Field Trees and Loose Quadtrees

Kenny Weiss

CMSC828S

Spring 2005

Images vs. Objects

- Image Based
  - Given a cell (location) find the object(s) of which it is a member
  - *What features are at a location?*
- Object Based
  - Given an object, find its constituent cells (locations)
  - *Where is an object located?*

Loosen Quadtree Restrictions

- Up until now…
  - Each cell only contained one object and objects cannot overlap
- Loosen this restriction
  - Cell need not be entirely covered by and object
  - Several objects may occupy a single cell
    - may also overlap
  - Arbitrary shapes allowed

Limits to decomposition

- Arbitrary shape may be decomposed infinitely
  - If placed in certain locations
  - Use Min Bounding Box to simplify representation

- Coverage based
  - Restrict number of blocks that can cover an object
- Density based
  - Restrict number of objects that can be covered by a block

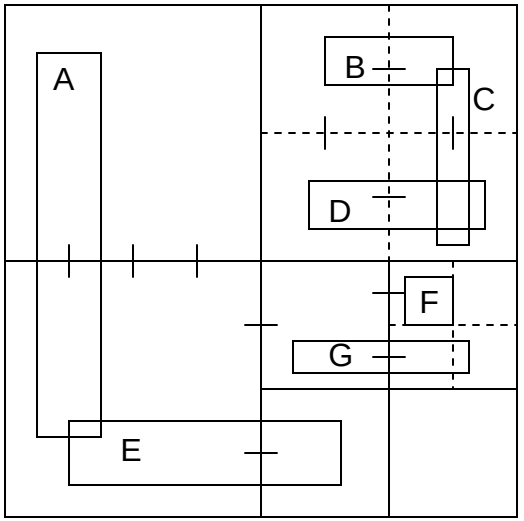
We will focus on Coverage based limits on image based representation

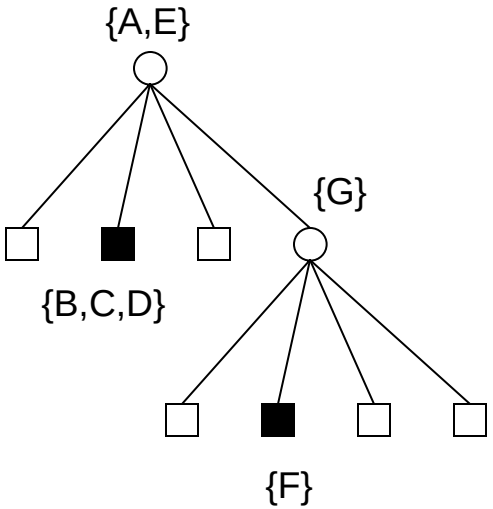
Quadtree Space Partition
Motivation

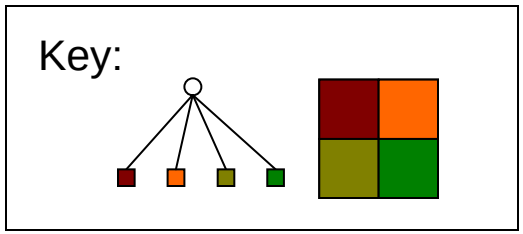
- Store the bounding box of objects in hierarchy
  - Must check every match (at all levels) to see if point is in object
- Goal: try to put objects in block at lowest possible level to minimize unnecessary computation
  - Prune as much as possible

MX-CIF Quadtree

- Object must be covered by at most ONE block
  - Minimum enclosing quadtree block
  - Resolve collisions
    - Use two 1-*dim* MX-CIF structures

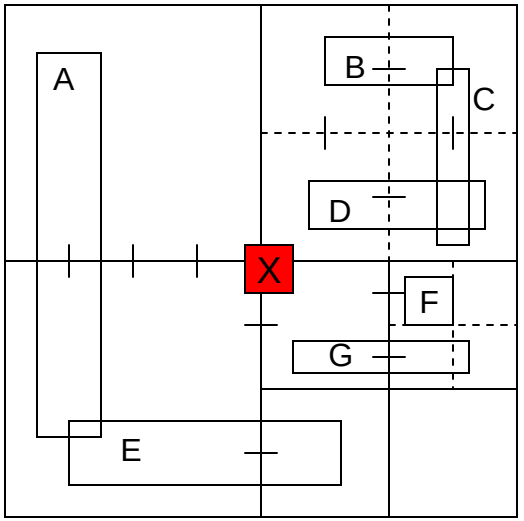


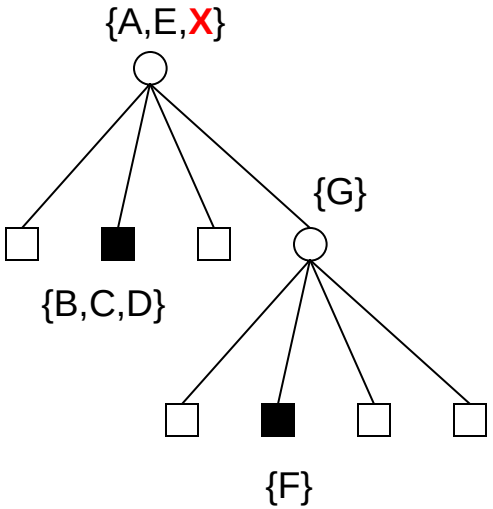


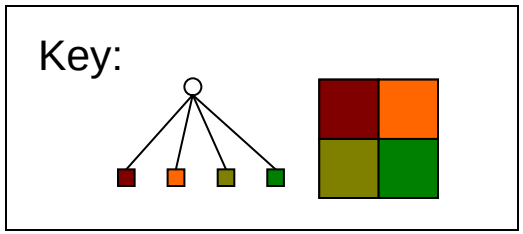


Problems

- Large blocks
- “Orphan” nodes
  - Small nodes covering several blocks
  - Artifacts of (arbitrary) choice of origin
  - Every query will include x in its results







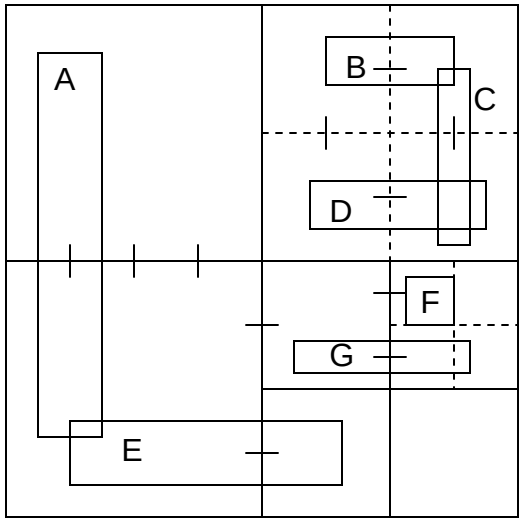
Loose Quadtree/Cover Fieldtree

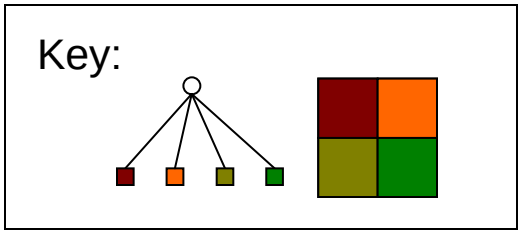
Frank and Barrera, Ulrich

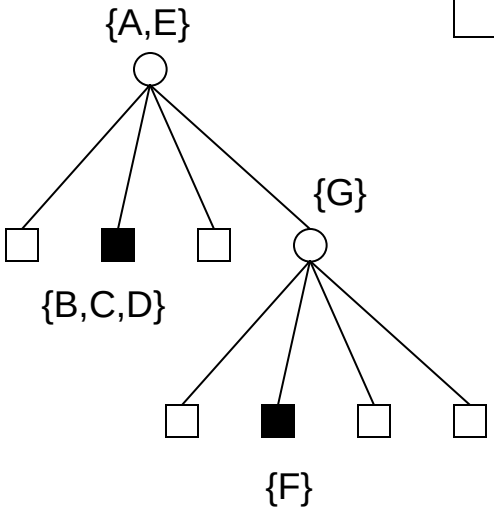
- Problem: For object o, the min bounding box is not related to size of o
- Uniformly expand size of space spanned by each block, *b*, of width, *w*, by a positive factor, *p*
  - Object associated with the min. enclosing expanded block
  - Expanded width of block, b = (1+p)*w
- Each object still covered by only one block
  - Similar to QMAT (quadtree medial axis transform)

Key result ➔ radius of min bounding box of object in b is larger than *p*w/4*

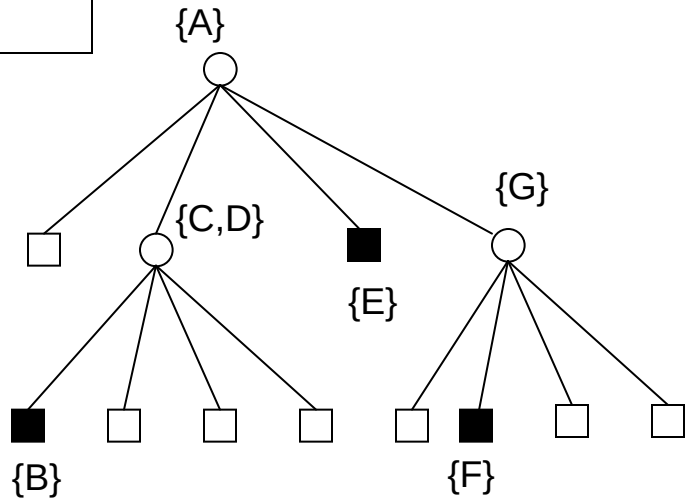
MX-CIF vs. Loose Quadtree







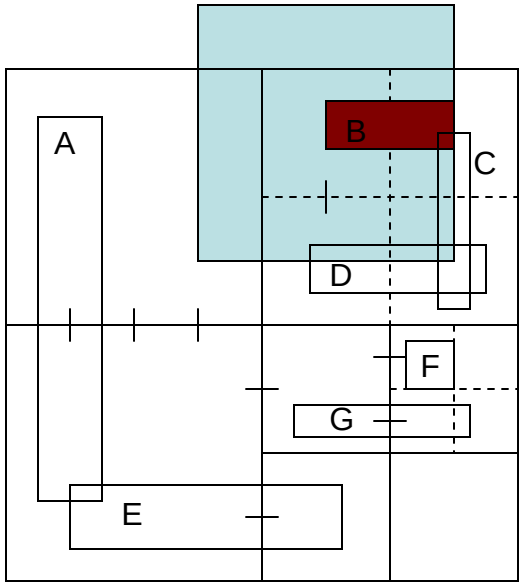
MX-CIF

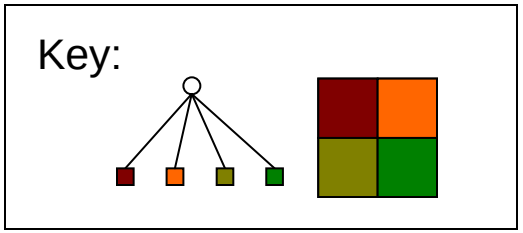


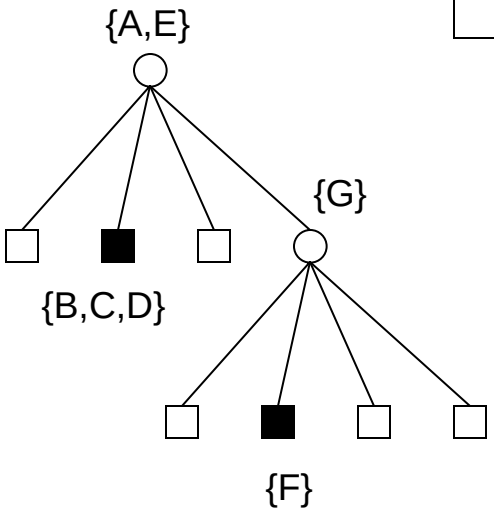
Loose Quadtree
p=1, width = (1+p)w = 2w

MX-CIF vs. Loose Quadtree

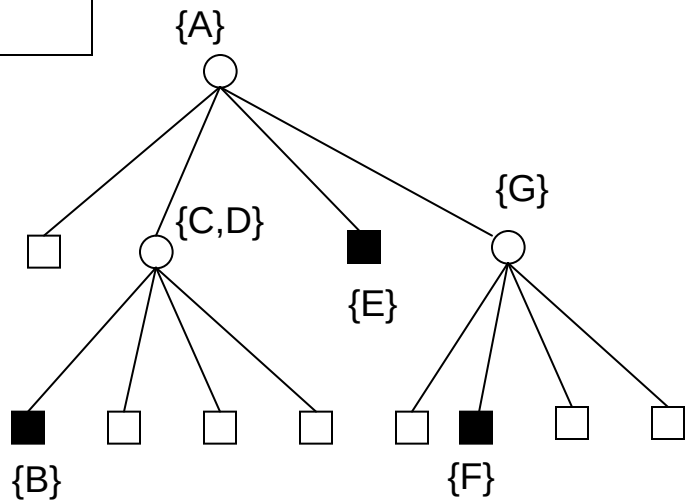
B is completely enclosed by this block





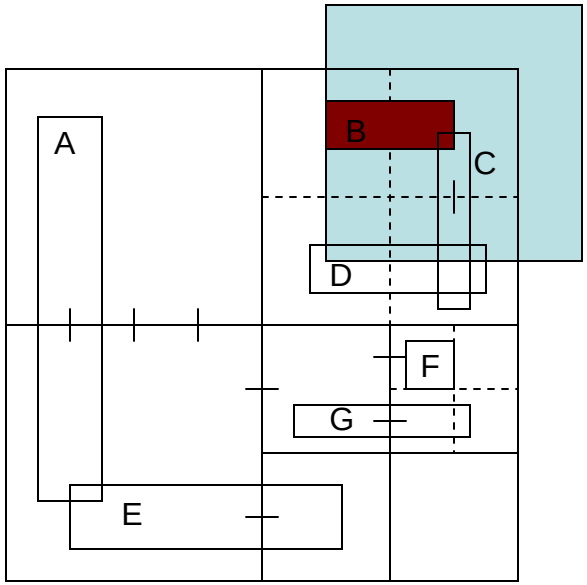


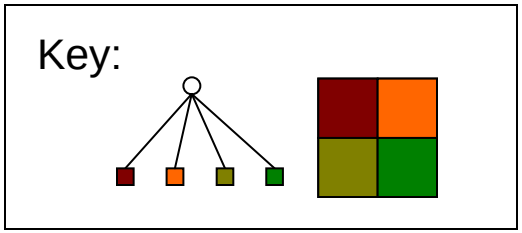
MX-CIF

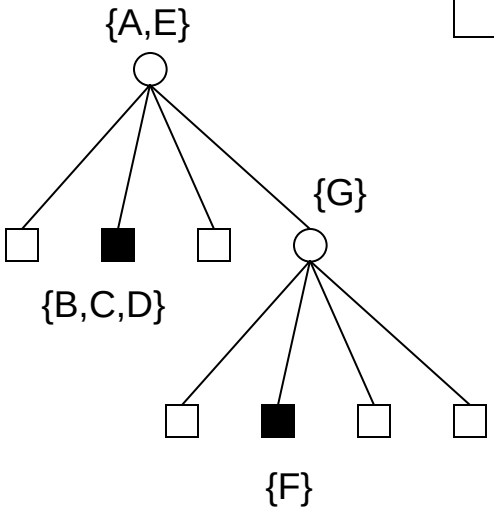


Loose Quadtree
p=1, width = (1+p)w = 2w

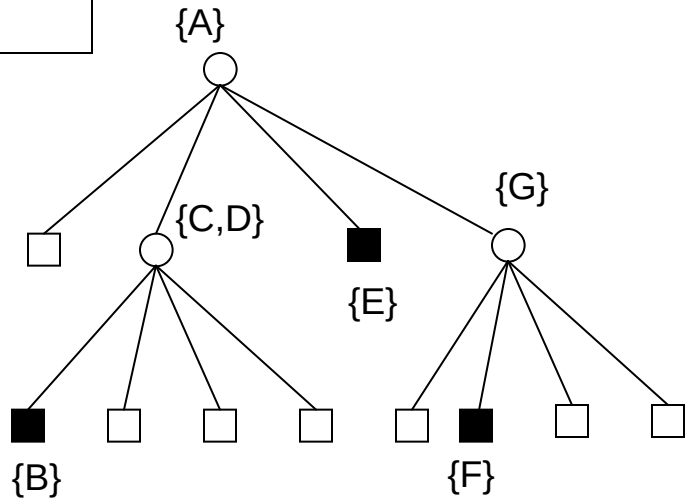
MX-CIF vs. Loose Quadtree





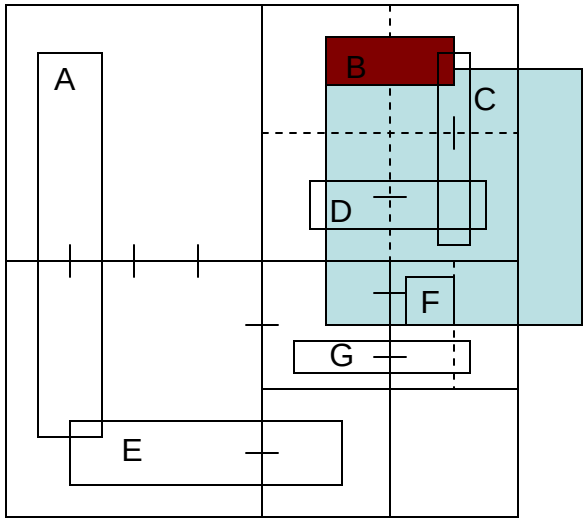


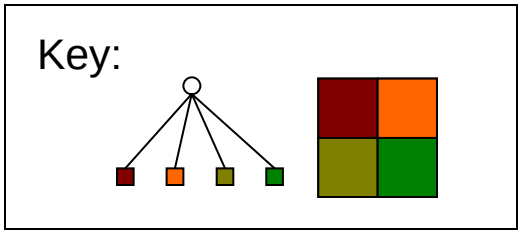
MX-CIF

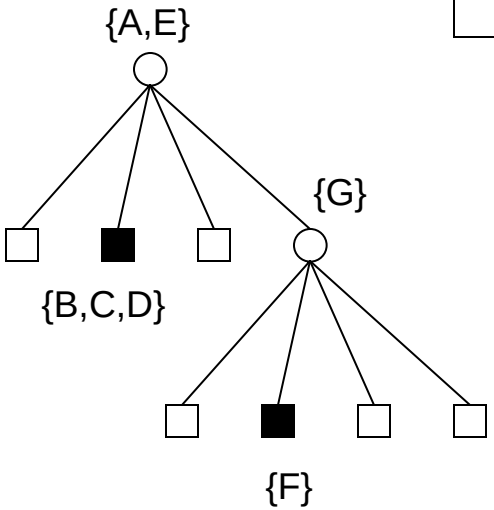


Loose Quadtree
p=1, width = (1+p)w = 2w

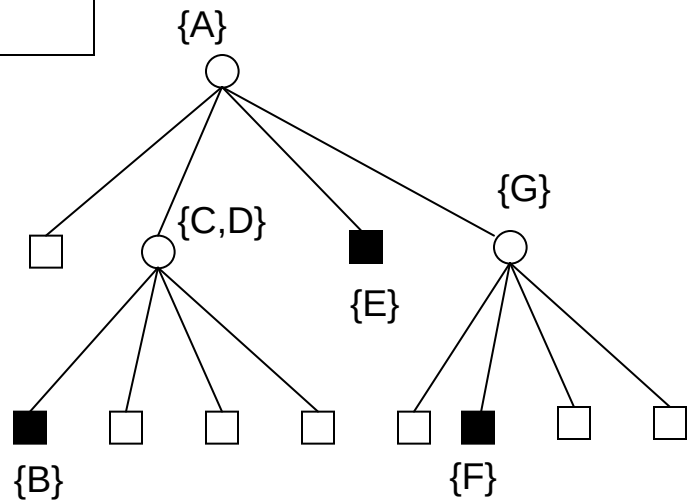
MX-CIF vs. Loose Quadtree





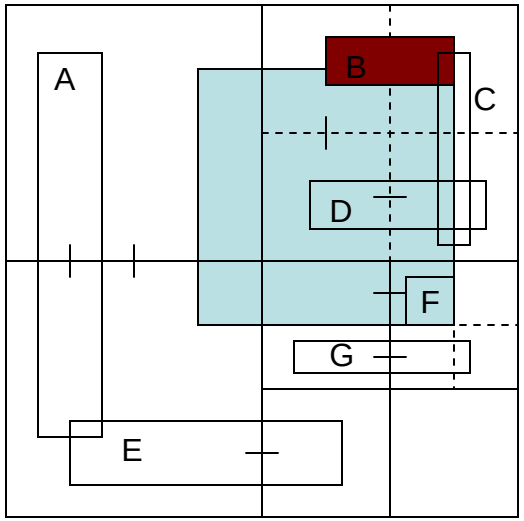


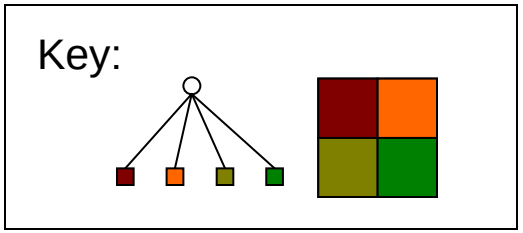
MX-CIF

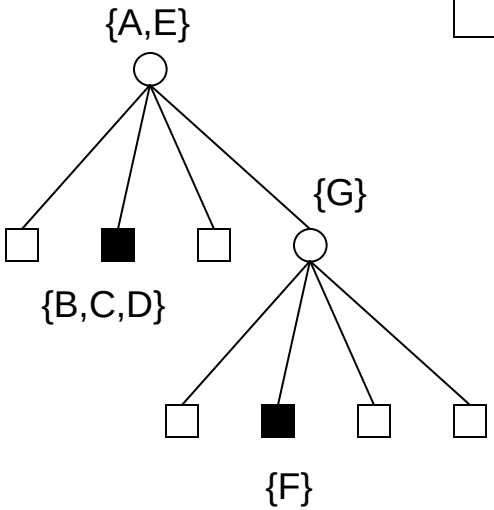


Loose Quadtree
p=1, width = (1+p)w = 2w

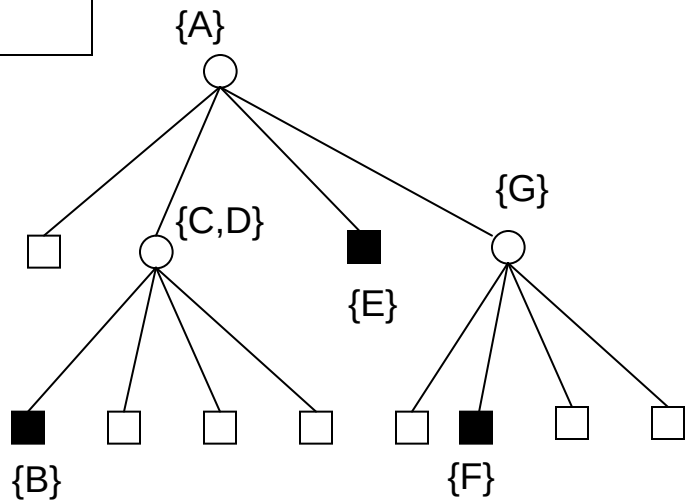
MX-CIF vs. Loose Quadtree







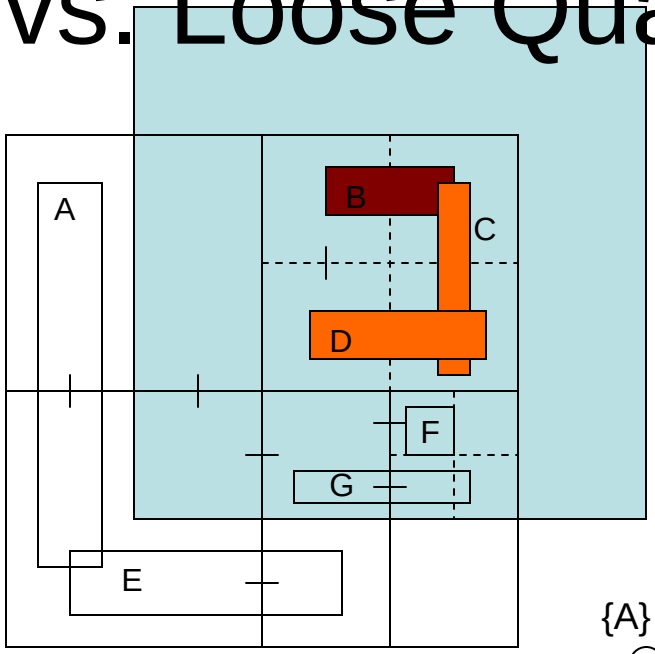
MX-CIF

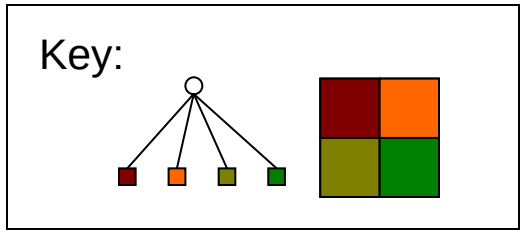


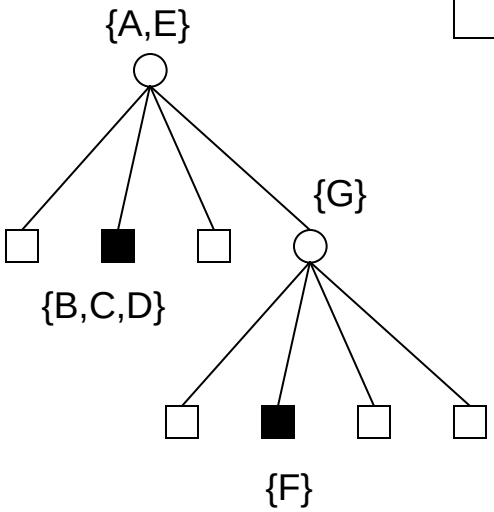
Loose Quadtree
p=1, width = (1+p)w = 2w

MX-CIF vs. Loose Quadtree

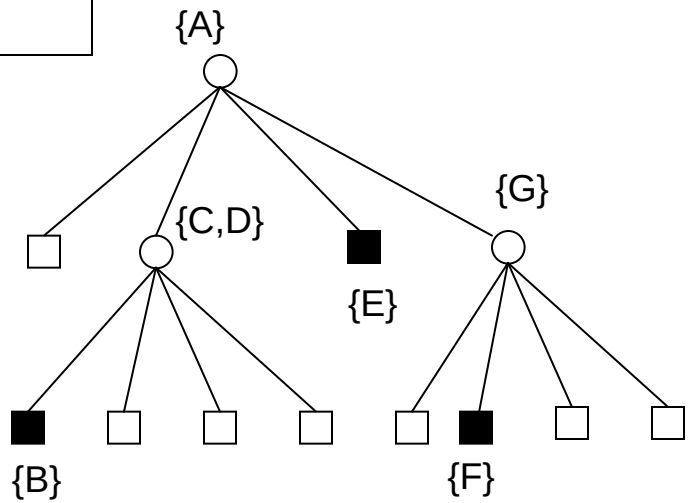
C and D are contained at this level





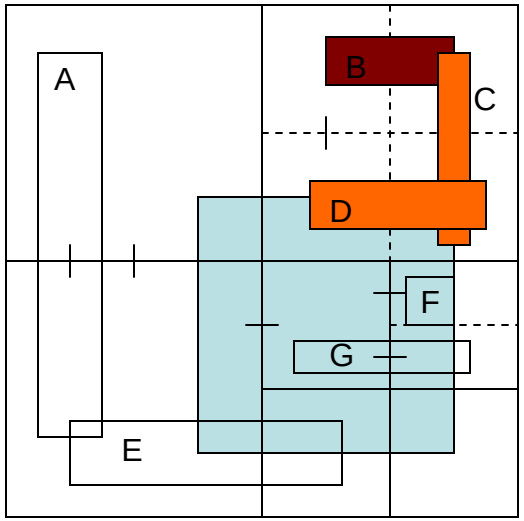


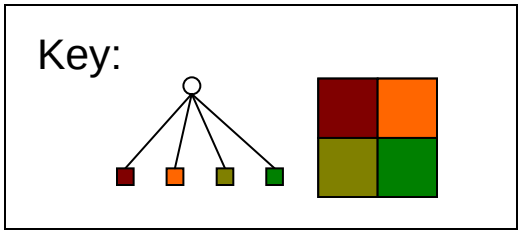
MX-CIF

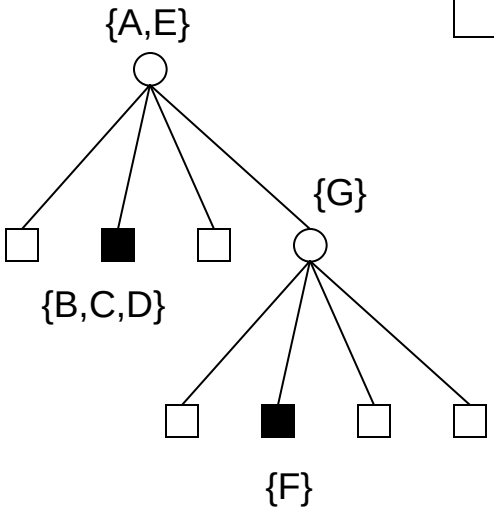


Loose Quadtree
p=1, width = (1+p)w = 2w

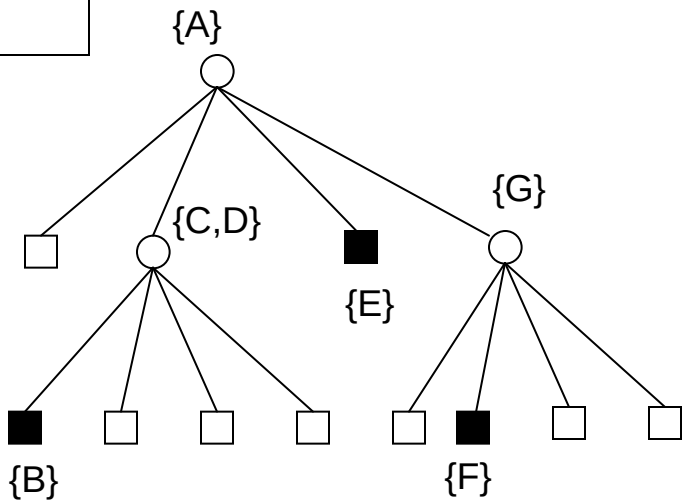
MX-CIF vs. Loose Quadtree





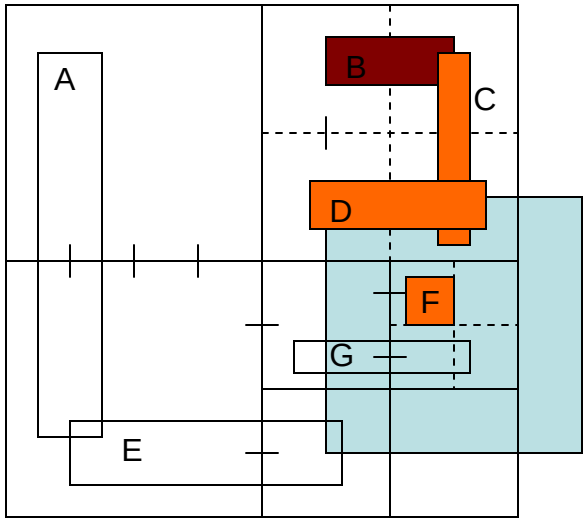


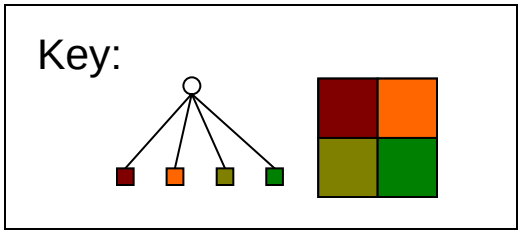
MX-CIF

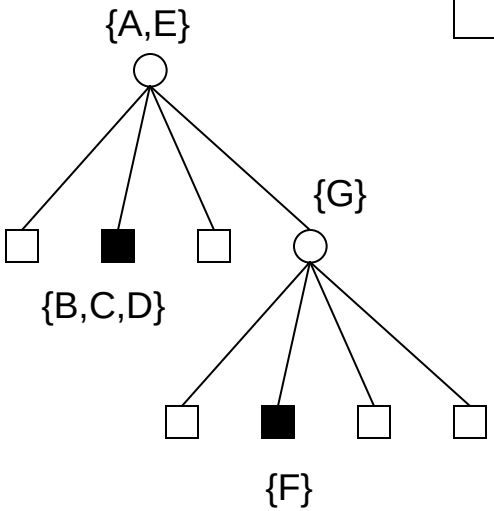


Loose Quadtree
p=1, width = (1+p)w = 2w

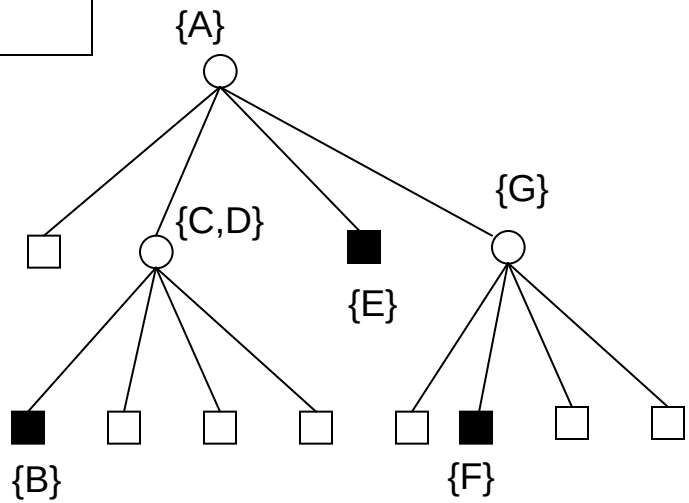
MX-CIF vs. Loose Quadtree







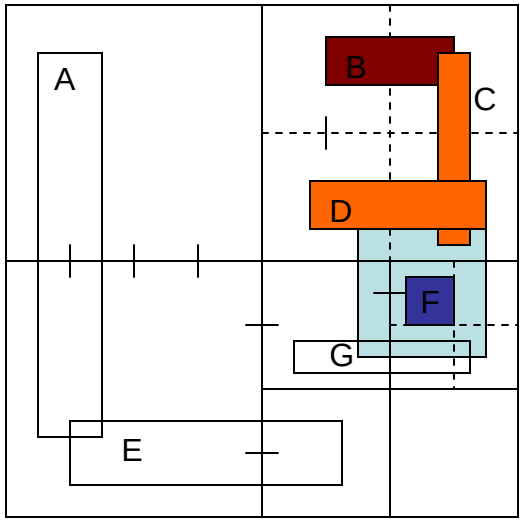
MX-CIF

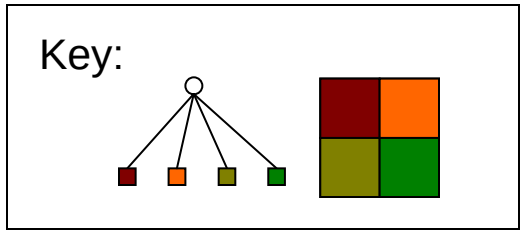


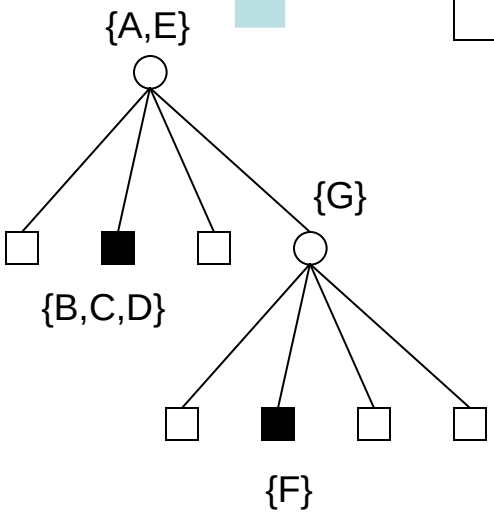
Loose Quadtree
p=1, width = (1+p)w = 2w

MX-CIF vs. Loose Quadtree

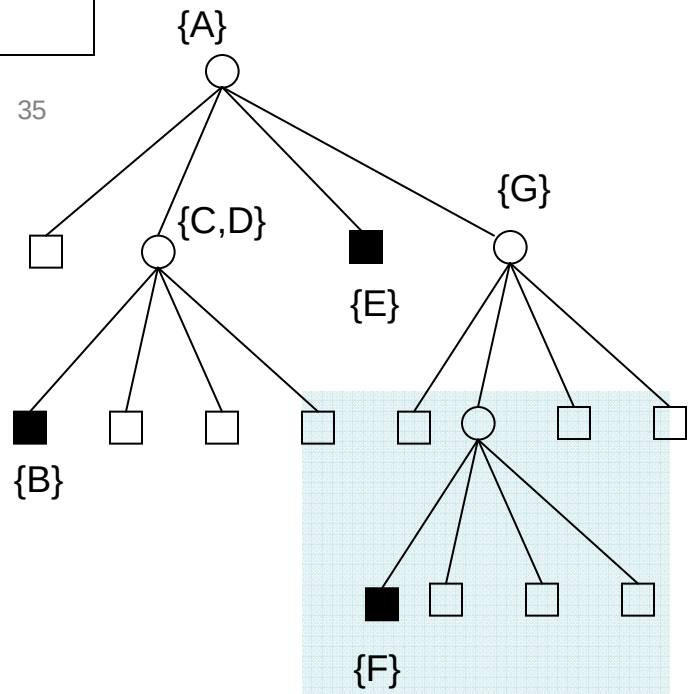
F = {(31,15), (35,19)}
Window =
{(27.5, 12.5), (37.5,22.5)}



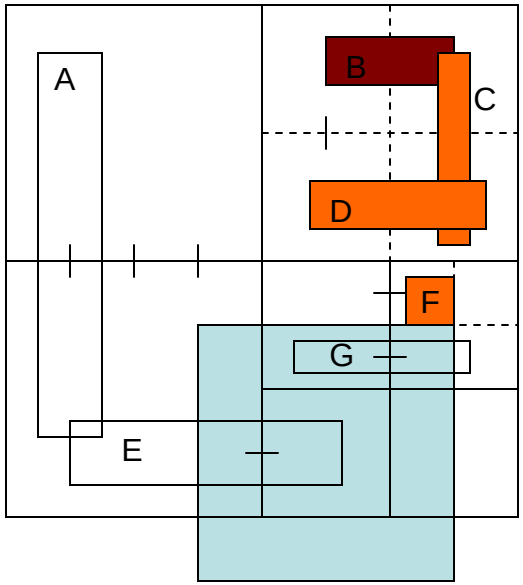


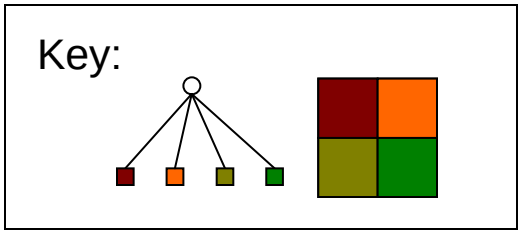


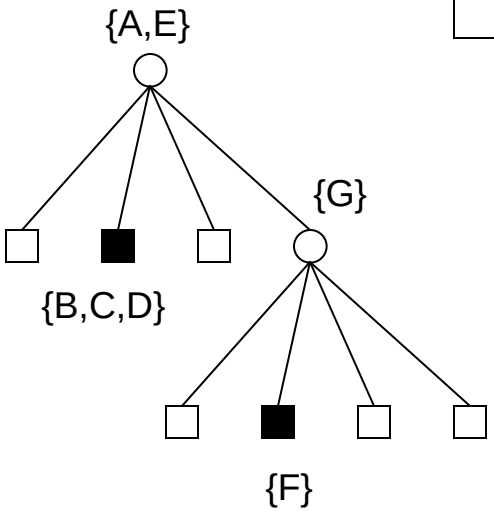
MX-CIF



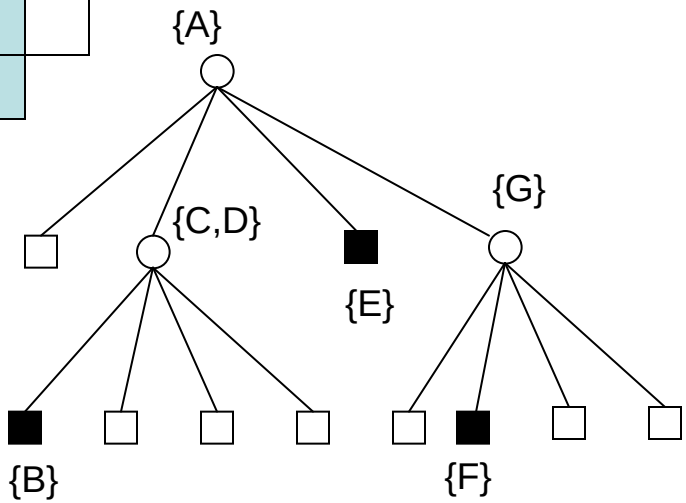
MX-CIF vs. Loose Quadtree





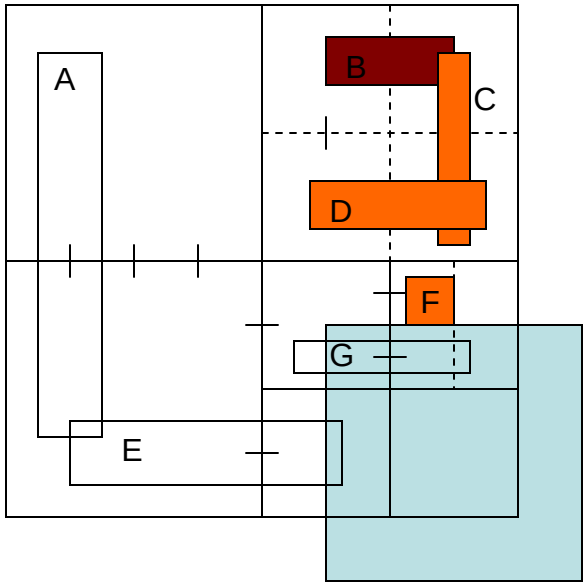


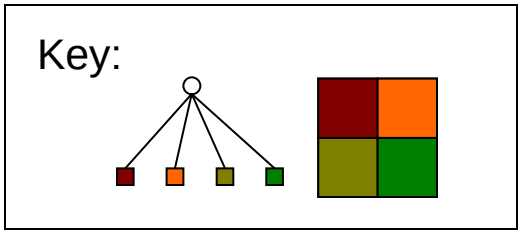
MX-CIF

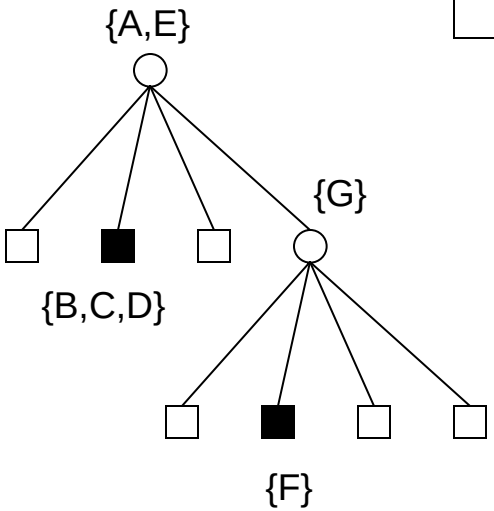


Loose Quadtree
p=1, width = (1+p)w = 2w

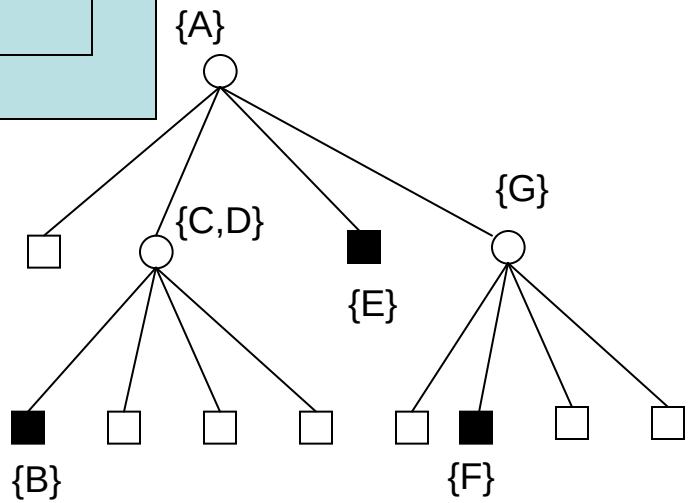
MX-CIF vs. Loose Quadtree





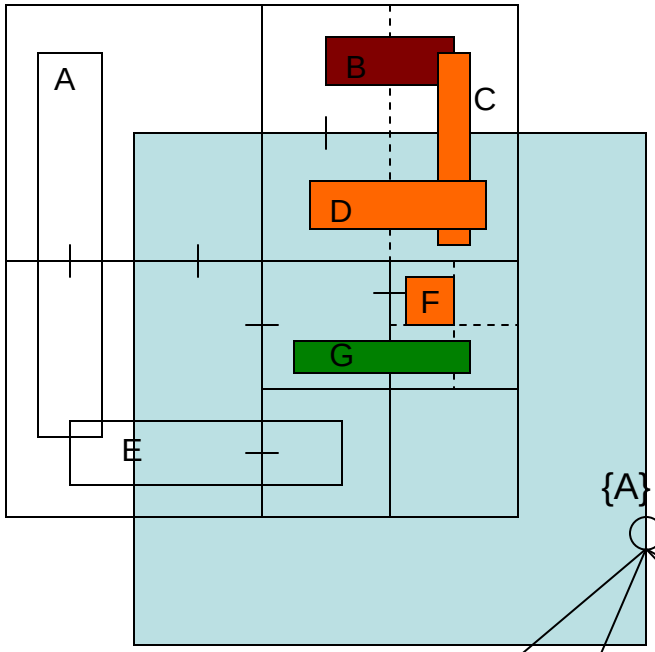


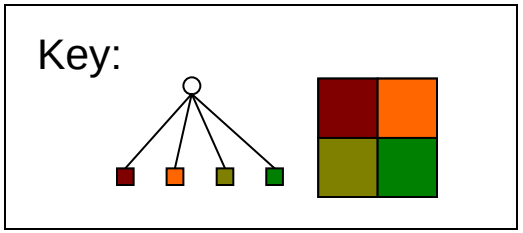
MX-CIF

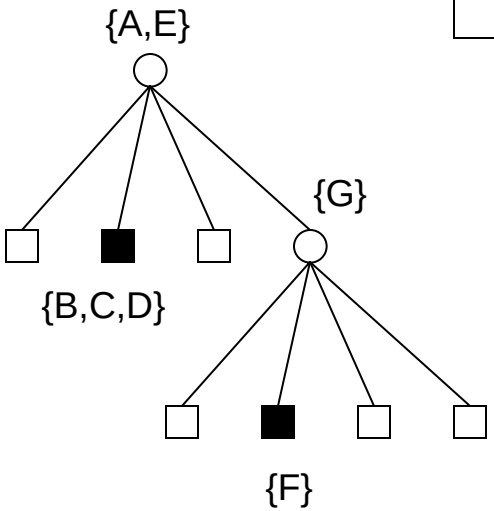


Loose Quadtree
p=1, width = (1+p)w = 2w

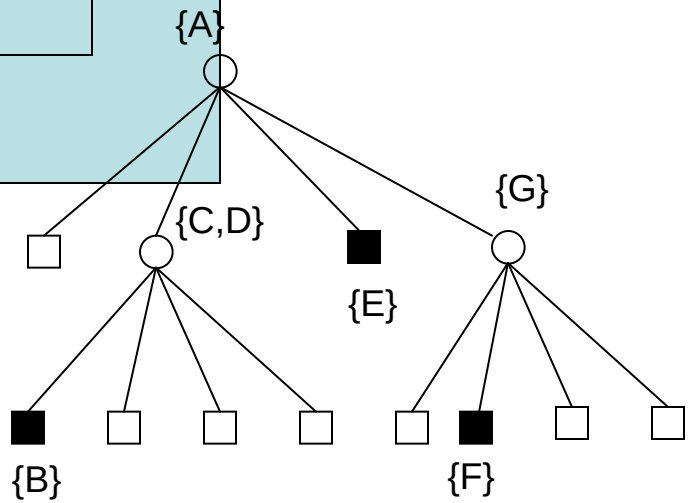
MX-CIF vs. Loose Quadtree





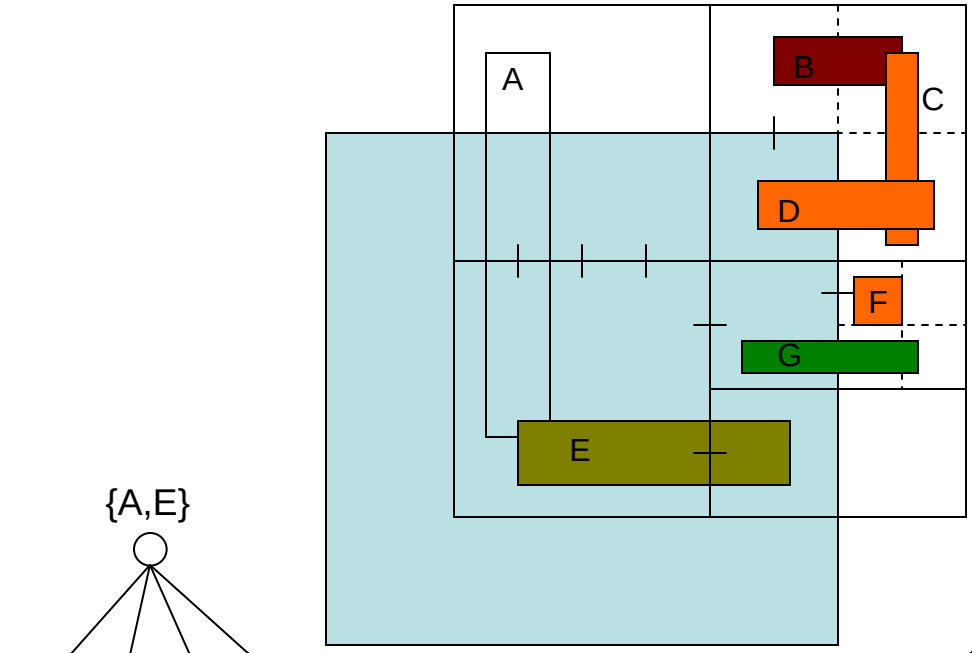


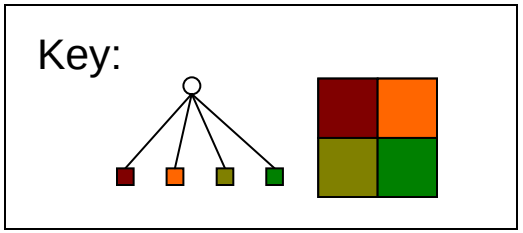
MX-CIF

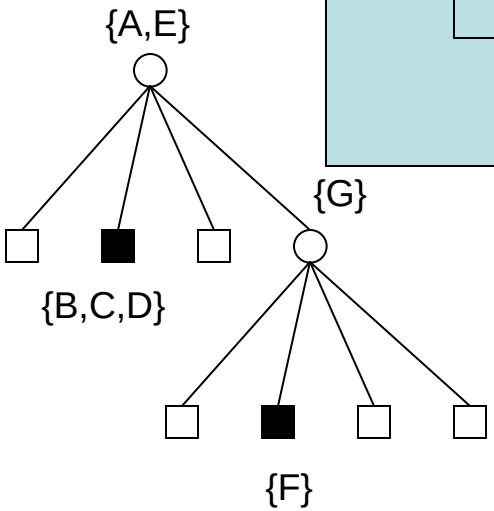


Loose Quadtree
p=1, width = (1+p)w = 2w

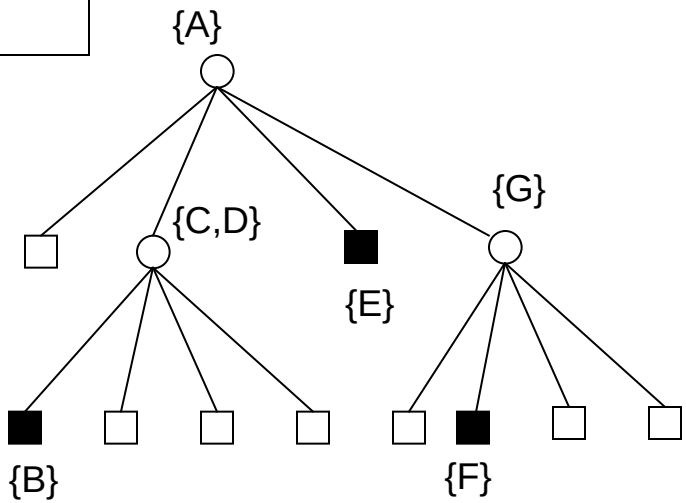
MX-CIF vs. Loose Quadtree





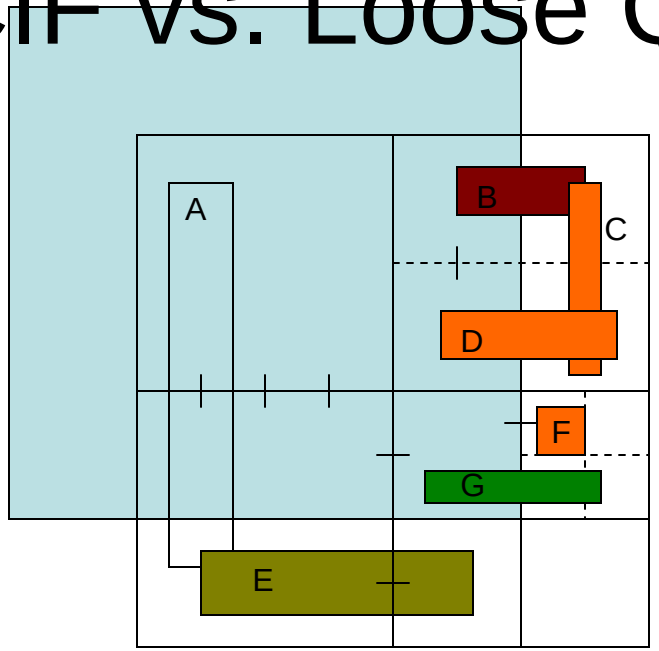


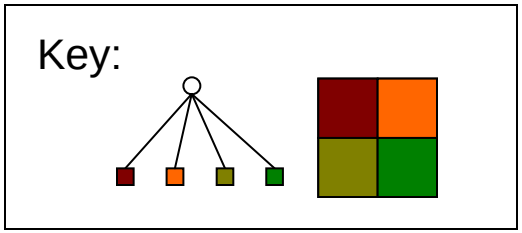
MX-CIF

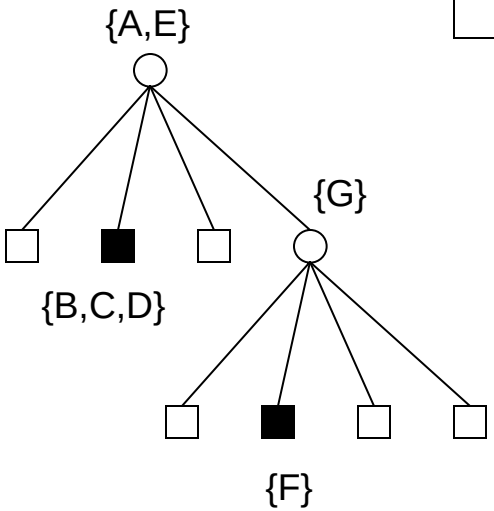


Loose Quadtree
p=1, width = (1+p)w = 2w

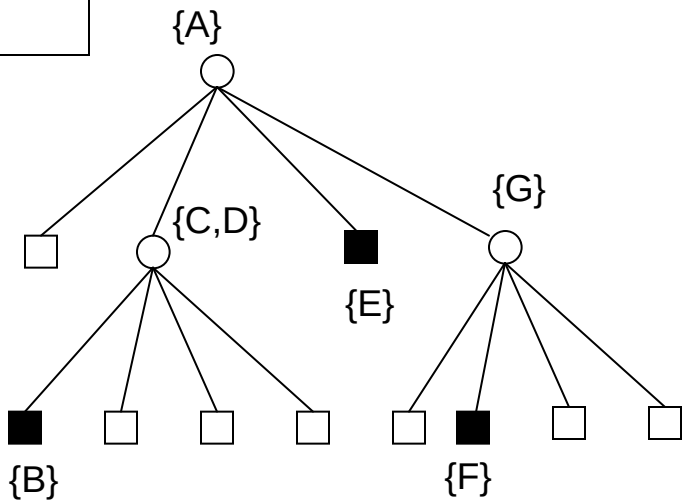
MX-CIF vs. Loose Quadtree







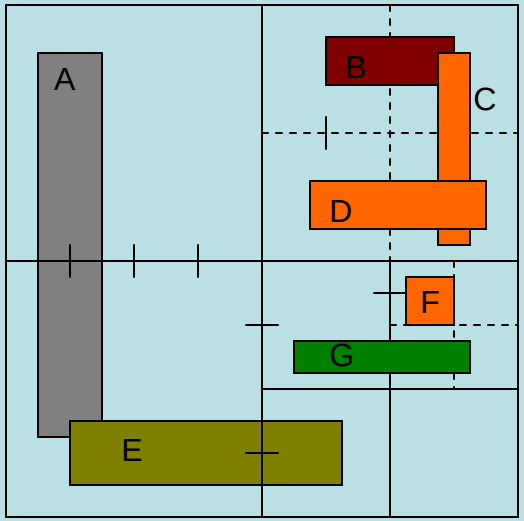
MX-CIF

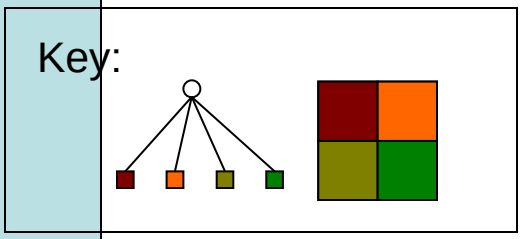


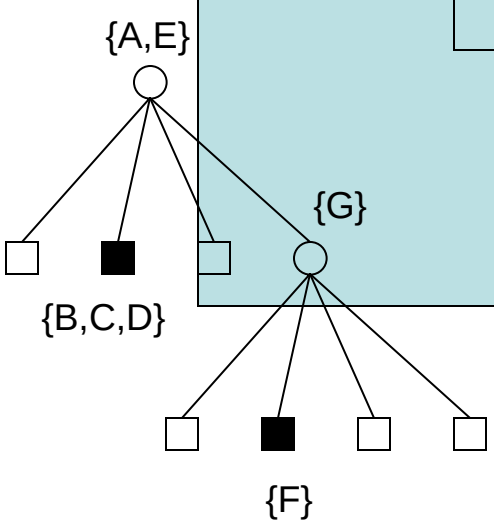
Loose Quadtree
p=1, width = (1+p)w = 2w

MX-CIF vs. Loose Quadtree

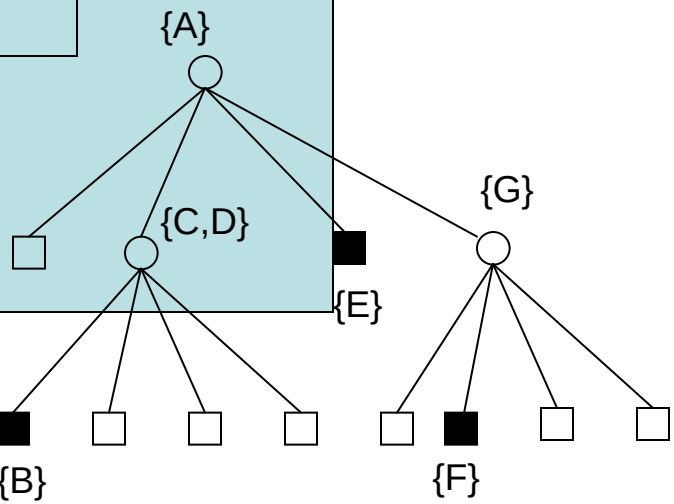
A fits into the top hierarchy





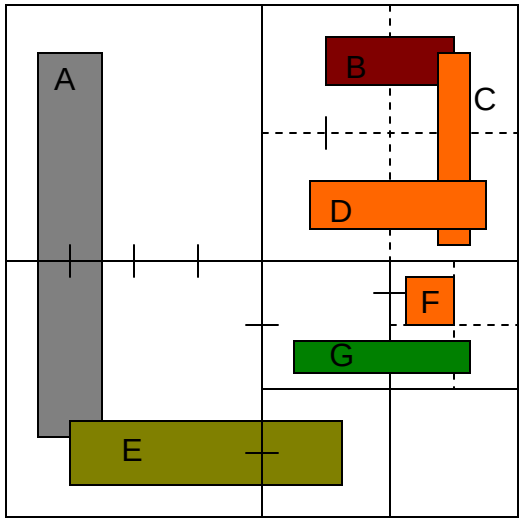


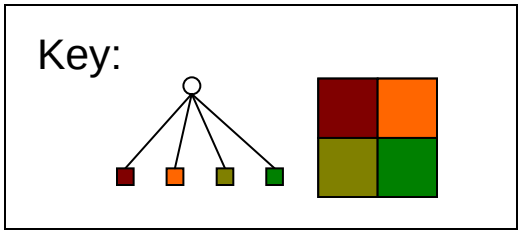
MX-CIF

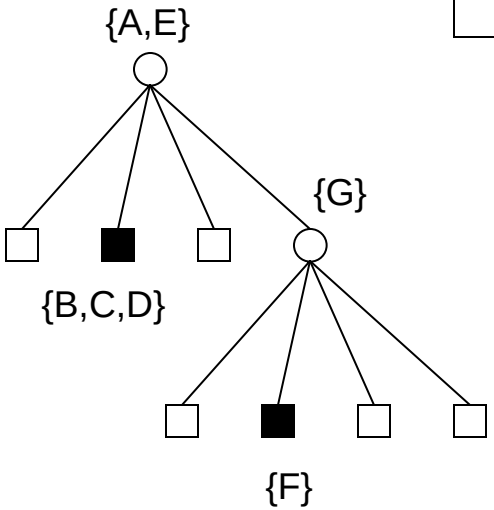


Loose Quadtree
p=1, width = (1+p)w = 2w

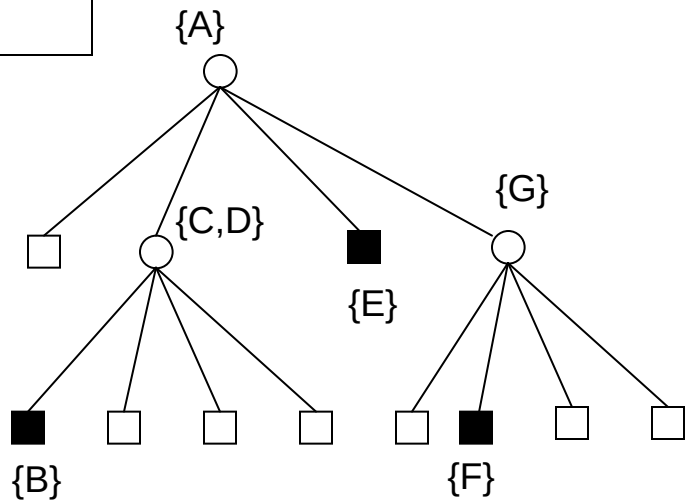
MX-CIF vs. Loose Quadtree







MX-CIF



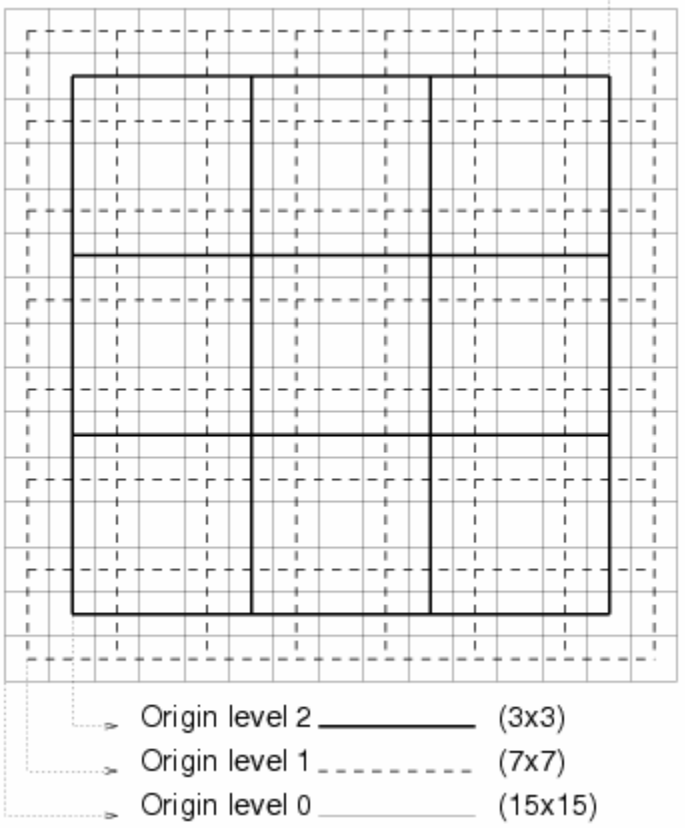
Loose Quadtree
p=1, width = (1+p)w = 2w

Partition Fieldtree

- Hierarchy of grids whose registration are shifted
  - Each cell is called a *field*
  - Each level forms a non-overlapping partitioning of the space
  - For each block, *b*, of width, *w*, that is being subdivided
    - origins are shifted by ***w/2***
- A new node is stored in the smallest field in which it completely fits
  - Object never has to be stored more than three levels above its proper size

Partition Fieldtree
Properties

- Boundaries of blocks at different levels will never coincide
- Grids at different levels have a different origin
- Blocks at different levels do not form a refinement of those at a previous level



Source: www.gdmc.nl/oosterom/slc.ps

Bound on size of enclosure

- Partition
  - Always bounded by three levels
    - 8x object size
- Cover
  - Radius must be larger than p*w/4
    - As p decreases, the minimum radius decreases
      - p = 1/2, ratio is at most 4
      - p = 1/4, ratio is at most 8
      - P = 1/8, ratio is at most 16 → tighter than partition

Comparison

- Goal of both
  - Expand the area spanned by the subblocks to reduce size of minimum enclosing quadtree block
    - When an object overlaps the axes that pass through the center of the block
- Cover
  - Area spanned by four subblocks is expanded
- Partition
  - Number of subblocks is enlarged by offsetting their position while retaining their size

Comparison (2)

- Subblocks span an area that overlaps the partition lines
  - Always for partition fieldtree
  - Not always for Cover fieldtree
    - some values of p will have partition lines coincide
      - i.e. p =1, powers of 2
  - Differs from regular quadtree, where successive levels are collinear

References

- Foundations of Multidimensional and Metric Data Structures
  Samet, Hanan
  to appear in 2005.

- A Survey Of Hierarchical Partitioning Methods For Vector Images,
  Noronha, Valerian T
  dgrc.ca/publicns/syd8808/syd8808.pdf

- The Spatial Location Code
  van Oosterom, Peter
  www.gdmc.nl/oosterom/slc.ps