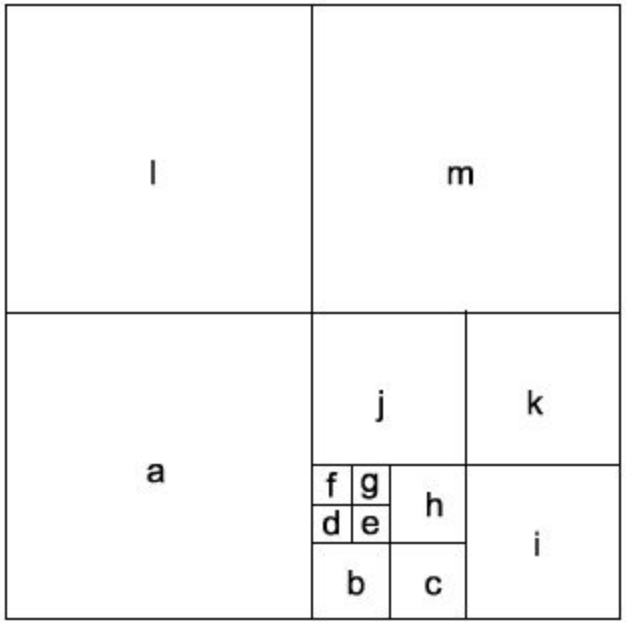
Linear Octree

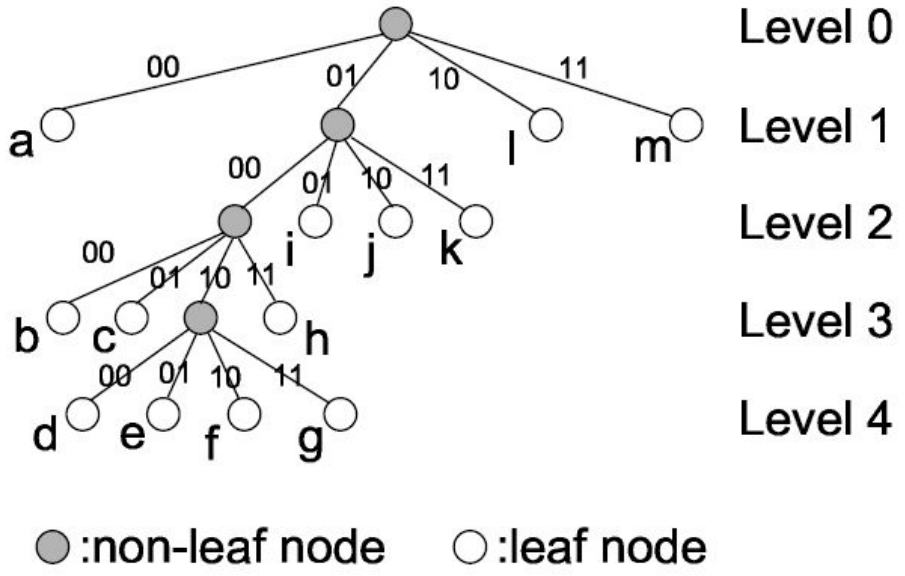
Ref:

- Tu and O'Hallaron 2004
- Simple and Efficient Traversal Methods for Quadtrees and Octrees, Frisken and Perry 2002

Pointer-based Representation



10	11
00	01



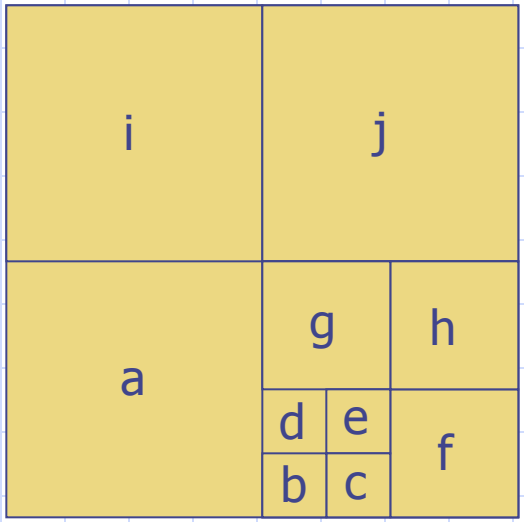
Linear Octree

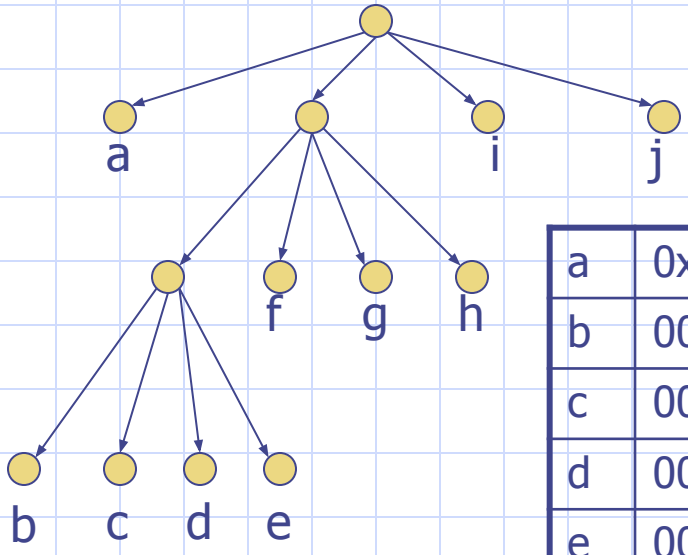
- Assign a unique key (locational code) to each node
- Represent an octree as a collection of lear nodes comprising it
- Extremely useful in practice when main memory cannot accommodate a pointer-based octree

Locational code

- The code for each node is of the same length (zero-padded)
- Level of the node is also attached

Octree and Linear Octree





a	000000_01	f	010100_10
b	000000_11	g	011000_10
c	000001_11	h	011100_10
d	000010_11	i	100000_01
e	000011_11	j	110000_01

a	0xx
b	000
c	001
d	002
e	003
f	01x
g	02x
h	03x
i	2xx
j	3xx

Base-4 codes

Observation

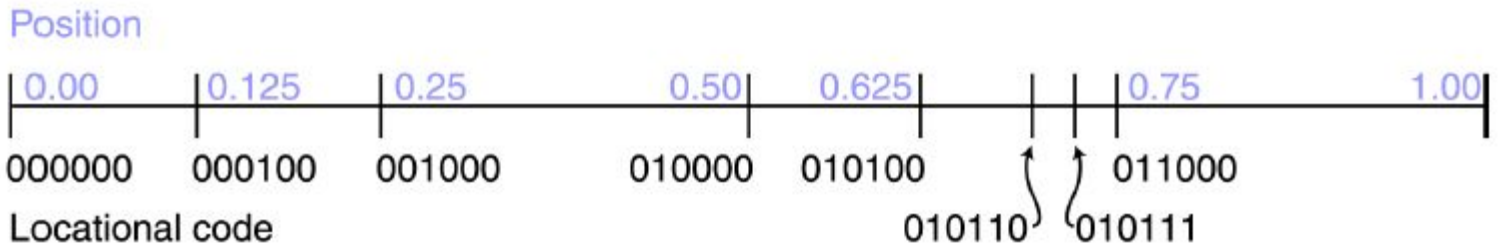
- When we sort the leaf nodes according to their locational codes (as binary scalars), the order is the same as the preorder traversal of the octree
  - In octree, we may use octal number for coding

Simple and Efficient Traversal Methods for Quadtrees and Octrees

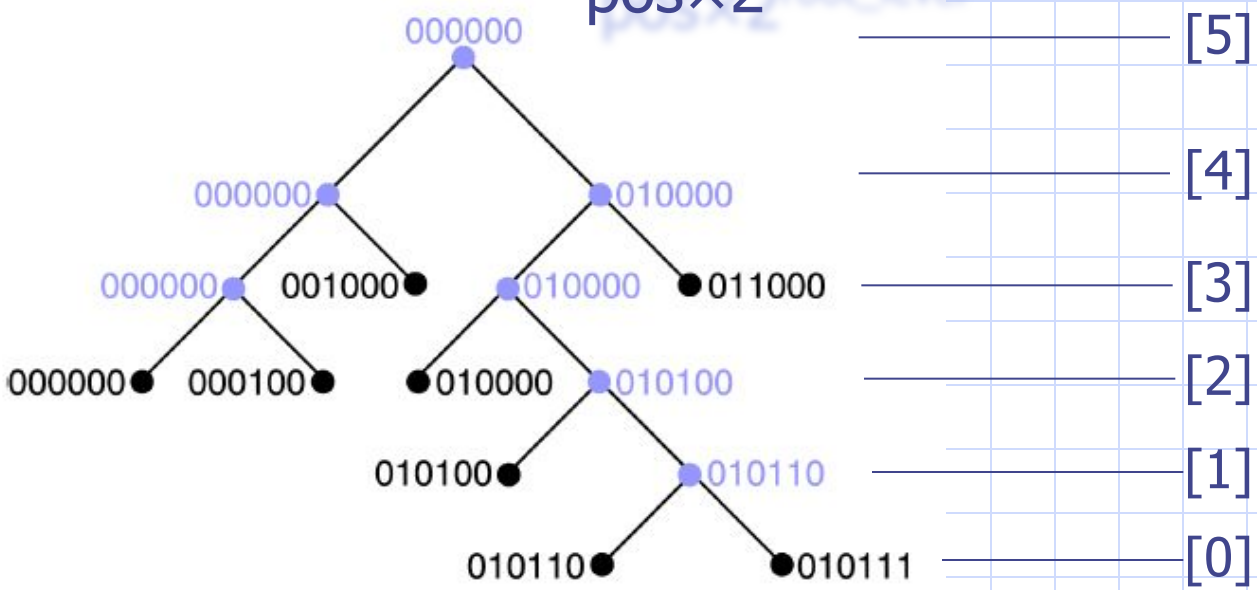
- Frisken and Perry 2002 (MERL)
- Usually locational code is for linear octrees (for more compact representation); here it is used in tree-based representations to facilitate a simpler and more efficient query for point location, region location and neighbor finding

Representation

Depth of tree: N_LEVELS
Level of root: N_LEVELS-1
Level of smallest possible cell: 0

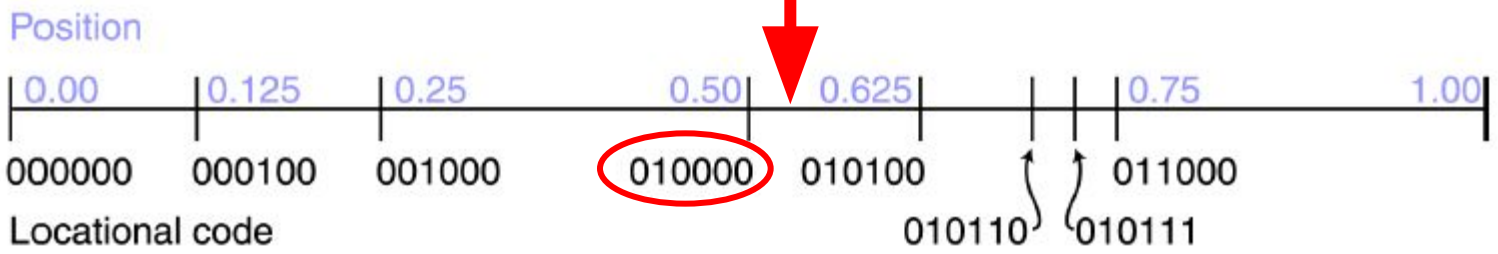


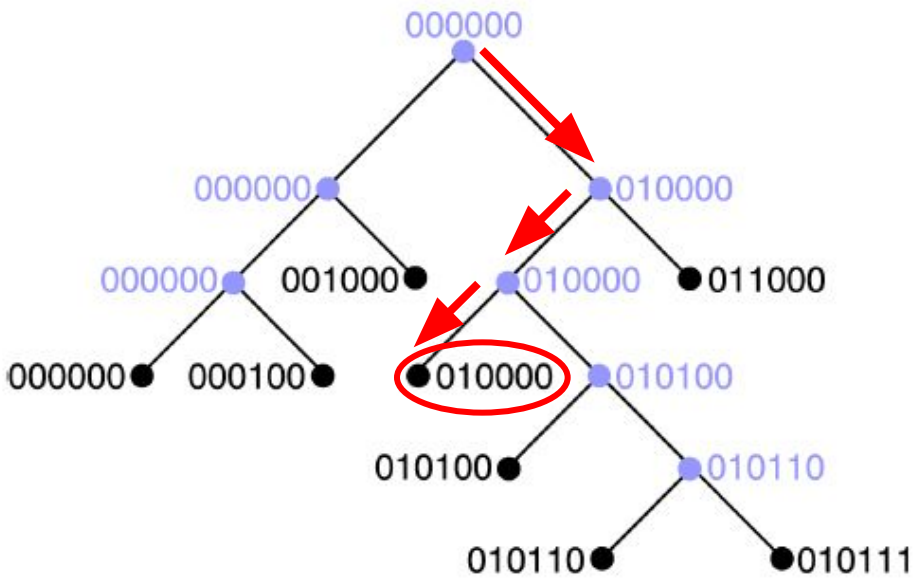
Locational code = $pos \times 2^{root\_level}$



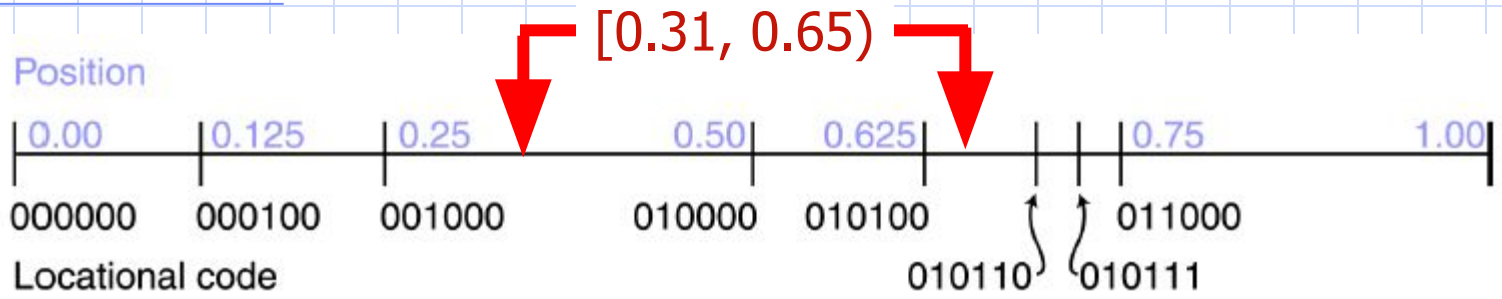
Point Location

Binary(trunc(0.55*32))
=binary(17) = 010001

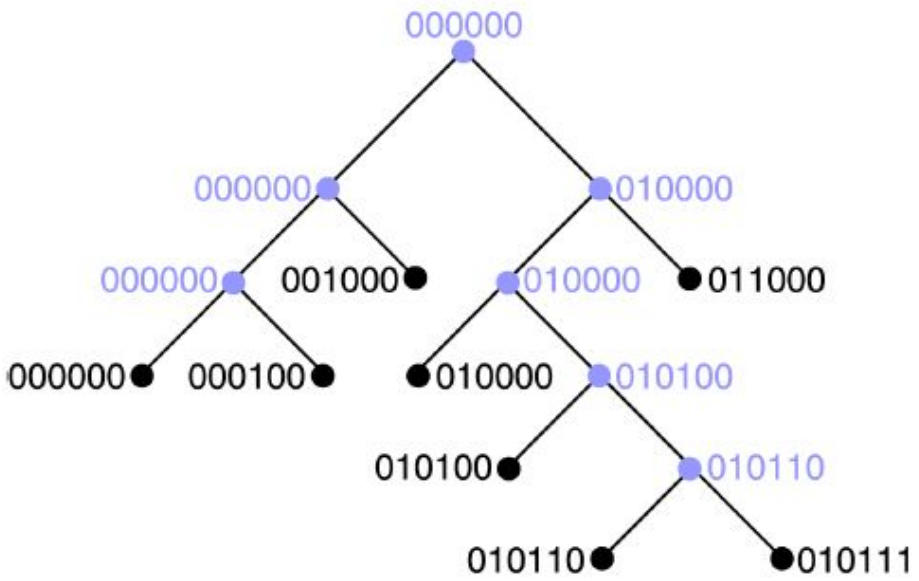




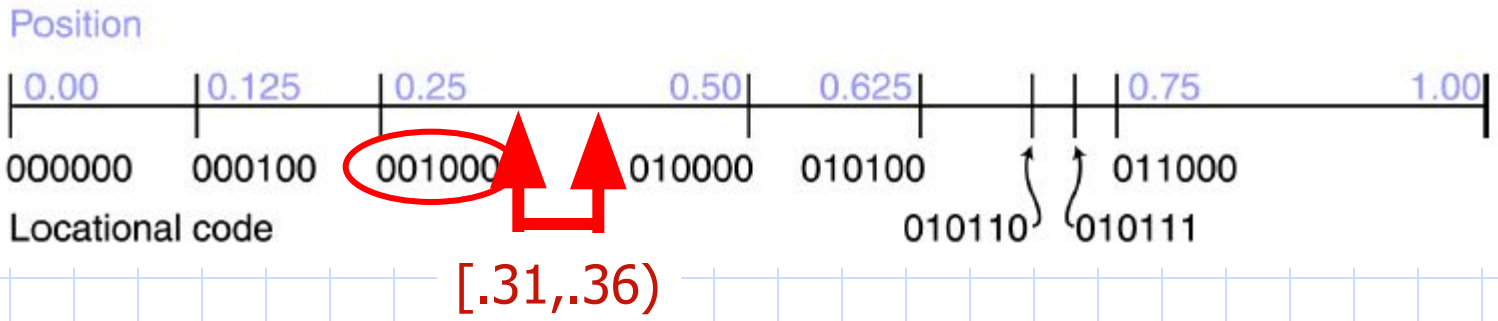
Region Location (1)

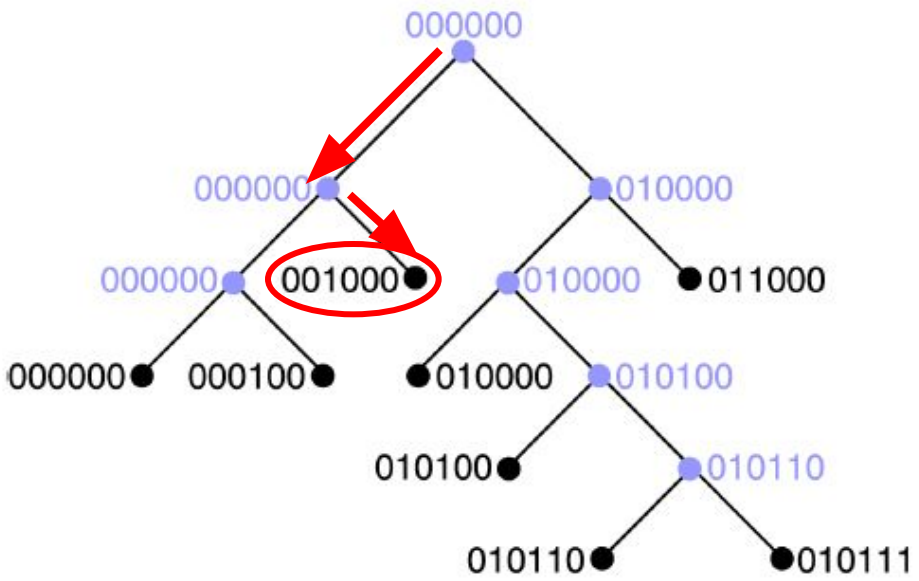


Code(0.31)=001001
Code(0.65)=010101
Xor =011100



Region Location (2)





Code(0.31)=001001
Code(0.36)=001010
Xor =000011

Neighbor Search