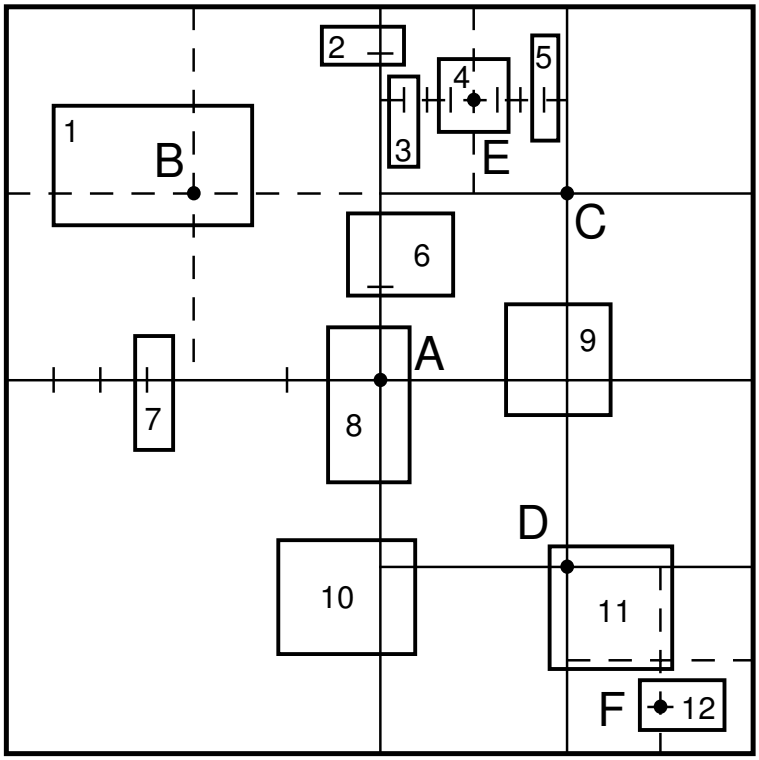
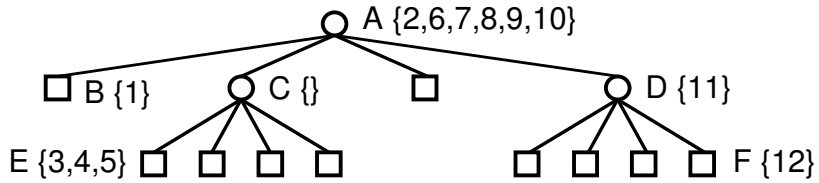
MX-CIF QUADTREE (Kedem)

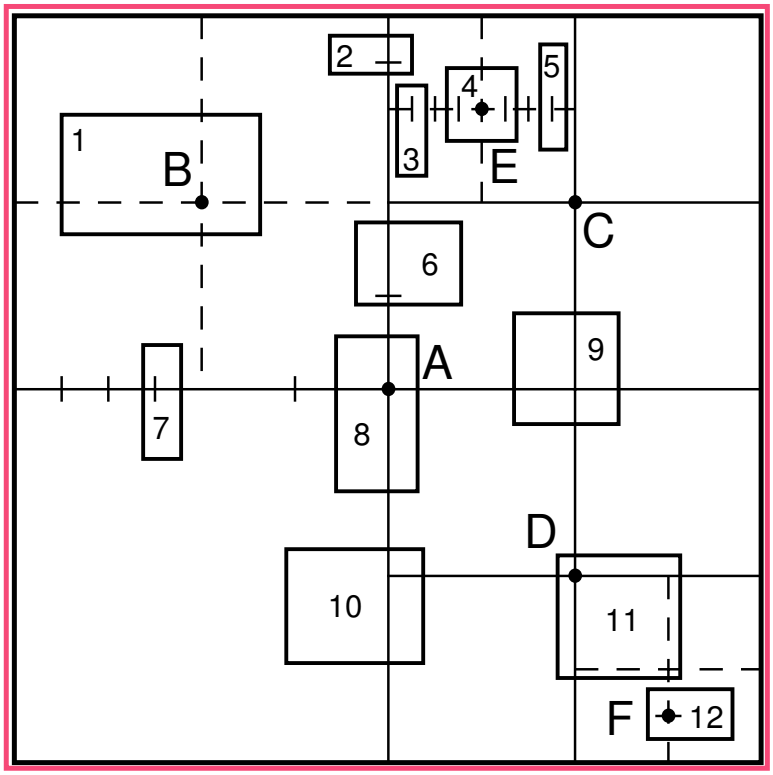
1. Collections of small rectangles for VLSI applications
2. Each rectangle is associated with its minimum enclosing quadtree block
3. Like hashing: quadtree blocks serve as hash buckets

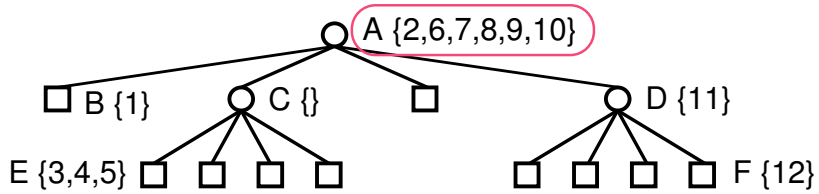




MX-CIF QUADTREE (Kedem)

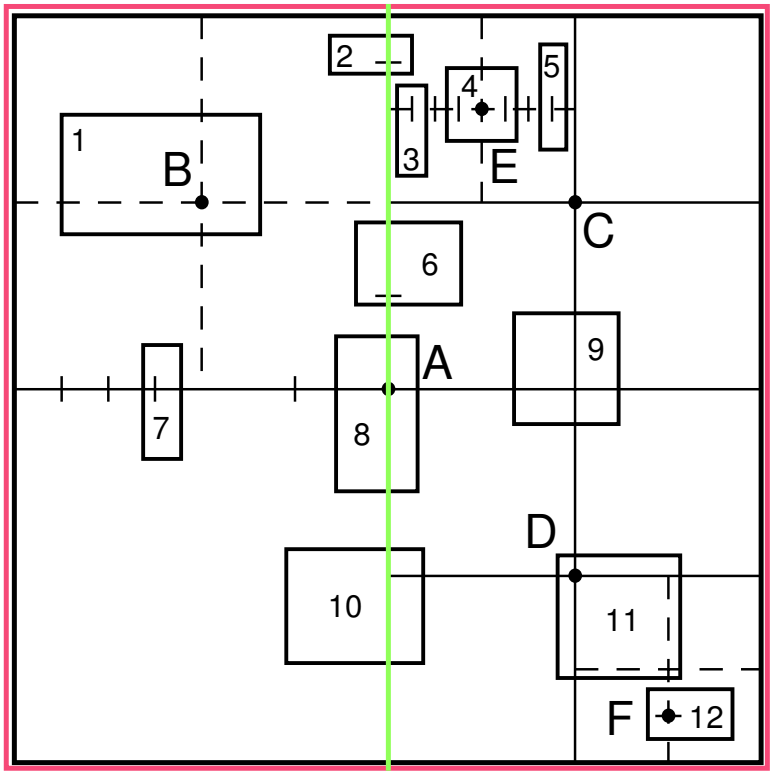
1. Collections of small rectangles for VLSI applications
2. Each rectangle is associated with its minimum enclosing quadtree block
3. Like hashing: quadtree blocks serve as hash buckets
4. Collision = more than one rectangle in a block
   - resolve by using two one-dimensional MX-CIF trees to store the rectangle intersecting the lines passing through each subdivision point



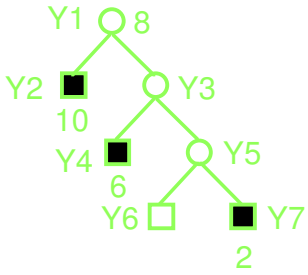


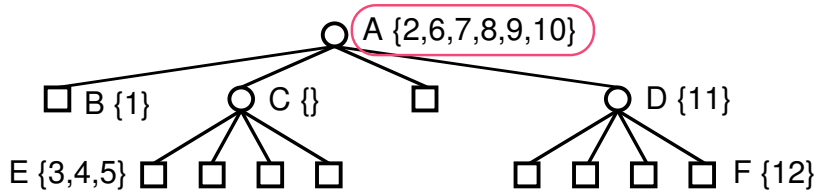
MX-CIF QUADTREE (Kedem)

1. Collections of small rectangles for VLSI applications
2. Each rectangle is associated with its minimum enclosing quadtree block
3. Like hashing: quadtree blocks serve as hash buckets
4. Collision = more than one rectangle in a block
   - resolve by using two one-dimensional MX-CIF trees to store the rectangle intersecting the lines passing through each subdivision point
   - one for y-axis



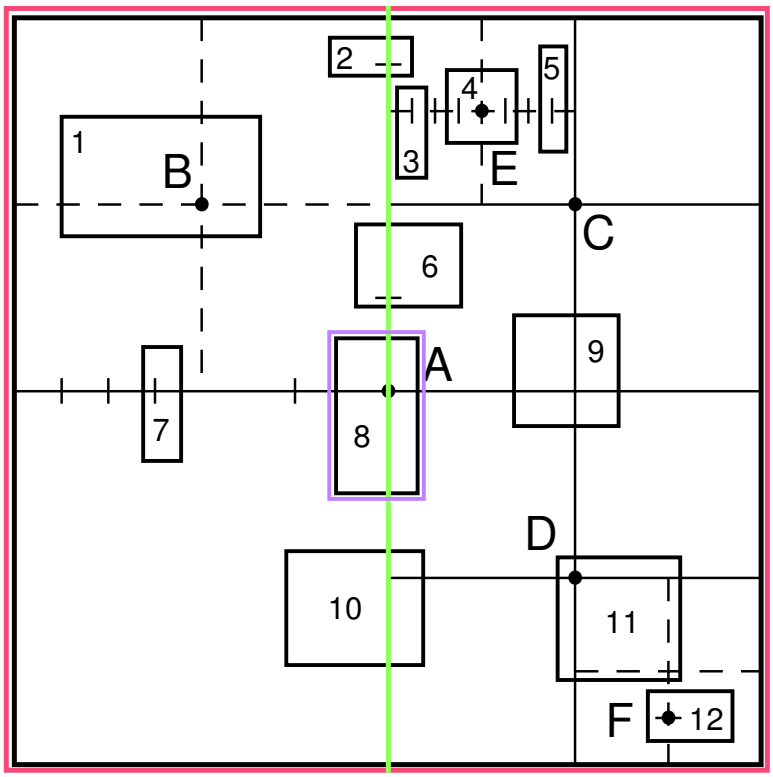
Binary tree for *y*-axis through A



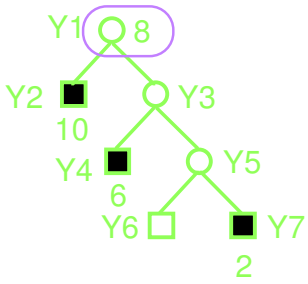


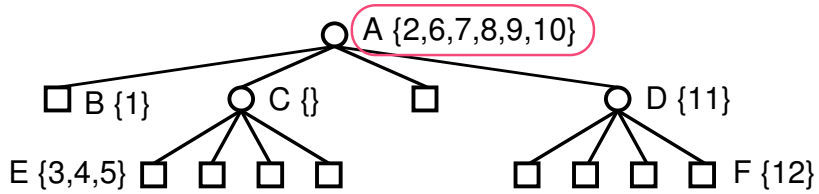
MX-CIF QUADTREE (Kedem)

1. Collections of small rectangles for VLSI applications
2. Each rectangle is associated with its minimum enclosing quadtree block
3. Like hashing: quadtree blocks serve as hash buckets
4. Collision = more than one rectangle in a block
   - resolve by using two one-dimensional MX-CIF trees to store the rectangle intersecting the lines passing through each subdivision point
   - one for y-axis
   - if a rectangle intersects both *x* and *y* axes, then associate it with the *y* axis



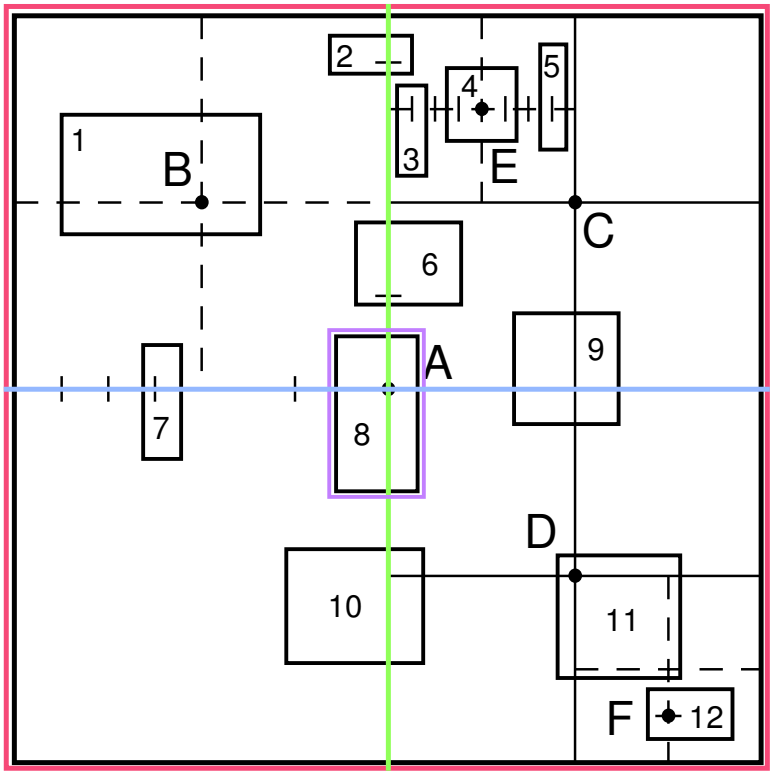
Binary tree for *y*-axis through A



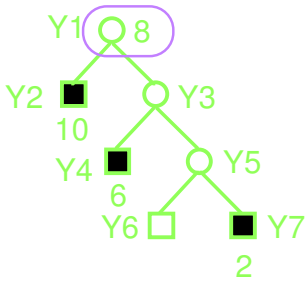


MX-CIF QUADTREE (Kedem)

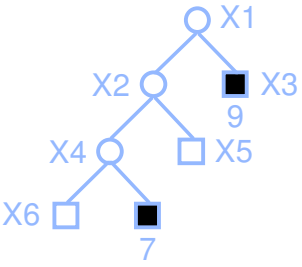
1. Collections of small rectangles for VLSI applications
2. Each rectangle is associated with its minimum enclosing quadtree block
3. Like hashing: quadtree blocks serve as hash buckets
4. Collision = more than one rectangle in a block
   - resolve by using two one-dimensional MX-CIF trees to store the rectangle intersecting the lines passing through each subdivision point
   - one for y-axis
   - if a rectangle intersects both *x* and *y* axes, then associate it with the *y* axis
   - one for x-axis

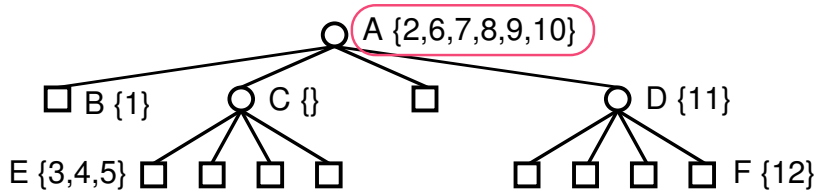


Binary tree for *y*-axis through A



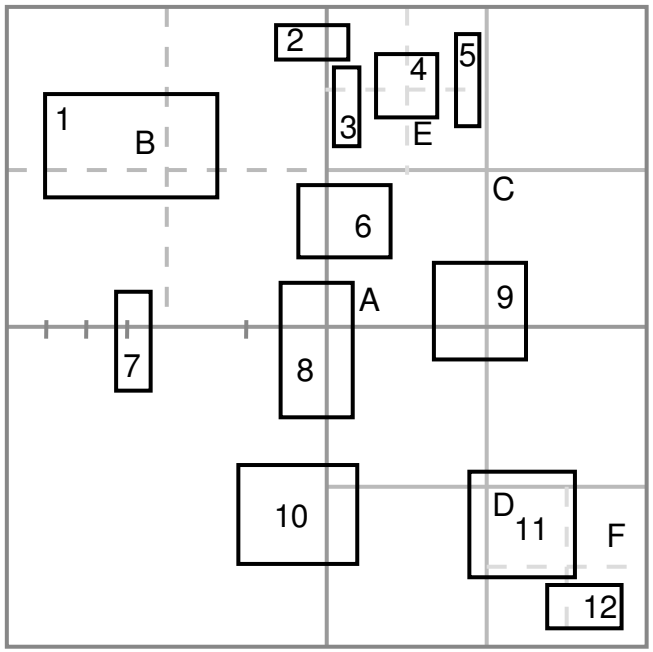
Binary tree for *x*-axis through A

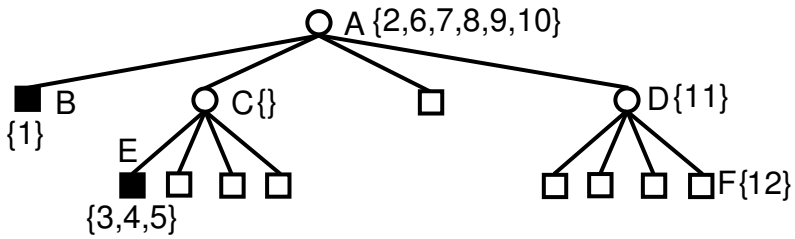




Loose Quadtree (Octree)/Cover Fieldtree

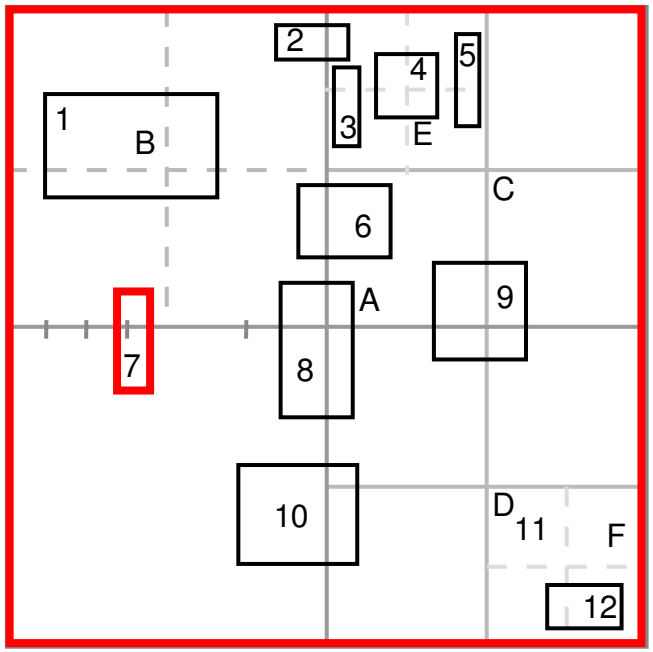
- Overcomes drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o

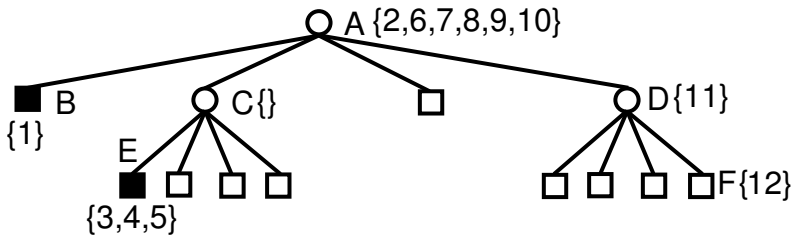




Loose Quadtree (Octree)/Cover Fieldtree

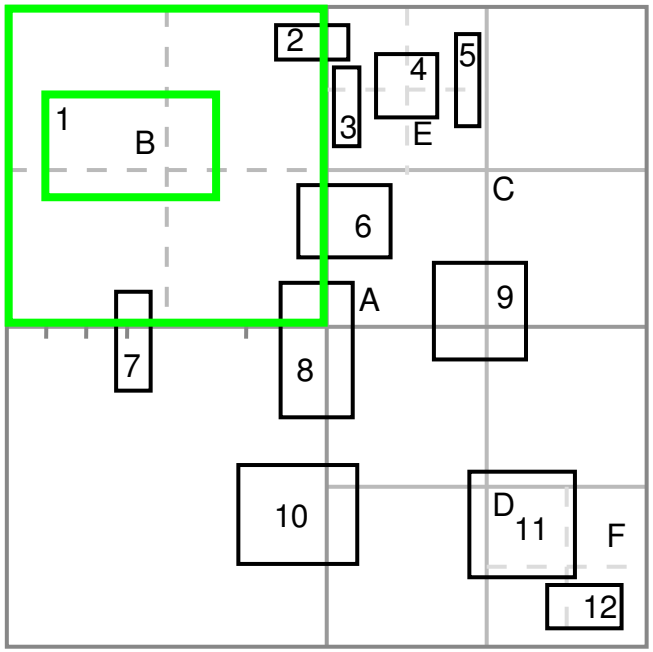
- Overcomes drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Instead, it depends on the position of the centroid of o and often considerably larger than o

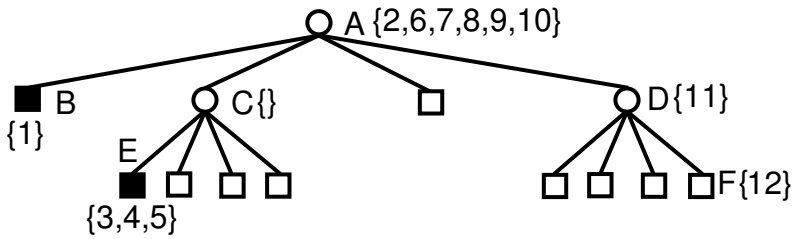




Loose Quadtree (Octree)/Cover Fieldtree

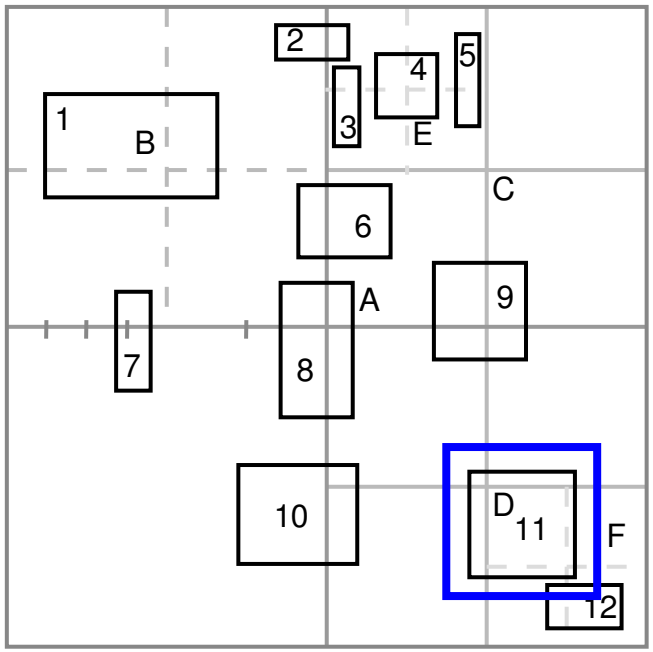
- Overcomes drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Instead, it depends on the position of the centroid of o and often considerably larger than o

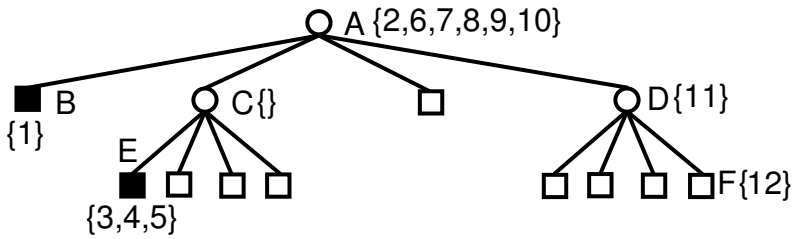




Loose Quadtree (Octree)/Cover Fieldtree

- Overcomes drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Instead, it depends on the position of the centroid of o and often considerably larger than o
- Solution: expand size of space spanned by each quadtree block of width w by expansion factor p ($p > 0$) so expanded block is of width $(1 + p)w$

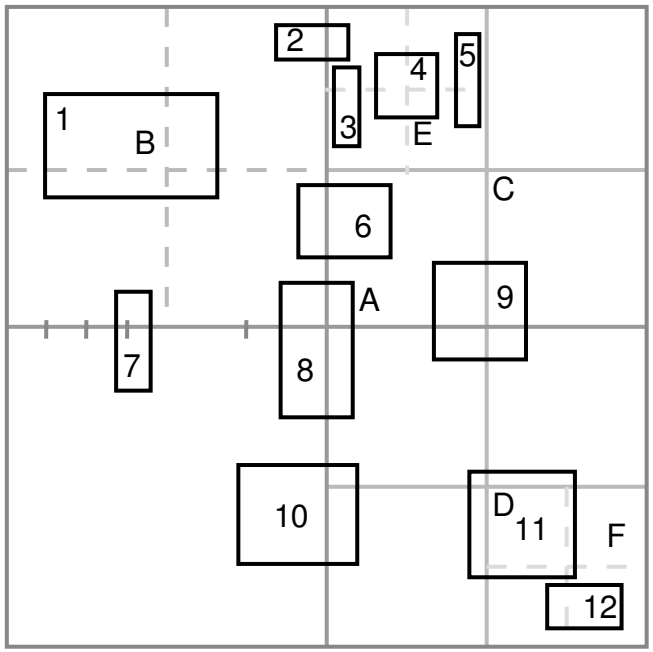


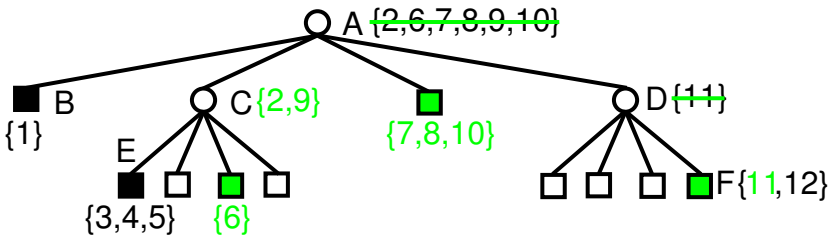


Loose Quadtree (Octree)/Cover Fieldtree

- Overcomes drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Instead, it depends on the position of the centroid of o and often considerably larger than o
- Solution: expand size of space spanned by each quadtree block of width w by expansion factor p ($p > 0$) so expanded block is of width $(1 + p)w$

  1. $p = 0.3$

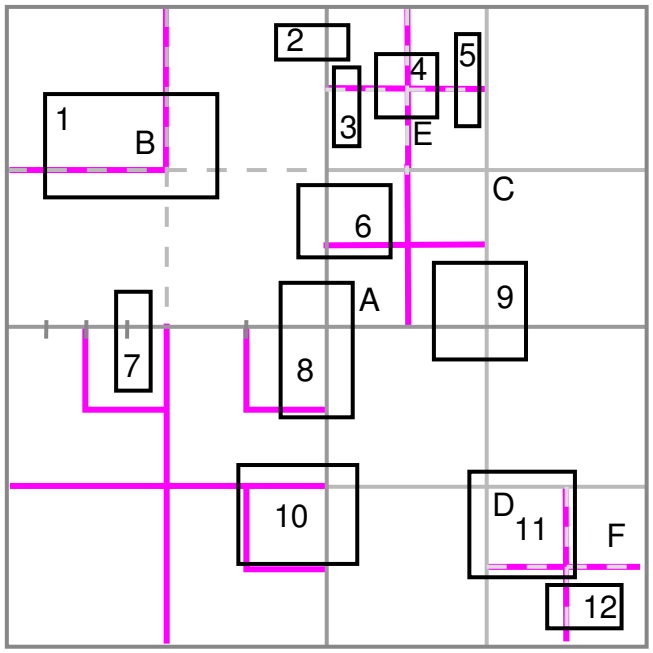


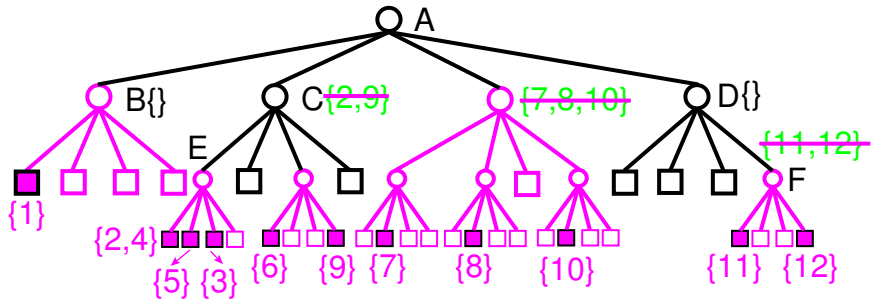


Loose Quadtree (Octree)/Cover Fieldtree

- Overcomes drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Instead, it depends on the position of the centroid of o and often considerably larger than o
- Solution: expand size of space spanned by each quadtree block of width w by expansion factor p ($p > 0$) so expanded block is of width $(1 + p)w$

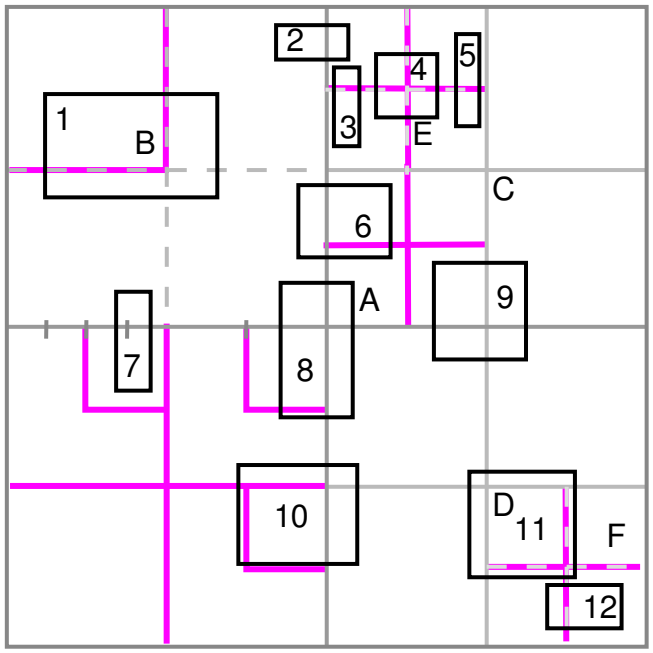
  1. $p = 0.3$
  2. $p = 1.0$

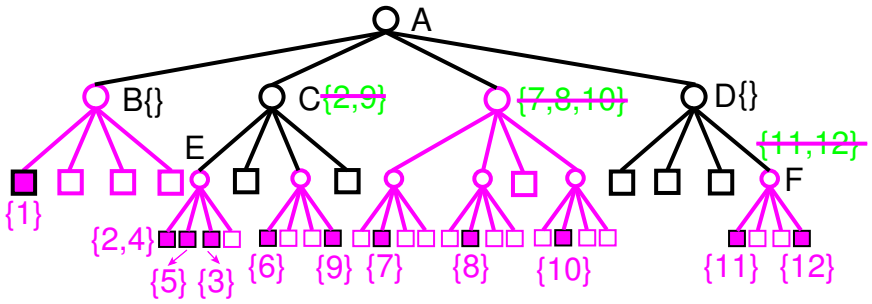




Loose Quadtree (Octree)/Cover Fieldtree

- Overcomes drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Instead, it depends on the position of the centroid of o and often considerably larger than o
- Solution: expand size of space spanned by each quadtree block of width w by expansion factor p ($p > 0$) so expanded block is of width $(1+p)w$
  1. $p = 0.3$
  2. $p = 1.0$
- Maximum w (i.e., minimum depth of minimum enclosing quadtree block) is a function of p and radius r of o and independent of position of centroid of o
  1. Range of possible ratios $w/2r$: $1/(1+p) \cdot w/2r < 2/p$
  2. For $p \geq 1$, restricting w and r to powers of 2, $w/2r$ takes on at most 2 values and usually just 1





Partition Fieldtree

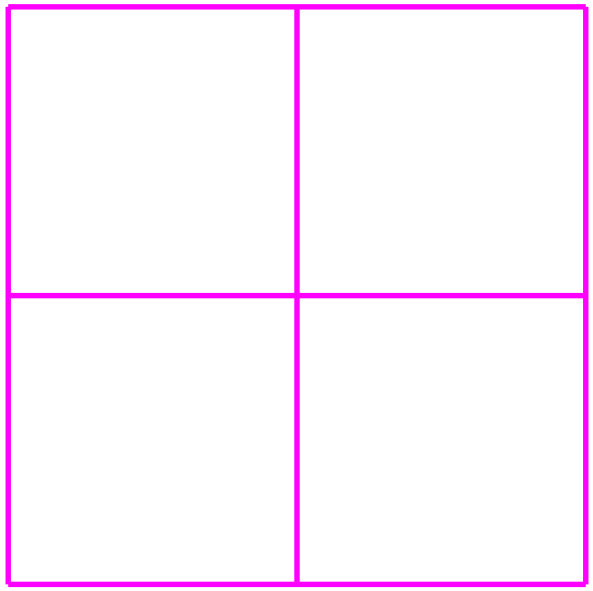
- Alternative to loose quadtree (octree)/cover fieldtree at overcoming drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o

Partition Fieldtree

- Alternative to loose quadtree (octree)/cover fieldtree at overcoming drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Achieves similar result by shifting positions of the centroid of quadtree blocks at successive levels of the subdivision by one half of the width of the block that is being subdivided

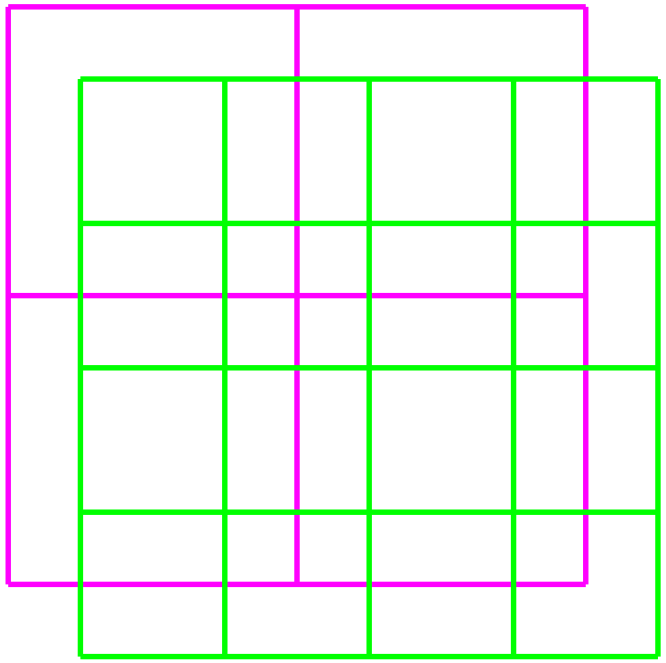
Partition Fieldtree

- Alternative to loose quadtree (octree)/cover fieldtree at overcoming drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Achieves similar result by shifting positions of the centroid of quadtree blocks at successive levels of the subdivision by one half of the width of the block that is being subdivided



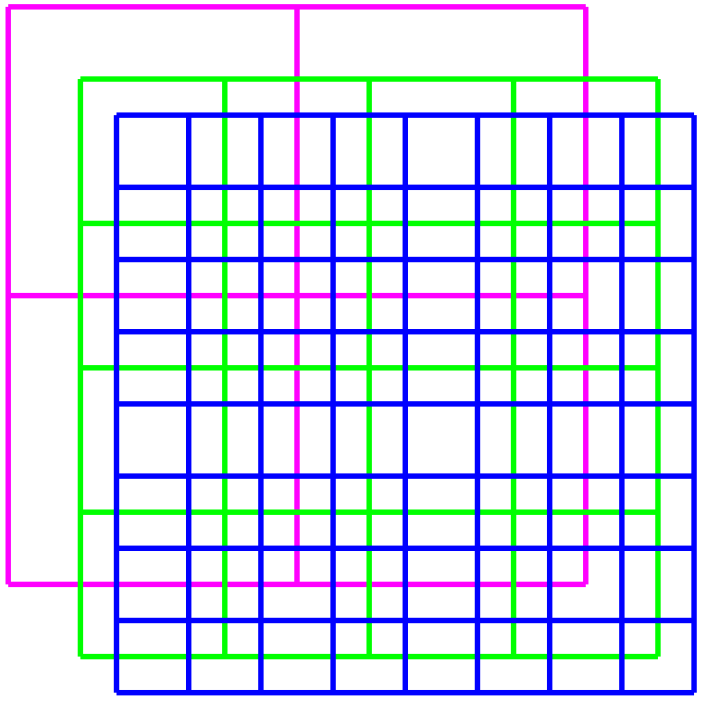
Partition Fieldtree

- Alternative to loose quadtree (octree)/cover fieldtree at overcoming drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Achieves similar result by shifting positions of the centroid of quadtree blocks at successive levels of the subdivision by one half of the width of the block that is being subdivided



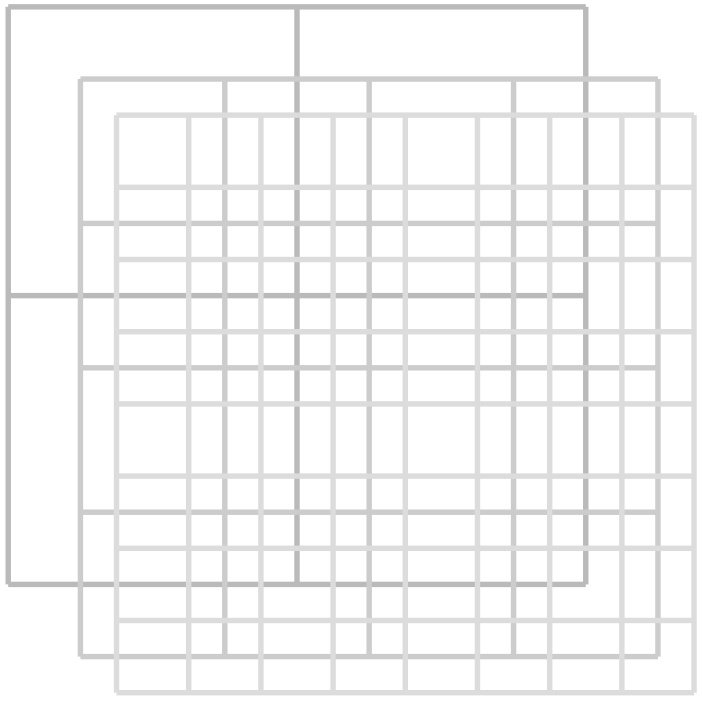
Partition Fieldtree

- Alternative to loose quadtree (octree)/cover fieldtree at overcoming drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Achieves similar result by shifting positions of the centroid of quadtree blocks at successive levels of the subdivision by one half of the width of the block that is being subdivided



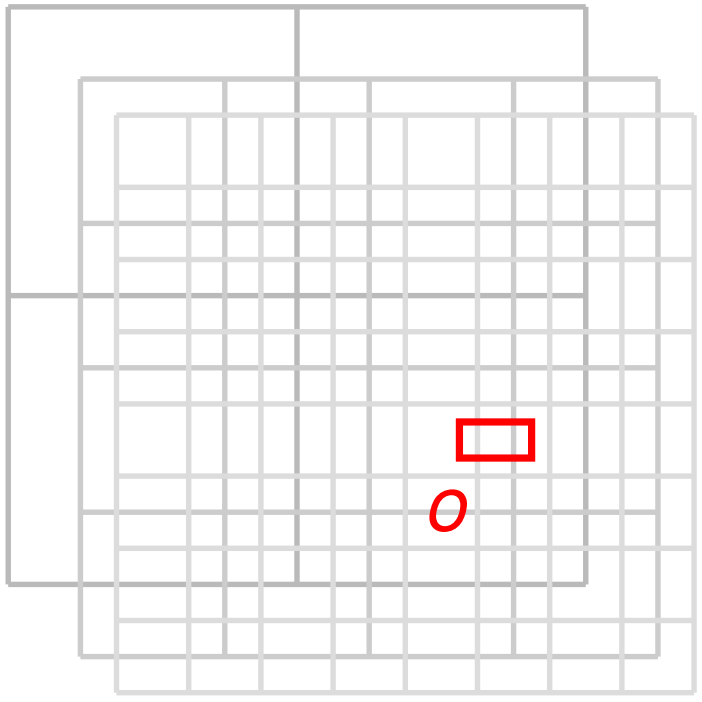
Partition Fieldtree

- Alternative to loose quadtree (octree)/cover fieldtree at overcoming drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Achieves similar result by shifting positions of the centroid of quadtree blocks at successive levels of the subdivision by one half of the width of the block that is being subdivided
- Subdivision rule guarantees that width of minimum enclosing quadtree block for rectangle o is bounded by 8 times the maximum extent r of o



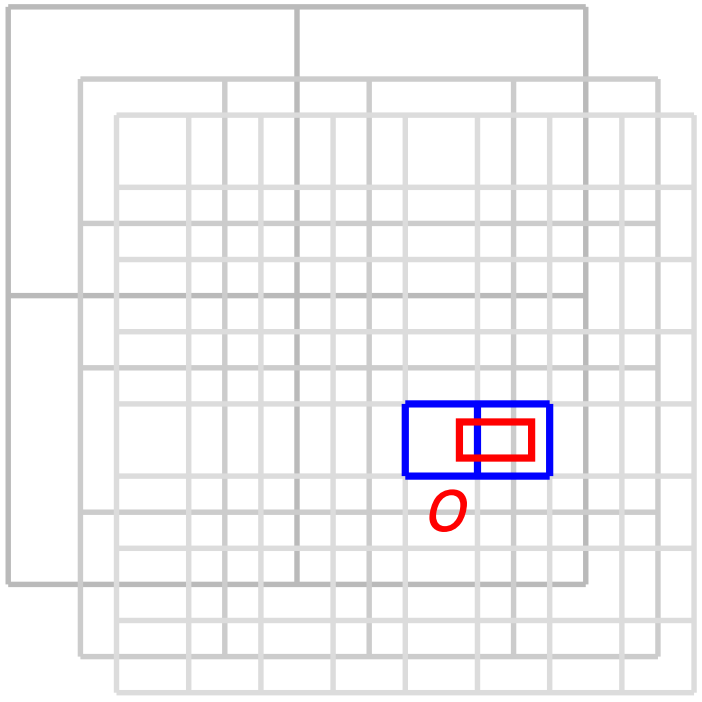
Partition Fieldtree

- Alternative to loose quadtree (octree)/cover fieldtree at overcoming drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Achieves similar result by shifting positions of the centroid of quadtree blocks at successive levels of the subdivision by one half of the width of the block that is being subdivided
- Subdivision rule guarantees that width of minimum enclosing quadtree block for rectangle o is bounded by 8 times the maximum extent r of o



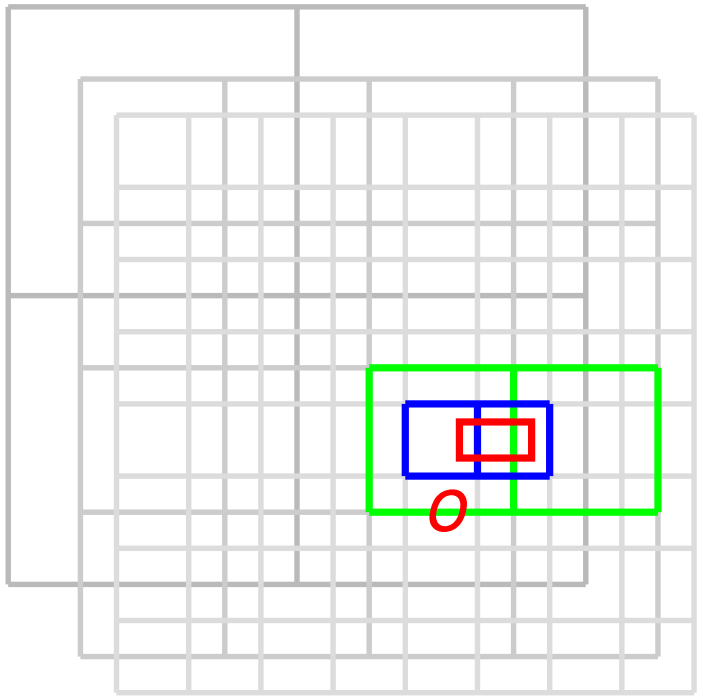
Partition Fieldtree

- Alternative to loose quadtree (octree)/cover fieldtree at overcoming drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Achieves similar result by shifting positions of the centroid of quadtree blocks at successive levels of the subdivision by one half of the width of the block that is being subdivided
- Subdivision rule guarantees that width of minimum enclosing quadtree block for rectangle o is bounded by 8 times the maximum extent r of o



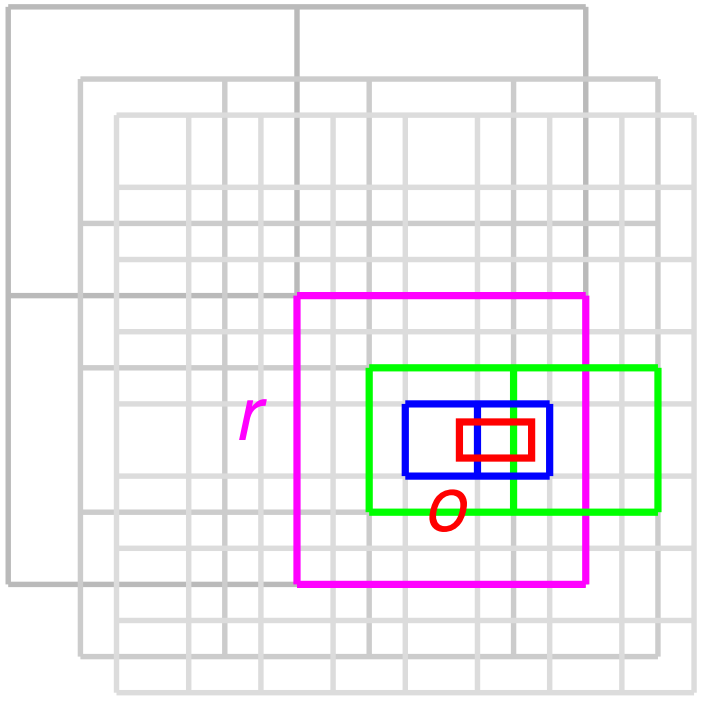
Partition Fieldtree

- Alternative to loose quadtree (octree)/cover fieldtree at overcoming drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Achieves similar result by shifting positions of the centroid of quadtree blocks at successive levels of the subdivision by one half of the width of the block that is being subdivided
- Subdivision rule guarantees that width of minimum enclosing quadtree block for rectangle o is bounded by 8 times the maximum extent r of o



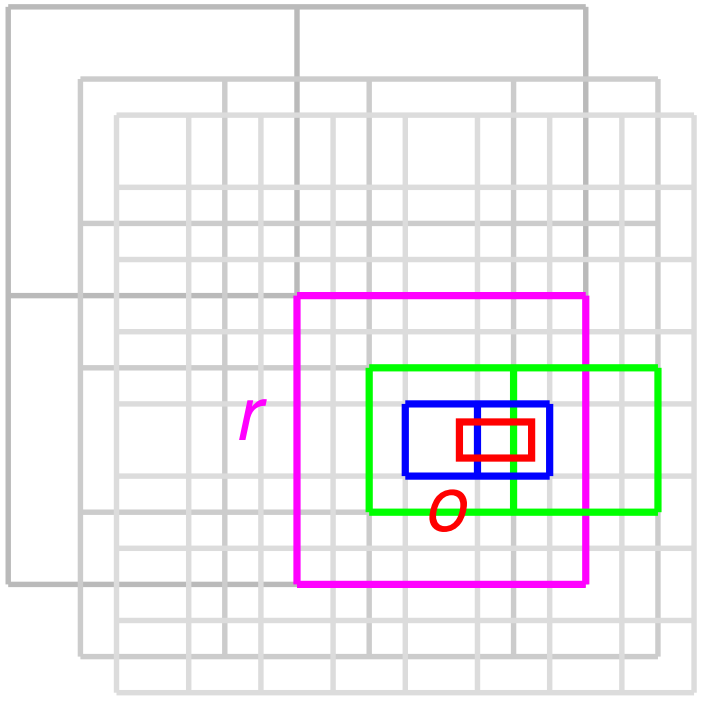
Partition Fieldtree

- Alternative to loose quadtree (octree)/cover fieldtree at overcoming drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Achieves similar result by shifting positions of the centroid of quadtree blocks at successive levels of the subdivision by one half of the width of the block that is being subdivided
- Subdivision rule guarantees that width of minimum enclosing quadtree block for rectangle o is bounded by 8 times the maximum extent r of o



Partition Fieldtree

- Alternative to loose quadtree (octree)/cover fieldtree at overcoming drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Achieves similar result by shifting positions of the centroid of quadtree blocks at successive levels of the subdivision by one half of the width of the block that is being subdivided
- Subdivision rule guarantees that width of minimum enclosing quadtree block for rectangle o is bounded by 8 times the maximum extent r of o
- Same ratio is obtained for the loose quadtree (octree)/cover fieldtree when p =1/4, and thus partition fieldtree is superior to the cover fieldtree when p <1/4



Partition Fieldtree

- Alternative to loose quadtree (octree)/cover fieldtree at overcoming drawback of MX-CIF quadtree that the width w of the minimum enclosing quadtree block of a rectangle o is not a function of the size of o
- Achieves similar result by shifting positions of the centroid of quadtree blocks at successive levels of the subdivision by one half of the width of the block that is being subdivided
- Subdivision rule guarantees that width of minimum enclosing quadtree block for rectangle o is bounded by 8 times the maximum extent r of o
- Same ratio is obtained for the loose quadtree (octree)/cover fieldtree when p =1/4, and thus partition fieldtree is superior to the cover fieldtree when p <1/4
- Summary: cover fieldtree expands the width of the quadtree blocks while the partition fieldtree shifts the positions of their centroids