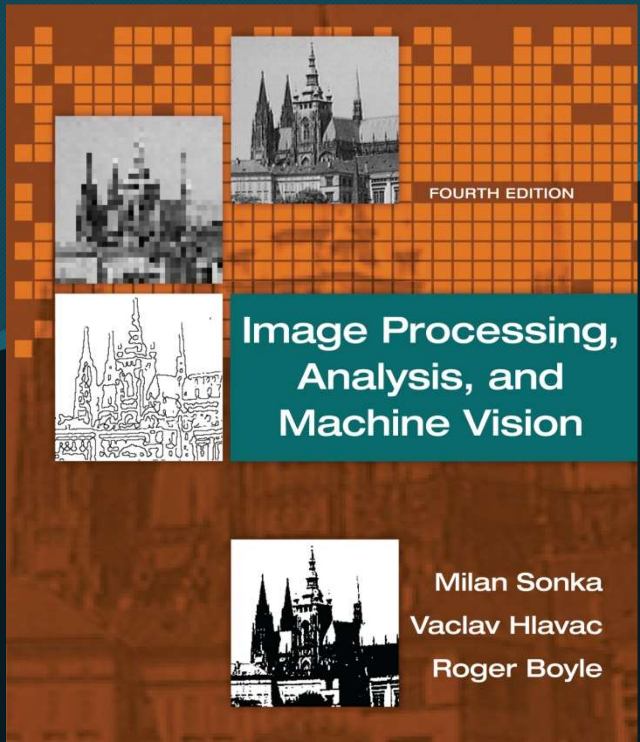
Chapter 4

Data structure for image analysis



Levels of image data representation

- The representation can be stratified in four levels [Ballard and Brown, 1982]
  - The lowest representational level: iconic (符號的) images
    - This level consists of image containing original data.
  - The second level: segmented images
    - Parts of images are joined into groups that probably belong to the same objects.
  - The third level: geometric representations
    - These representations hold knowledge about 2D and 3D shapes.
  - The fourth level: relational models
    - These models give us the ability to treat data more efficiently and at a higher level of abstraction.
  - There are no strict borders between these levels.
  - These four levels are ordered from signals at a low level of abstraction to the description that a human can perceive.

Traditional image data structures

- Matrices
  - The most common data structure for low-level representation of an image.
  - Image information in the matrix is accessible through the co-ordinates of a pixel that correspond with row and column indices.
  - The matrix is a full representation of the image, independent of the contents of image data.
  - Spatial relation
    - The space is two-dimensional in the case of an image.
    - One very natural spatial relation is the neighborhood relation.

Matrices

- Global information of images
  - Histogram is the most popular example of global information.
  - Co-occurrence matrix [Pavlidis, 1982] is another example of global information.
    - Given an image $f(i,j)$ and if pixel (i_1, j_1) has intensity z and pixel (i_2, j_2) has intensity y, then the co-occurrence matrix of $f(i,j)$ can be obtained from algorithm 4.1.

Algorithm 4.1 Co-occurrence matrix $C_r(z, y)$ for the relation r

1. Set $C_r(z, y) = 0$ for all $z, y \in [0, L]$, where L is the maximum brightness.
2. For all pixels (i_1, j_1) in the image, determine all (i_2, j_2) which have the relation r with the pixel (i_1, j_1), and perform

$$C_r[f(i_1, j_1), f(i_2, j_2)] = C_r[f(i_1, j_1), f(i_2, j_2)] + 1$$

Matrices

- An example of Algorithm 4.1
  - If the relation r is *to be a southern or eastern 4-neighbor of the pixel (i_1, j_1), or identity* (see p. 103) , elements of the co-occurrence matrix have some interesting properties.
    - Diagonal elements of the matrix $C_r(k,k)$ are equal to the area of the regions in the image with brightness k, and so correspond to the histogram.
    - Off-diagonal elements $C_r(k,j)$ are equal to the length of the border dividing regions with brightnesses k and j, $k \neq j$.
  - For instance,
    - In an image with low contrast, the elements of the co-occurrence matrix that are far from the diagonal are equal to zero or are very small.
    - For high-contrast images, the opposite is true.

An example of Algorithm 4.1

- Relation r: *a southern or eastern 4-neighbor of the pixel (i_1, j_1), or identity* (see p. 103)
  - Diagonal elements:
    the area of the regions in the image with brightness k (histogram)
  - Off-diagonal elements:
    the length of the border dividing regions with brightnesses k and j, $k \neq j$.

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

Image

$$\begin{array}{c c} & \begin{matrix} 0 & 1 & 2 \end{matrix} \\ \begin{matrix} 0 \\ 1 \\ 2 \end{matrix} & \begin{bmatrix} 1 & 1 & 0 \\ 2 & 8 & 5 \\ 0 & 6 & 16 \end{bmatrix} \end{array}$$

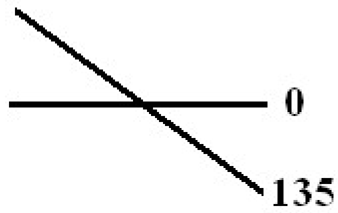
Co-occurrence matrix

Another example: $\boldsymbol{r}$ = (orientation, distance)

Image:

$$\begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 2 & 2 & 2 \\ 2 & 2 & 3 & 3 \end{bmatrix}$$

$$\boldsymbol{r} = (0,\ 1), \qquad C_{(0,\,1)} = \begin{array}{c c} & \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 4 & 2 & 1 & 0 \\ 2 & 4 & 0 & 0 \\ 1 & 0 & 6 & 1 \\ 0 & 0 & 1 & 2 \end{bmatrix} \end{array}$$



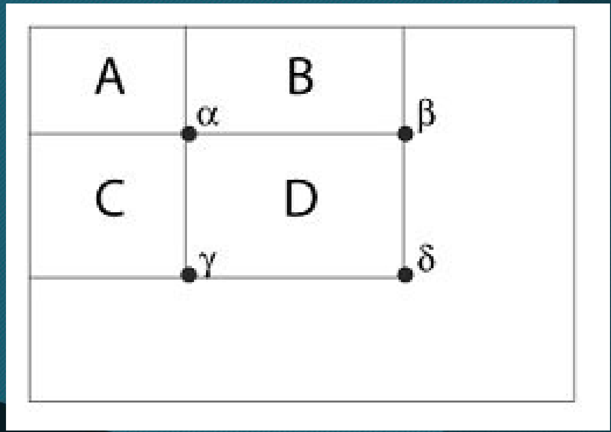
$$\boldsymbol{r} = (135,\ 1), \quad C_{(135,\,1)} = \begin{array}{c c} & \begin{array}{cccc} 0 & 1 & 2 & 3 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \\ 3 \end{array} & \begin{bmatrix} 2 & 1 & 3 & 0 \\ 1 & 2 & 1 & 0 \\ 3 & 1 & 0 & 2 \\ 0 & 0 & 2 & 0 \end{bmatrix} \end{array}$$

$\boldsymbol{r}$ = (orientation, distance)

Matrices

- The integral image
  - Another matrix representation that holds global information.
  - Its values $ii(i,j)$ in the location (i,j) represent the sums of all the original image pixel values left of the above (i,j).
  - Given the original image f

$$ii(i,j) = \sum_{k \le i; l \le j} f(k,l)$$



Matrices

Algorithm 4.2 Integral image construction

1. Let $s(i, j)$ denote a cumulative row sum, and set $s(i, -1) = 0$.
2. Let $ii(i, j)$ be an integral image, and set $ii(-1, j) = 0$.
3. Make a single row-by-row pass through the image.

   For each pixel (i, j) calculate the cumulative row sums $s(i, j)$ and the integral image value $ii(i, j)$

   $$s(i, j) = s(i, j - 1) + f(i, j)$$

   $$ii(i, j) = ii(i - 1, j) + s(i, j)$$

4. After completing a single pass through the image, the integral image ii is constructed.

1	2	2	4	1
3	4	1	5	2
2	3	3	2	4
4	1	5	4	6
6	3	2	1	3

input image

0	0	0	0	0	0
0	1	3	5	9	10
0	4	10	13	22	25
0	6	15	21	32	39
0	10	20	31	46	59
0	16	29	42	58	74

integral image

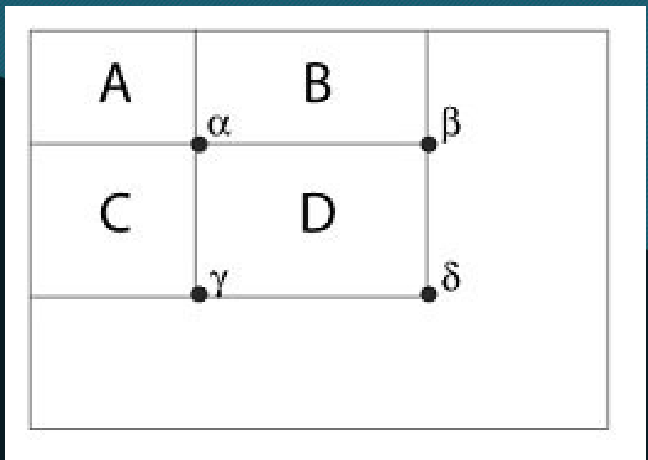
Matrices

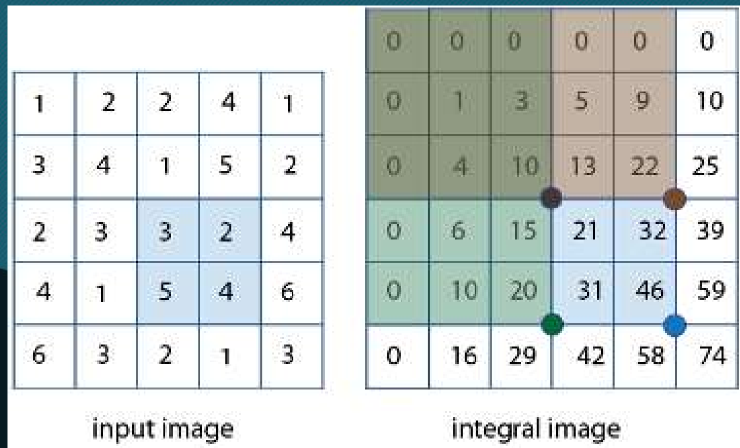
- Calculation of rectangle features from an integral image.
  - The sum of pixels within rectangle D can be obtained using four array references.

$$D_{sum} = ii(\delta) + ii(\alpha) - (ii(\beta) + ii(\gamma))$$

where $ii(\delta)$ is the value of the integral image at point δ.

  - For example, 46 +10 – (22 + 20) = 14 = 3 + 2 + 5 + 4





1	2	2	4	1
3	4	1	5	2
2	3	3	2	4
4	1	5	4	6
6	3	2	1	3

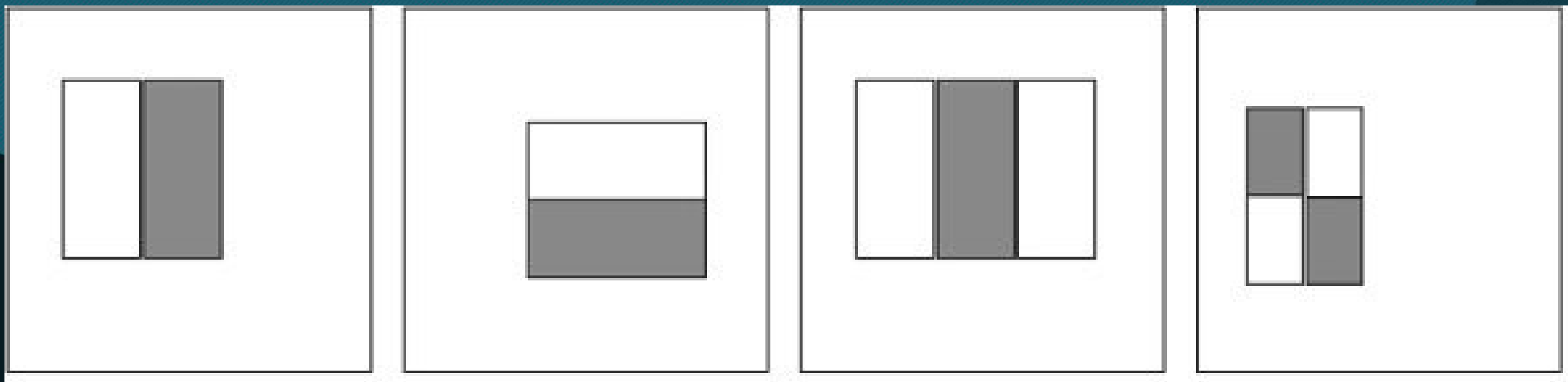
input image

0	0	0	0	0	0
0	1	3	5	9	10
0	4	10	13	22	25
0	6	15	21	32	39
0	10	20	31	46	59
0	16	29	42	58	74

integral image

Matrices

- Rectangle-based features may be calculated from an integral image by subtraction of the sum of the shaded rectangle(s) from the non-shaded rectangle(s).
- The figure shows (a, b) two-rectangle, (c) three-rectangle, and (d) four-rectangle features.



(a) (b) (c) (d)

Traditional image data structures

- Chains
  - Chains are used for the description of object borders in computer vision.
  - For example: chain codes (8-neighborhoods)

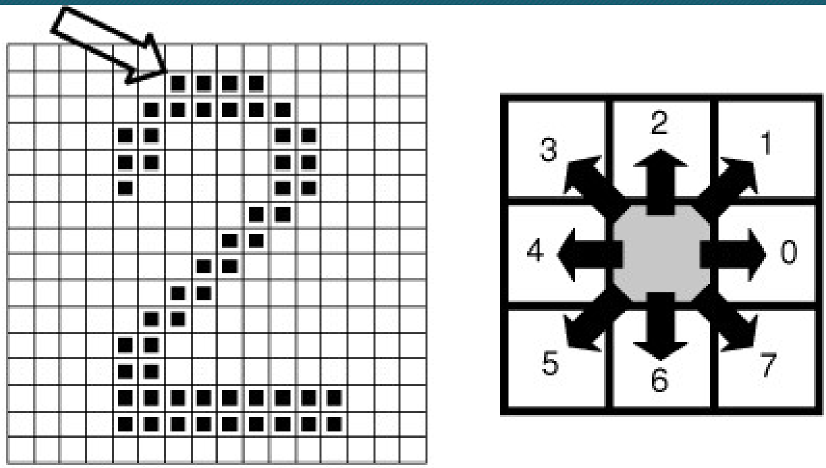
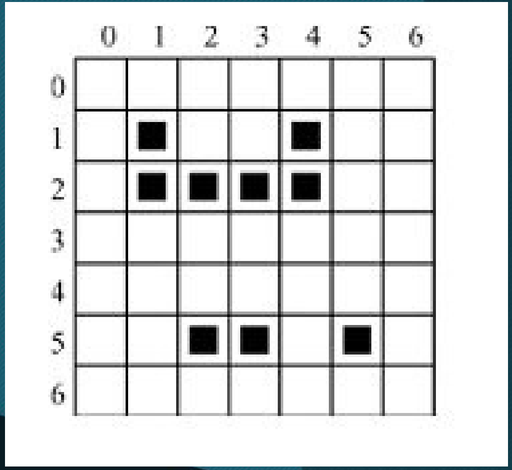


Figure 4.3: An example chain code; the reference pixel starting the chain is marked by an arrow: 00077665555556600000006444444442221111112234445652211. © *Cengage Learning 2015.*

Chains

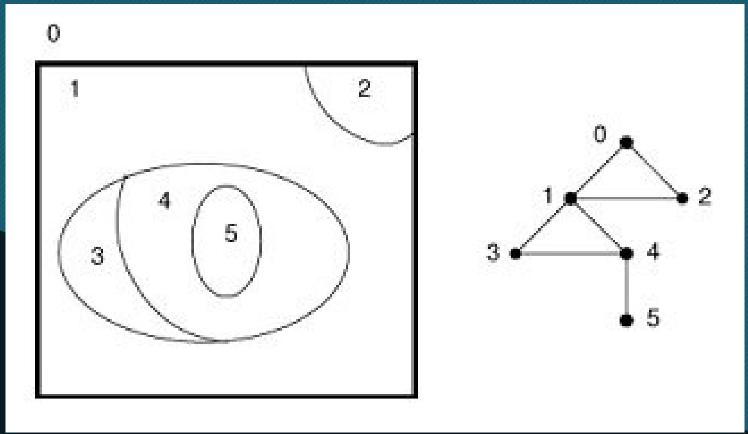
- Another example: run length coding
  - Run length coding has been used to represent strings of symbols in an image matrix.
  - Run length coding records only areas that belong to objects in the image.
  - The area is then represented as a list of lists.
  - The code of this example is
    ((1 1 1 4 4)(2 1 4)(5 2 3 5 5))

For binary images:
((Row#, begin col., end col. .... begin col., end col.)
....
(Row#, begin col., end col. .... begin col., end col.))



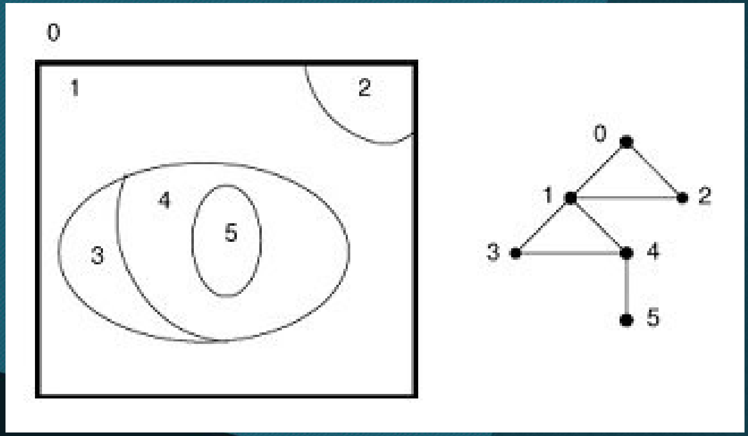
Traditional image data structures

- Topological data structure
  - Graph
    - A weighted graph is a graph in which values are assigned to arcs, to nodes, or to both.
    - The region adjacency graph is typical of this class of data structure.
    - For example,



Topological data structure

- The property of the region adjacency graph
  - If a region enclosed other regions, then the part of the graph corresponding with the areas inside can be separated by a cut in the graph.
  - Nodes of degree 1 represent simple holes.
    - For example, node 5.



Topological data structure

- The region adjacency graph is usually created from the region map.
  - Region map is a matrix of the same dimensions as the original image matrix whose elements are identification labels of the regions.
- The region adjacency graph can be used to approach region merging.
  - The region merging may create holes.
    (The topological property changes)

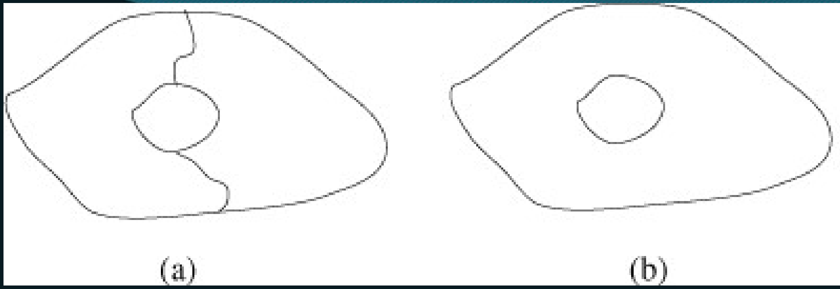


Figure 4.6: Region merging may create holes: (a) Before a merge. (b) After. © *Cengage Learning 2015.*

Relational structures

- Relational databases can also be used for representation of information from an image.
  - The image should be segmented first.
  - The information of objects, the important parts of the image, are then recorded in the relational table.

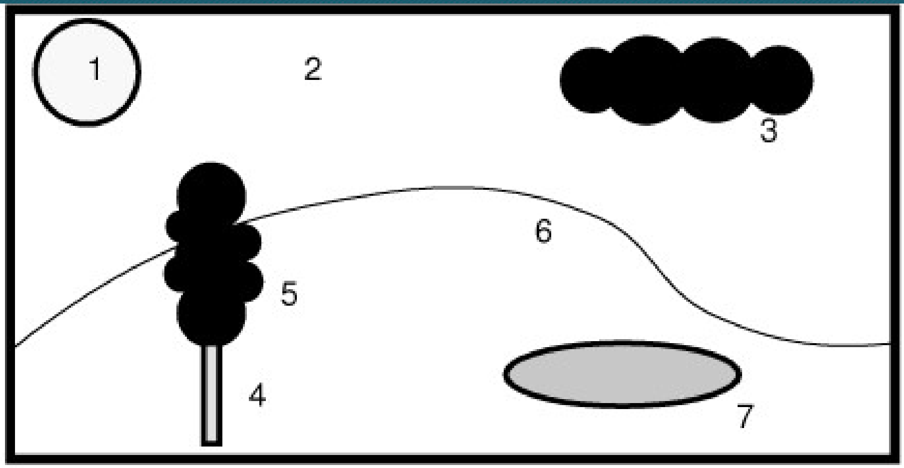
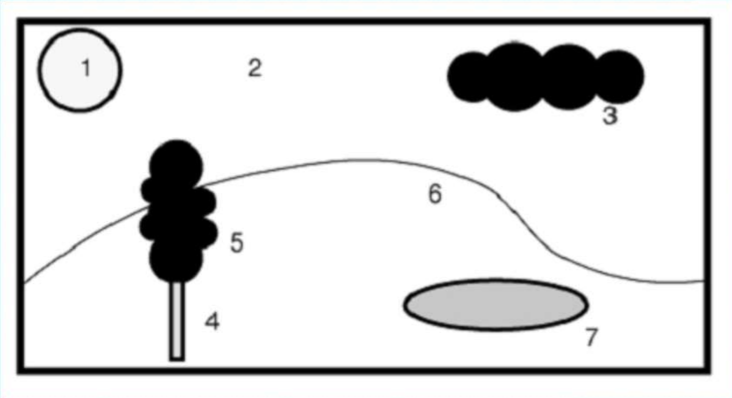


Figure 4.7: Description of objects using relational structure. © *Cengage Learning 2015.*

Relational structures

- Relational table
  - Relations are recorded in the form of table.

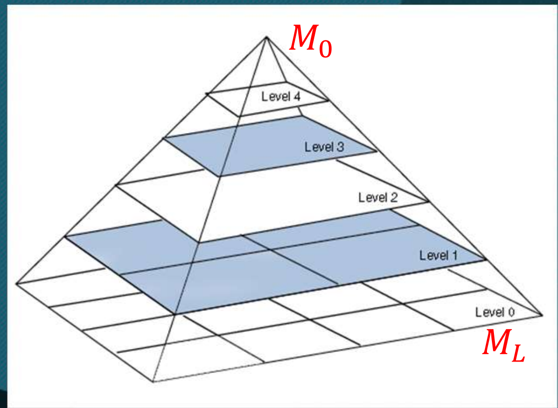


No.	Object name	Color	Min. row	Min. col.	Inside
1	sun	white	5	40	2
2	sky	blue	0	0	–
3	cloud	gray	20	180	2
4	tree trunk	brown	95	75	6
5	tree crown	green	53	63	–
6	hill	light green	97	0	–
7	pond	blue	100	160	6

Table 4.1: Relational table. © *Cengage Learning 2015.*

Hierarchical data structures

- Pyramids
  - Pyramids are among the simplest hierarchical data structures.
  - M-pyramids (Matrix-pyramids)
    - A M-pyramid is a sequence $\{M_L, M_{L-1}, \dots, M_0\}$ of images.
    - M_L has the same dimensions and elements as the original image.
    - M_{i-1} is derived from the M_i by reducing the resolution by one-half.
    - M_0 corresponds to one pixel only.
    - M-pyramids are used when it is necessary to work with an image at different resolutions simultaneously.



Hierarchical data structures

- T-pyramids (Tree-pyramids)
  - Let 2^L be the size of an original image.
  - A tree-pyramid (T-pyramid) is defined by

    1. A set of nodes P

       $P = \{p = (k, i, j)$ such that level $k \in [0, L]; i, j \in [0, 2^k - 1]\}$.

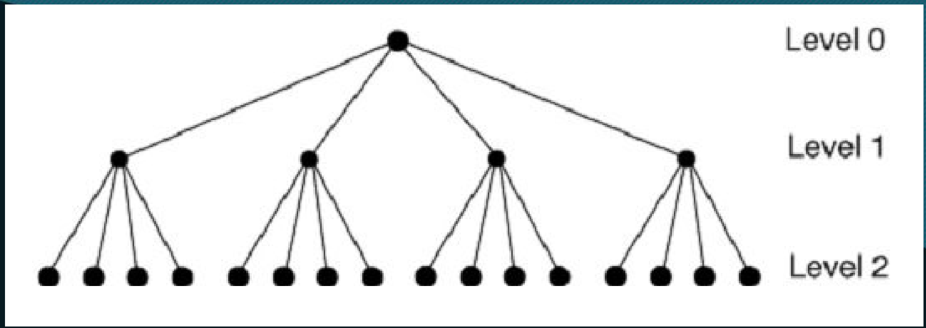
    2. A mapping F between subsequent nodes P_{k-1}, P_k of the pyramid

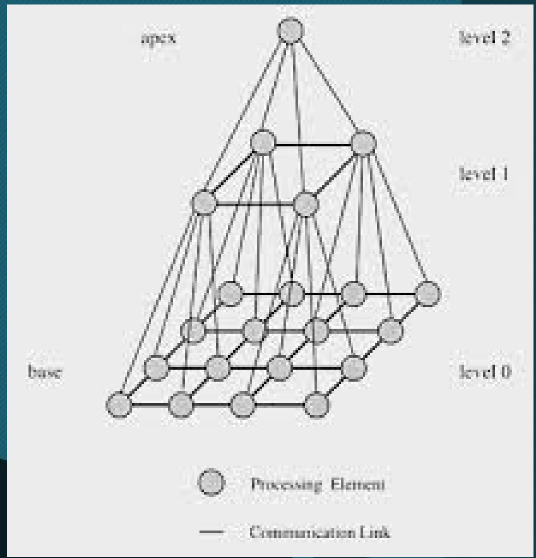
    $$F(k, i, j) = (k - 1, floor\left(\frac{i}{2}\right), floor\left(\frac{j}{2}\right))$$

    3. A function V that maps a node of the pyramid P to Z,
       where Z is a set of brightness levels,
       for example, $Z = \{0,1,2, \ldots, 255\}$

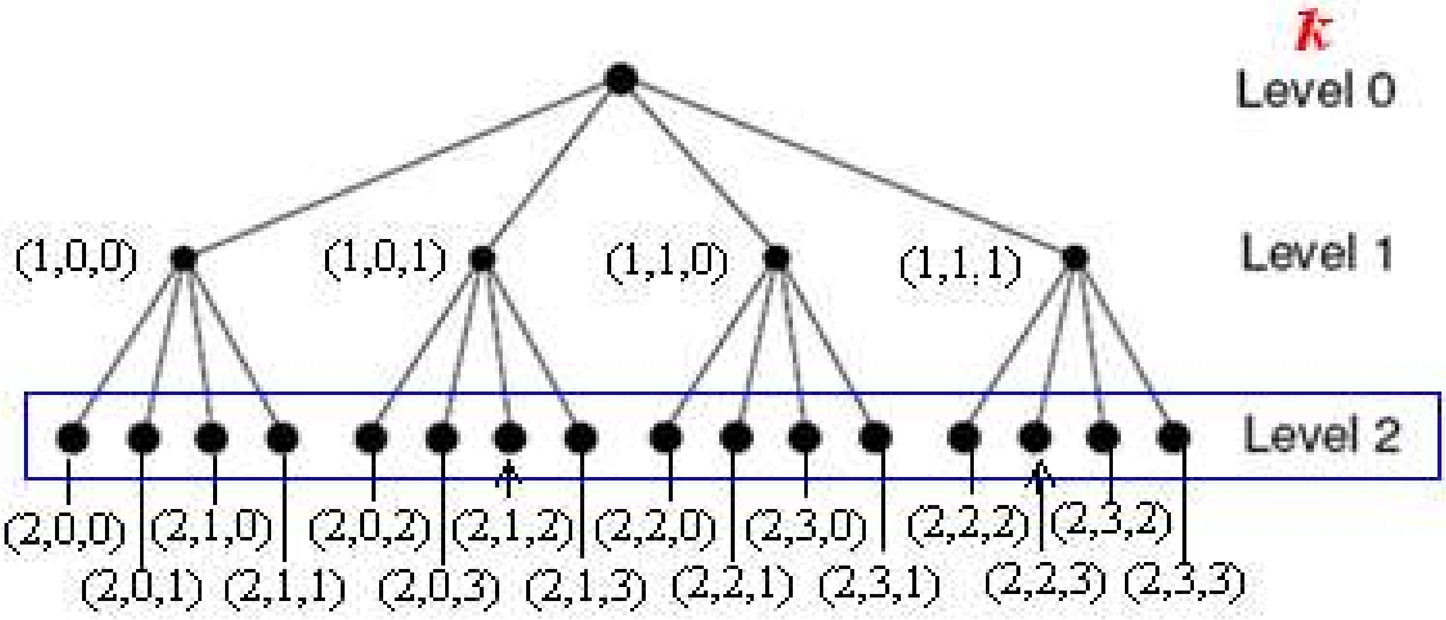
Hierarchical data structures

- T-pyramids (Tree-pyramids)
  - Function V defines the values of nodes.
    - For example, average, maximum, minimum,…
  - Values of leaf nodes are the same as values of the image function (brightness) in the original image at the finest resolution.
    - The image size is 2^L.





Tree-pyramids



$$F(2,1,2) = (1,0,1),$$
$$F(2,3,1) = (1,1,0)$$

$$F(k, i, j) = \left(k-1, floor\left(\frac{i}{2}\right), floor\left(\frac{j}{2}\right)\right)$$

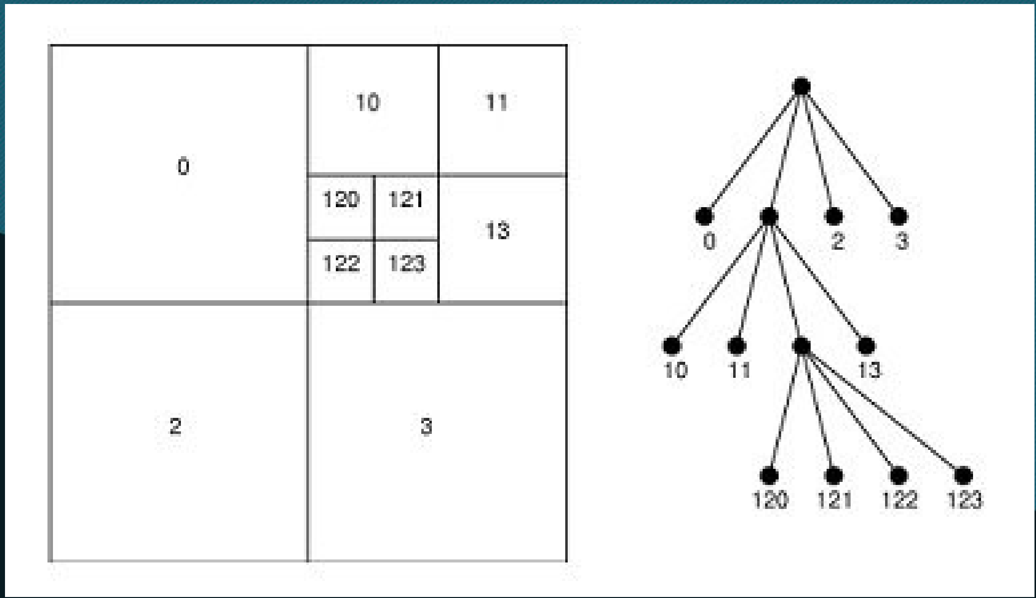
$V : P \rightarrow Z$ defines the values of nodes, e.g., average, maximum, minimum

Z: a set of brightness levels

Leaf nodes have pixel brightness values

Hierarchical data structures

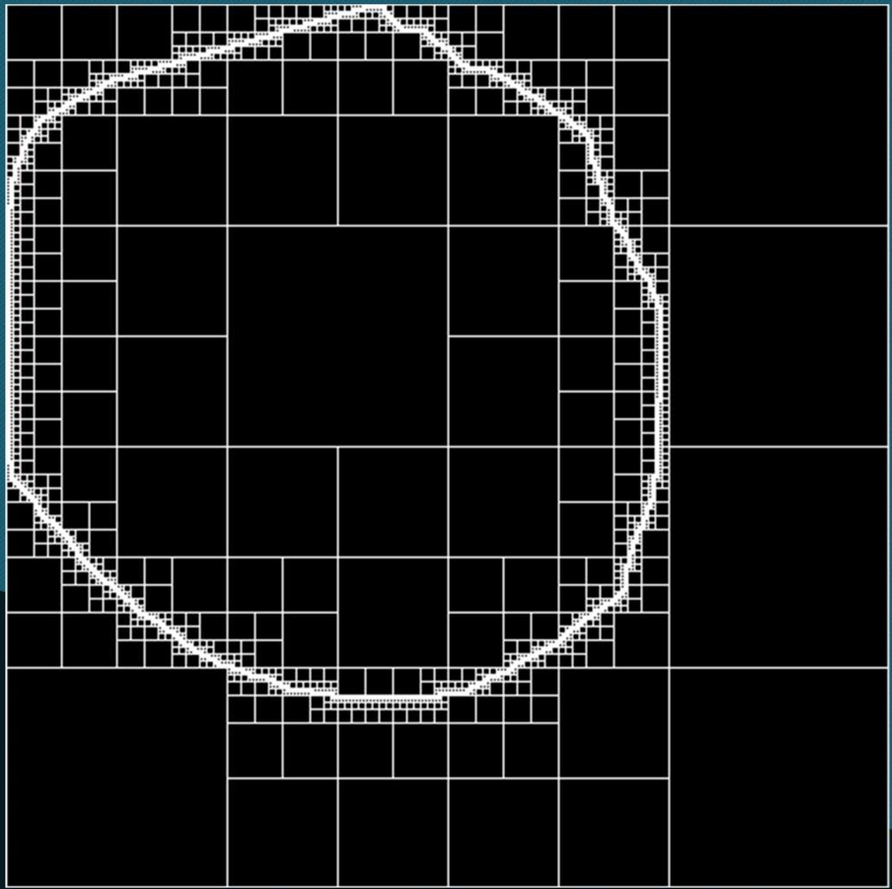
- Quadtrees
  - Quadtrees are modifications of T-pyramids.
  - Every node of the tree except the leaves has four children.

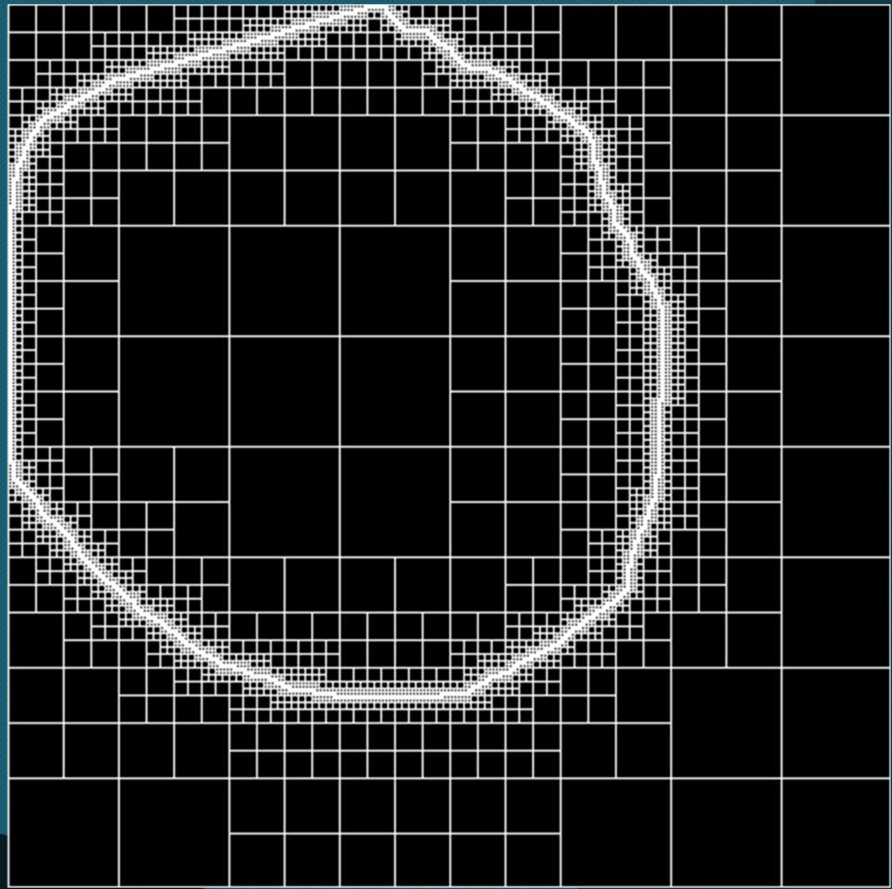


Node type
Pointer to the NW son
Pointer to the NE son
Pointer to the SW son
Pointer to the SE son
Pointer to the father
Other data

Record describing a quadtree node

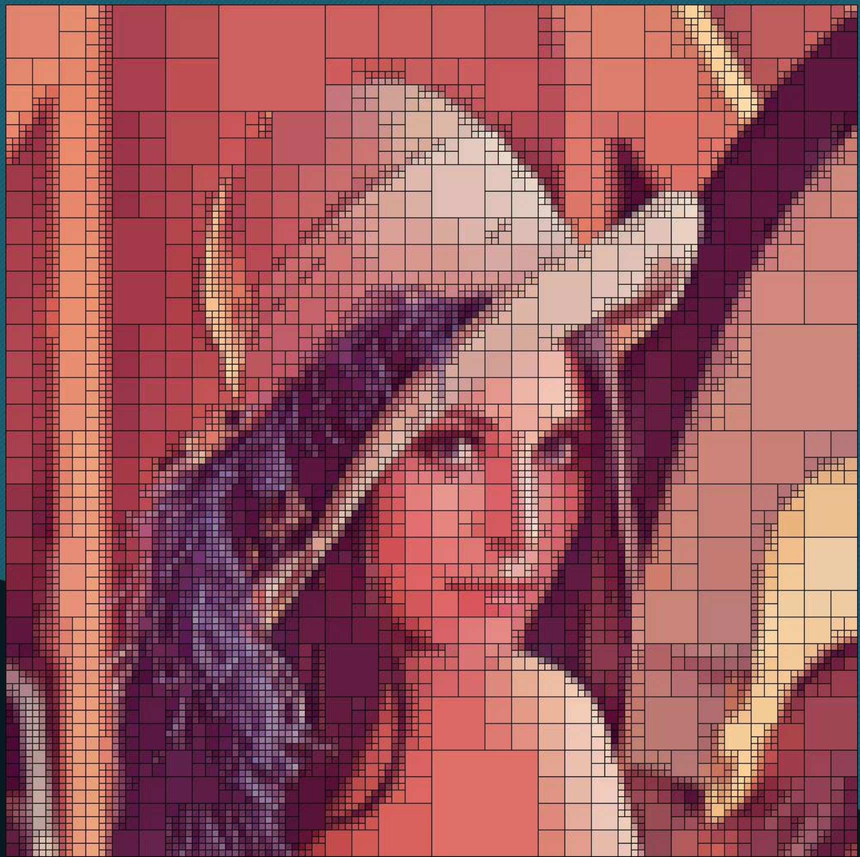
Hierarchical data structures

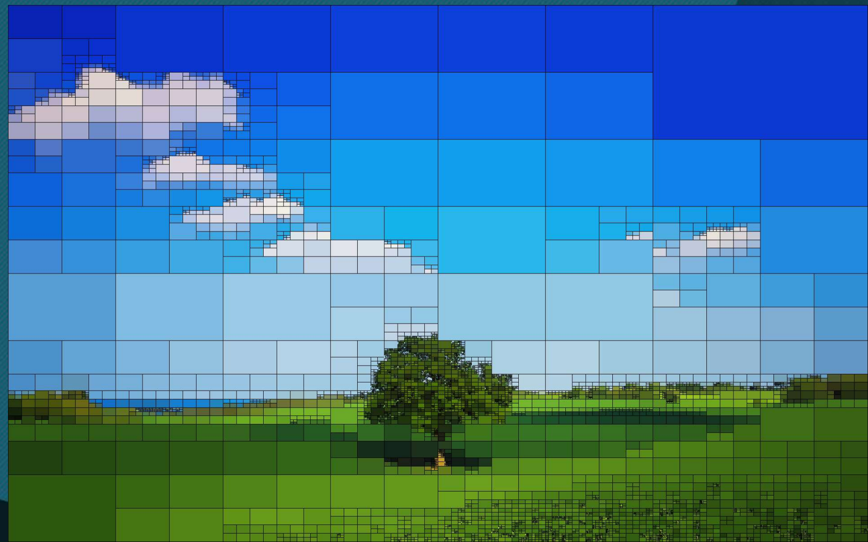




http://cybertron.cg.tu-berlin.de/pdci1 1ws/gdi/

Computer art based on quadtrees





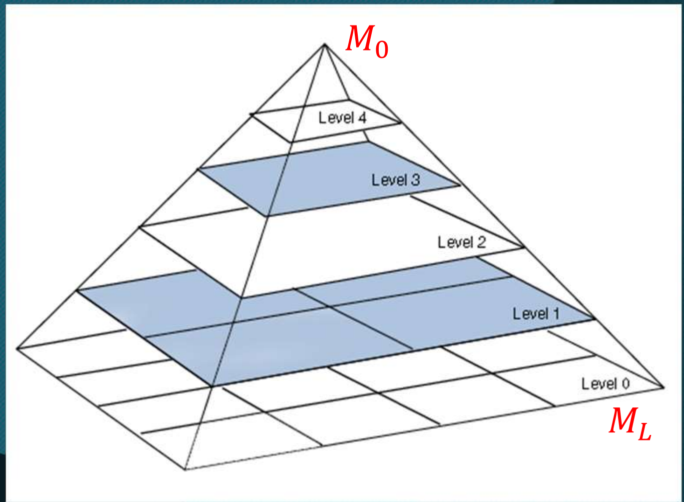
https://pythonawesome.com/computer-art-based-on-quadtrees/

Hierarchical data structures

- Advantage
  - An advantage of image representation by means of quadtrees is the existence of simple algorithms for addition of images, computing object areas, and statistical moments.
- Disadvantage
  - The main disadvantages of quadtrees and pyramid hierarchical representations is their dependence on the position, orientation, and relative size of objects.
  - Two similar images with just very small differences can have very different pyramid or quadtree representations.

Other pyramidal structures

- Reduction window
  - Recalling that a M-pyramid was defined as a sequence of images $\{M_L, M_{L-1}, \dots, M_0\}$ in which M_i is a 2×2 reduction of M_{i+1}.
  - Reduction window: for every cell c of M_i, the reduction window is its set of children in M_{i+1}, $w(c)$.
- Regular
  - If the images are constructed such that all interior cells have the same number of neighbors, and they all have the same number of children, the pyramid is called regular.



Other pyramidal structures

- Several regular pyramid definitions.

  (a) $2 \times 2\ /4$ (b) $2 \times 2\ /2$ (c) $3 \times 3\ /2$

  - (reduction window) / (reduction factor)
  - Reduction factor: the rate at which the image area decreases between levels.
  - Solid dots are at the higher level, i.e., the lower-resolution level.