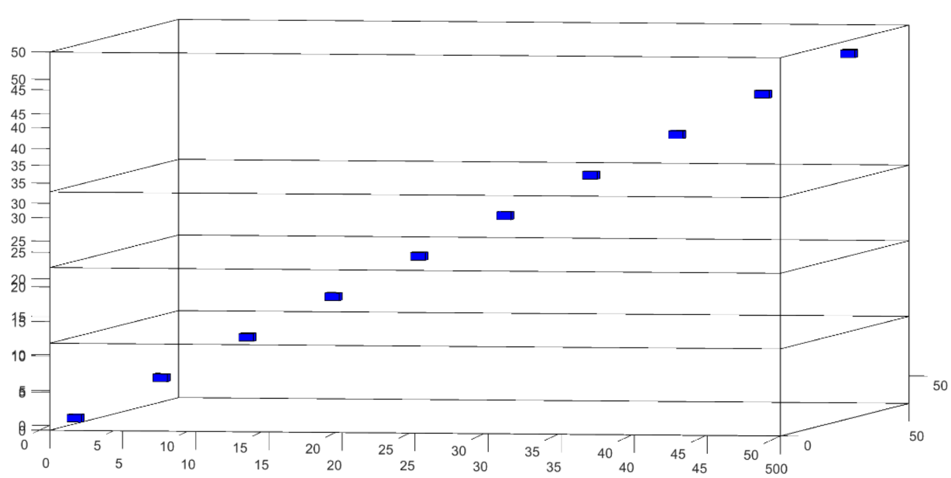
For input data with tile_id as 2465, and each line as min_x, min_y, min_z, max_x, max_y, max_z representing the bounding boxes of the spatial objects.

2465 1 1 1 2 2 2
2465 6 6 6 7 7 7
2465 11 11 11 12 12 12
2465 16 16 16 17 17 17
2465 21 21 21 22 22 22
2465 26 26 26 27 27 27
2465 31 31 31 32 32 32
2465 36 36 36 37 37 37
2465 41 41 41 42 42 42
2465 46 46 46 47 47 47

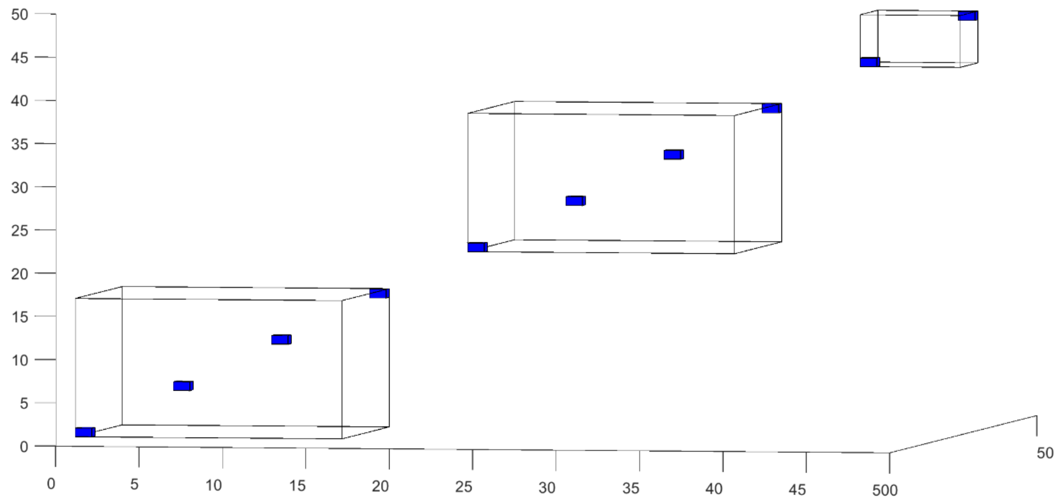
NOTE: Blue bounding boxes are the input spatial objects and the whole space is partitioned using BSP/STR partitioning.

1. **Binary space partitioning (BSP):**
   BSP creates partitioning output as for space volume as 0,0,0,50,50,50 and bucket_size=4.

2465BSP0	0	0	0	50	50	11.5
2465BSP1	0	0	11.5	50	50	21.5
2465BSP2	0	0	21.5	50	50	31.5
2465BSP3	0	0	31.5	50	50	50



2. **Sort tile partitioning (STR):**
   STR creates partitioning output as:
   2465STR0 1 1 1 17 17 17
   2465STR1 21 21 21 37 37 37
   2465STR2 41 41 41 47 47 47



References:

1. Vo, H., Aji, A. and Wang, F., 2014, November. Sato: A spatial data partitioning framework for scalable query processing. In *Proceedings of the 22nd ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*(pp. 545-548). ACM.
2. Leutenegger, S.T., Lopez, M.A. and Edgington, J., 1997, April. STR: A simple and efficient algorithm for R-tree packing. In *Data Engineering, 1997. Proceedings. 13th international conference on* (pp. 497-506). IEEE.