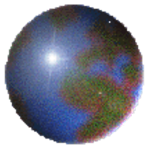
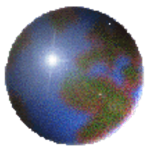


CSEG601 & CSE5601:

Spatial Data Management & Applications

Sungwon Jung

Big Data Processing & DB Lab
Dept. of Computer Science and Engineering
Sogang University
Seoul, Korea
Tel: +82-2-705-8930
Email : jungsung@sogang.ac.kr

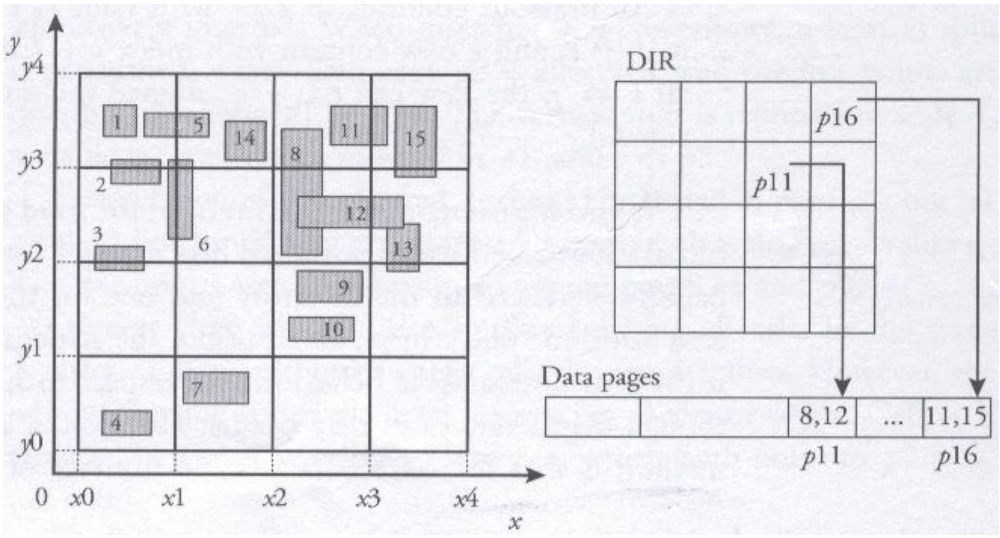


Spatial Access Methods 2

Space-Driven Structure: The Grid File

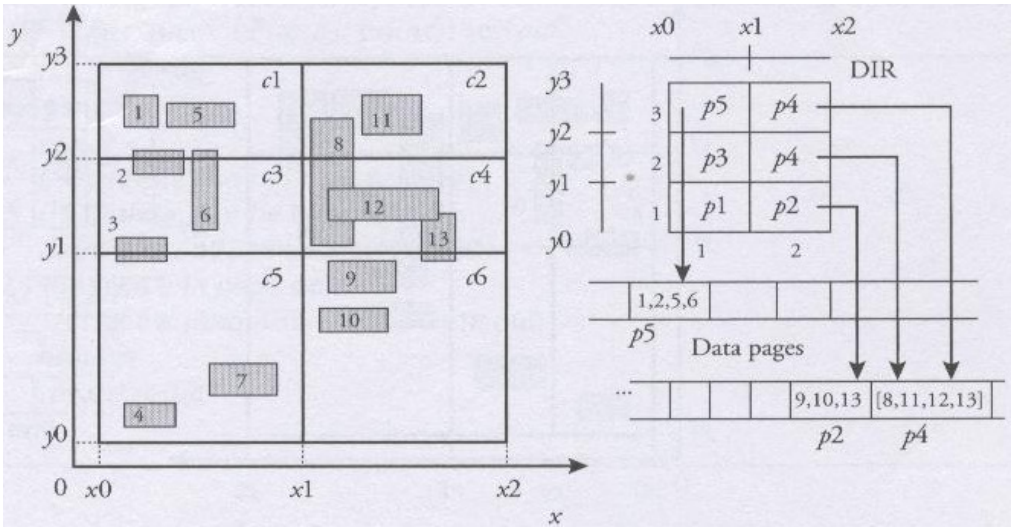
Space-Driven Structure: The Grid File

- Rectangle indexing with Grids
  - A fixed grid for rectangle indexing



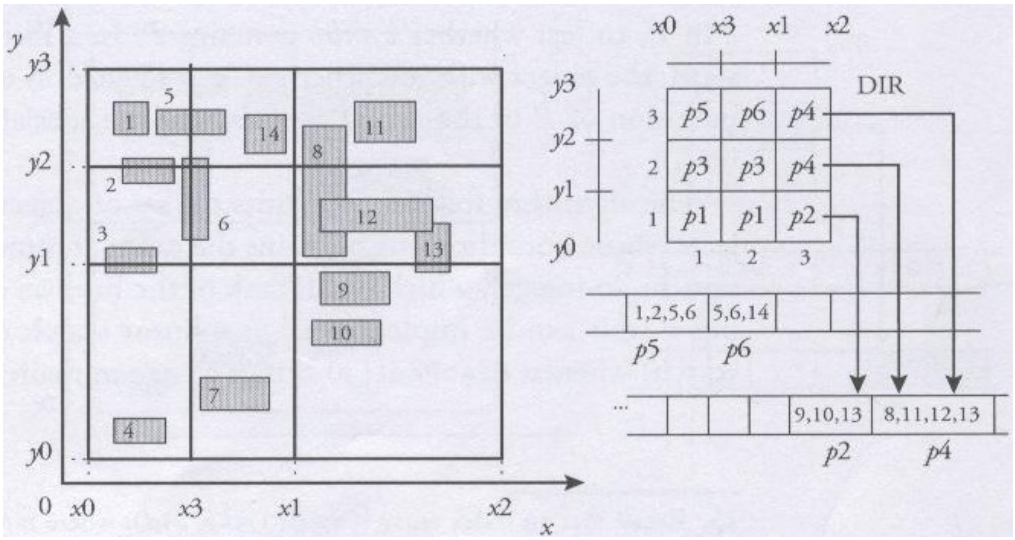
Space-Driven Structure: The Grid File

- Rectangle indexing with Grids
  - A grid file for rectangle indexing



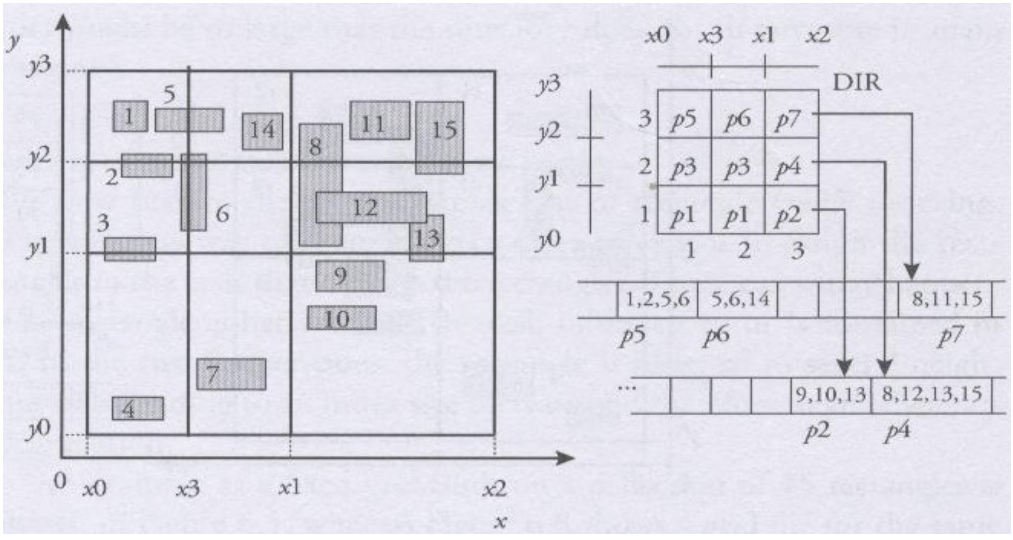
Space-Driven Structure: The Grid File

- Rectangle indexing with Grids
  - Insertion of object 14: Cell Split and Directory Split



Space-Driven Structure: The Grid File

- Rectangle indexing with Grids
  - Insertion of object 15: Cell Split without Directory Split



Space-Driven Structure: The Grid File

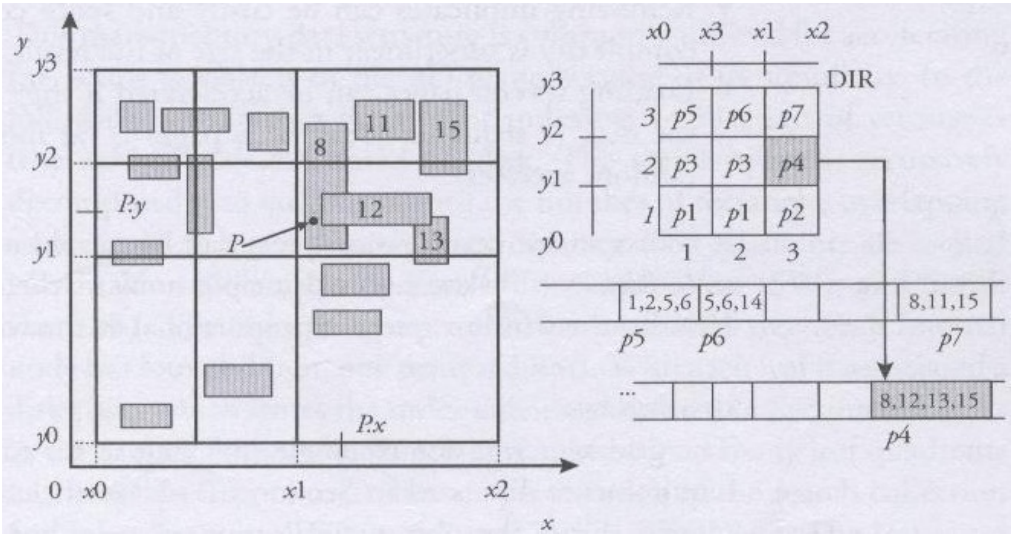
- Point query algorithm with the grid file

```
GF-POINTQUERY (P(a, b): point): set(oid)

begin
   result = ∅
   // Compute the cell containing P
   i = RANK(a, Sx); j = RANK(b, Sy)
   page = READPAGE (DIR[i,j].PageID)
   for each e in page do
      if (P ∈ e.mbb) then result += {e.oid}
   end for
   return result
end
```


Space-Driven Structure: The Grid File

- An example of point query with grid file
  - Cell DIR[3, 2] is retrieved



Space-Driven Structure: The Grid File

- Window query algorithm with grid file
  1. Compute all cells that overlap the window argument
  2. Each of the overlapped cells is scanned as in the point query algorithm
  3. Because the result is likely to have duplicates, they need to be removed
     1. Done by sorting
  - Removing duplicates can be costly and space consuming
    - Time complexity is superlinear in the size of the result
  - Loading several pages can be accelerated if they are consecutive on the disk
    - The # of I/Os is proportional to the window area

```
GF-WINDOWQUERY (W(x1, y1, x2, y2): rectangle): set(oid)

begin
  result = ∅
  // Compute the low-left cell intersecting W
  i1 = RANK (x1, Sx); j1 = RANK (y1, Sy)
  // Compute the top-right cell intersecting W
  i2 = RANK (x2, Sx); j2 = RANK (y2, Sy)
  // Scan the grid cells
  for (i = i1; i ≤ i2; i++) do
    for (j = j1; j ≤ j2; j++) do
      // Read the page, and test each entry
      page = READPAGE (DIR[i,j].PageID)
      for each (e in page) do
        if (W ∩ e.mbb ≠ ∅) then result += {e.oid}
      end for
    end for
  end for
  // Sort the result, and remove duplicates
  SORT (result); REMOVEDUPL (result);
  return result
end
```


Space-Driven Structure: The Grid File

- Discussion
  - The object duplication in neighbor cells increases the # of entries in the index, and therefore the index size
    - This effect becomes more and more significant as the data size increases and as the cell size decreases down to the *mbb*'s size
  - Removing duplicates from the result is expensive when the result size is large
  - The efficiency of the SAM relies on the assumption that the directory is resident in central memory
    - Impossible for very large data sets
    - If the directory has to be partly on disk, its management becomes complex and the response time is degraded
      - Supplementary I/Os are necessary to access the directory