Winter 2019

DMET 901 – Computer Vision

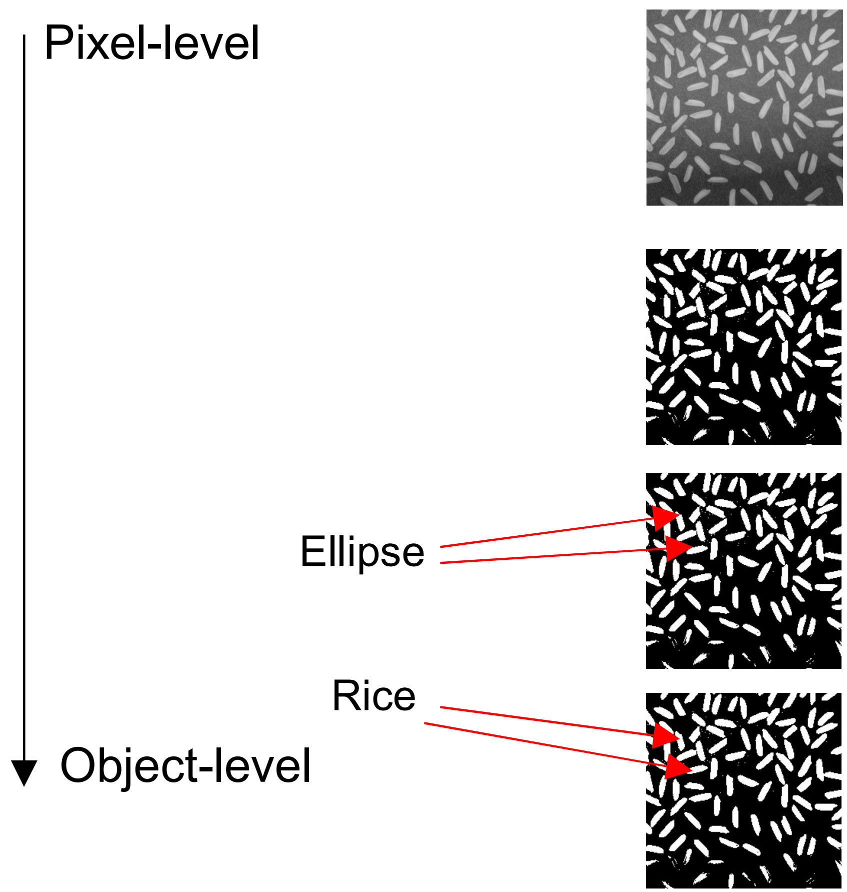
Image Representation and Properties (2)

Seif Eldawlatly

Data Structures

- Computer vision includes the design of
  - Data structures to represent an image
  - Algorithms used for processing and creating a model of the image

- Levels of Representation
  - Iconic Images
  - Segmented Images
  - Geometric Representation
  - Relational Models



Data Structures

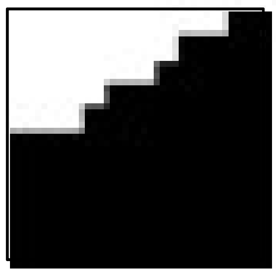
- Traditional Data Structures
  - Matrices
  - Chains
  - Topological
  - Relational
  - Hierarchical

Matrices

- Represent the most common data structure
- Examples

1. Binary Image: Entries of the matrix take the value of 0 or 1

Image

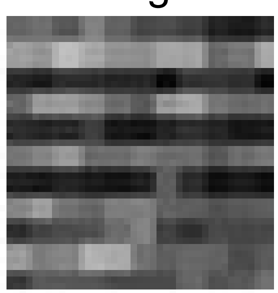


Matrix Representation

1	1	1	1	1	1	1	1	1	0	0
1	1	1	1	1	1	1	0	0	0	0
1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0

2. Gray Scale Image: Each value represents the intensity of a pixel

Image



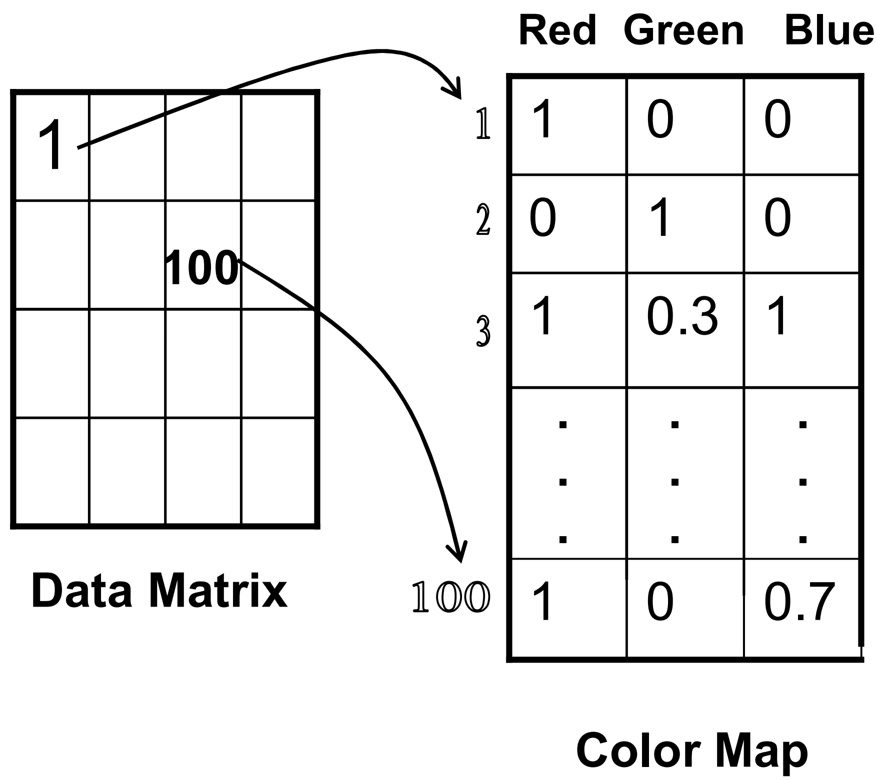
Matrix Representation

102	110	88	129	98	72	77	65	37	28	42
157	148	187	165	144	141	163	161	104	118	166
41	33	50	46	40	41	15	54	55	48	14
100	149	152	135	105	88	145	162	116	103	108
47	41	44	64	27	22	38	49	39	25	30
115	129	149	106	107	120	107	107	85	106	117
34	28	38	29	23	34	92	45	19	28	18
131	153	111	100	123	126	97	92	80	87	93
49	70	76	83	108	131	66	49	73	70	71
153	124	133	177	168	112	94	103	98	75	91
59	89	88	88	74	74	75	103	81	88	112

Matrices

- Examples (cont.)

  3. Indexed Images: Consists of two matrices (Data Matrix and Color Map)



Matrices

- Examples (cont.)

  4. Co-occurrence Matrix:

     - Used to capture a certain relation between pixels
     - For example:
       Given the following 5x5 image, find the co-occurrence matrix for each pixel and its south

4	6	8	5	4
5	5	8	7	7
6	7	7	7	9
8	8	4	8	6
9	8	9	5	6

5x5 image

South

North

N\S	4	5	6	7	8	9
4	0	1	0	1	0	1
5	0	0	1	2	0	0
6	0	1	1	0	1	0
7	1	0	0	1	2	1
8	0	1	0	1	2	1
9	0	0	1	0	0	0

C_r :Co-occurrence matrix

Matrices

- Examples (cont.)
  - Co-occurrence Matrix Algorithm:

    1- Let $C_r(z, y)$ = 0 for all $z, y \in$ [0,L], where L is maximum brightness of the image f

    2- For all pixels (i, j) in the image determine (h, k) which has the relation r with the pixel (i, j) and compute:

    $$C_r[f(i,j),\ f(h,k)] = C_r[f(i,j),\ f(h,k)] + 1$$

  - Interesting properties about images can be deduced from co-occurrence matrix
  - Example: Contrast can be estimated as $\sum_{i=1}^{m}\sum_{j=1}^{n} C_r(i,j)\left[f_r(i) - f_c(j)\right]^2$

4	5	6	7	*8*
4	5	6	7	*8*
4	5	6	7	*8*
4	5	6	7	*8*

4x5 Low Contrast Image

N\S	4	5	6	7	8
4	3	0	0	0	0
5	0	3	0	0	0
6	0	0	3	0	0
7	0	0	0	3	0
8	0	0	0	0	3

C_r : Co-occurrence matrix

Contrast = 3x(4-4)+0…..+0+3x(5-5)+0…..+3x(8-8)=0

Matrices

- Examples (cont.)
  - Co-occurrence Matrix Algorithm:

Contrast = 20728444

Contrast = 39771242

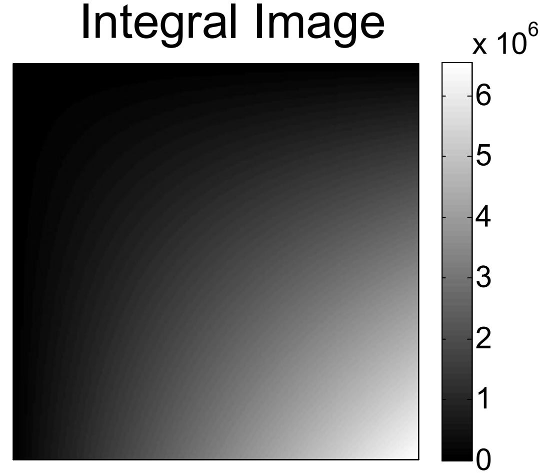
Matrices

- Examples (cont.)
  4. Integral Image:
     - Constructed such that the value $ii(i, j)$ is the sum of the intensity of all pixels left of and above the pixel at (i, j)
     - Algorithm:
       1. Let $s(i, j)$ denote a cumulative row sum, let $s(i, -1) = 0$
       2. Let $ii(i, j)$ be an integral image, let $ii(-1, j) = 0$
       3. For each row and each pixel (i, j), calculate $s(i, j)$ and $ii(i, j)$ as

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

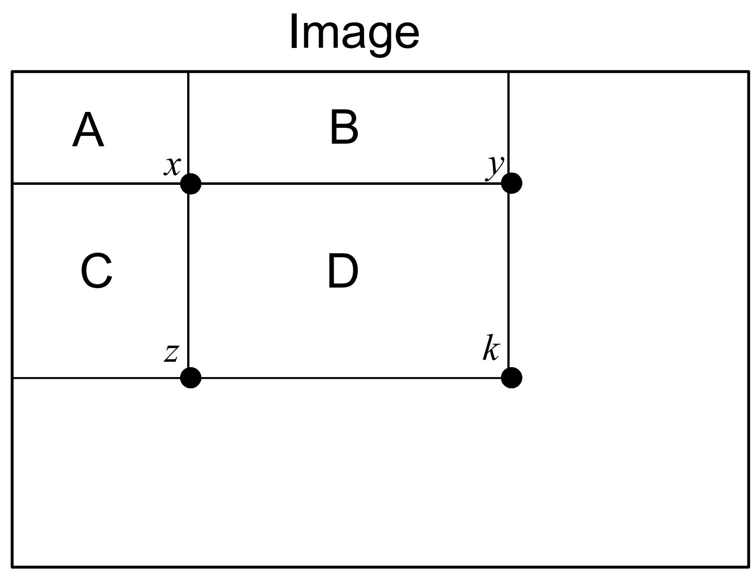
Image with intensity = 100 for all pixels

Integral Image



Matrices

- Examples (cont.)
  - Integral image is useful in computing the total (or average) intensity in any rectangular section of the image



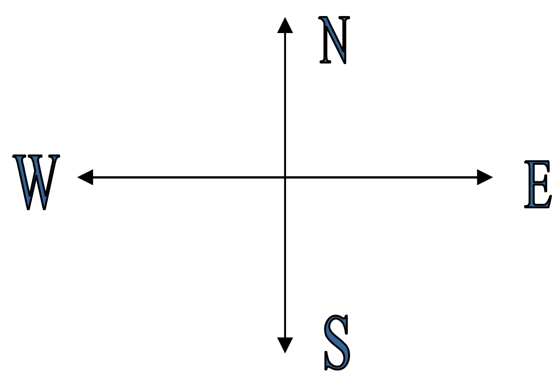
Total Intensity in section A = $ii(x)$

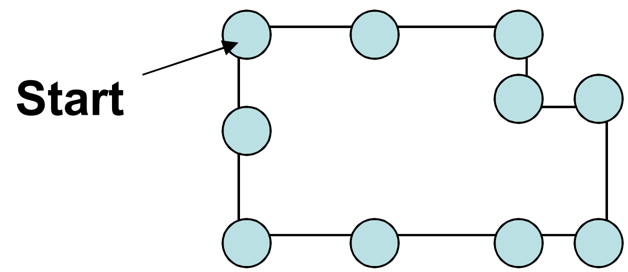
Total Intensity in section C = $ii(z) - ii(x)$

Total Intensity in section D = $ii(k) + ii(x) - ii(z) - ii(y)$

Chains

- Most widely used to describe object borders
- 4-neighbor chain code





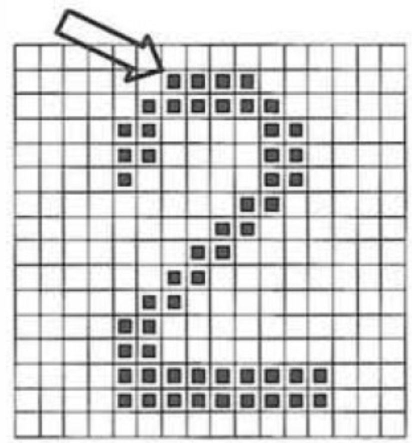
E-E-S-E-S-W-W-W-N-N

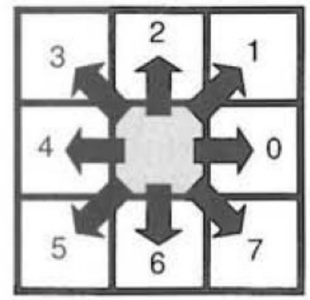
- 8-neighbor chain code (More symbols, smaller chain)

E-E-SE-S-W-W-W-N-N

Chains

- Example





00077665555556600000006444444442221111112234445652211

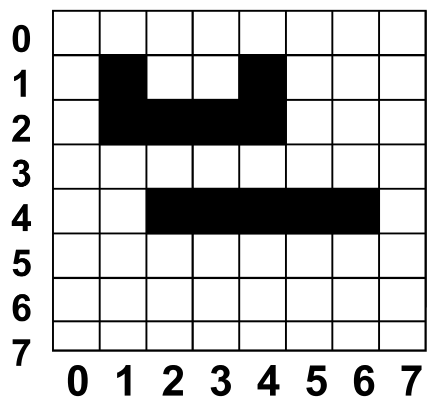
- Chains Limitations
    - Relative as they depend on starting pixel
    - Can be modified dramatically because of noise
    - The entire chain must be searched to find a certain relation

Chains

- Run-Length (RL) Coding
  - Based on the observation that pixels that are close together are likely to have the same brightness

  - Example: One simple RL code for binary images

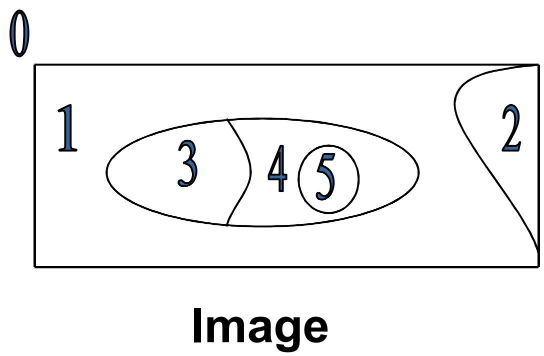
    (row, {first element in run, last element in run}*)*

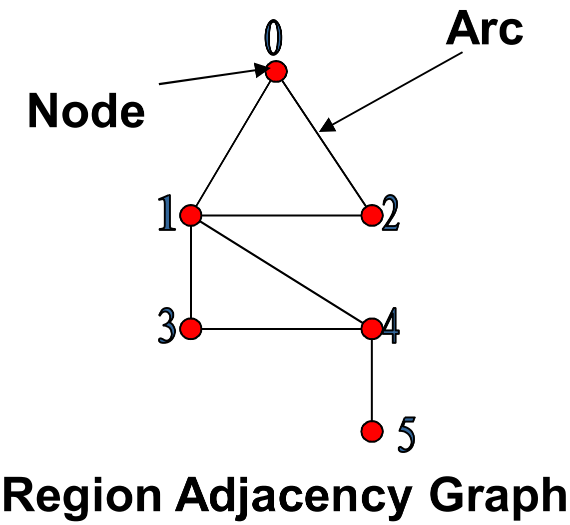


RL code: (11144)(214)(426)

Topological Data Structures

- Used to determine adjacency between regions
- Represented as a graph with nodes V and arcs E

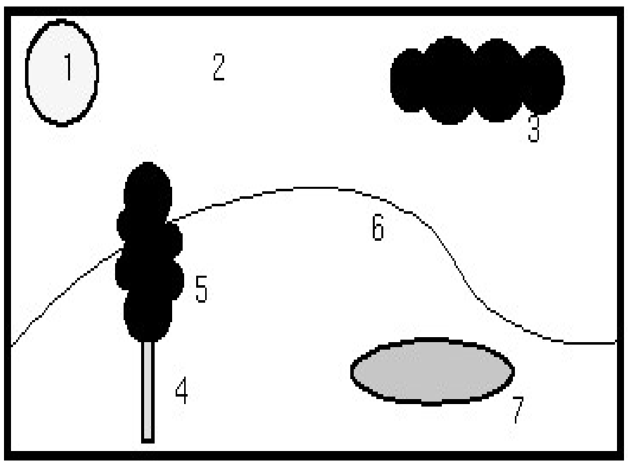




- Can detect if a region encloses another region and if a region is a hole (nodes with single arc)
- Region adjacency graph is constructed from the region map: A matrix with the same size as the original image whose elements are identification labels of the regions

Relational Data Structures

- Information is concentrated in relations (tables) between semantically important parts of the image
- The important parts (objects) are identified using segmentation



Description of objects using relational structure.

No.	Object name	Colour	Min. row	Min. col.	Inside
1	sun	white	5	40	2
2	sky	blue	0	0	-
3	cloud	grey	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

Relational table

Hierarchical Data Structures

- Computer vision has 2 conflicting goals
  - Large data sets should be processed
  - Process should end in the shortest time

- To solve this problem, parallel computing can be used by dividing data into smaller pieces

- Sometimes it is hard to divide a computer vision problem among processors

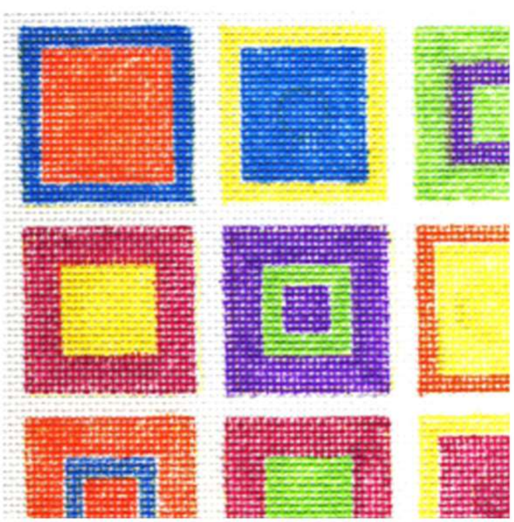
- Hierarchical structures can find an intermediate solution in this case

- Hierarchical data structures examples:
  - Pyramid
  - Quadtree

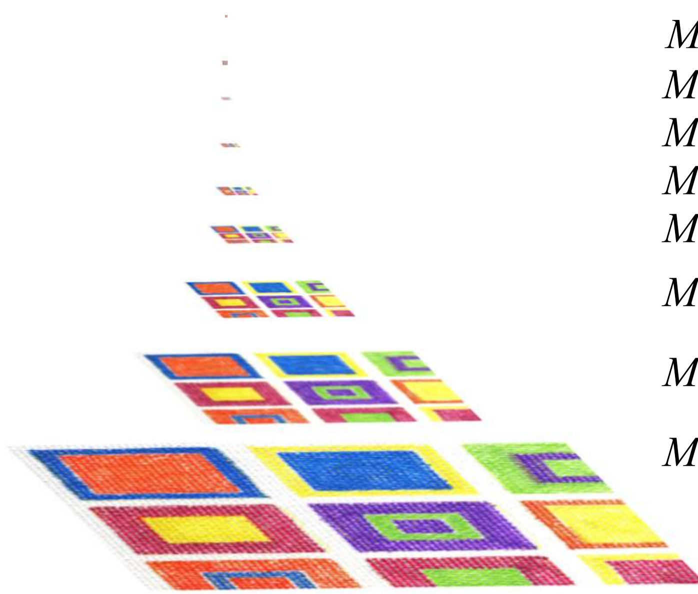
Hierarchical Data Structures

- Pyramids
  - A matrix pyramid (M-pyramid) is a sequence $\{M_L, M_{L-1}, \ldots, M_0\}$ of images where M_L is the original image and the resolution of M_{i-1} is half that of M_i
  - Can be used to deal with an image at different resolutions simultaneously
  - Example

Original Image (256 x 256)



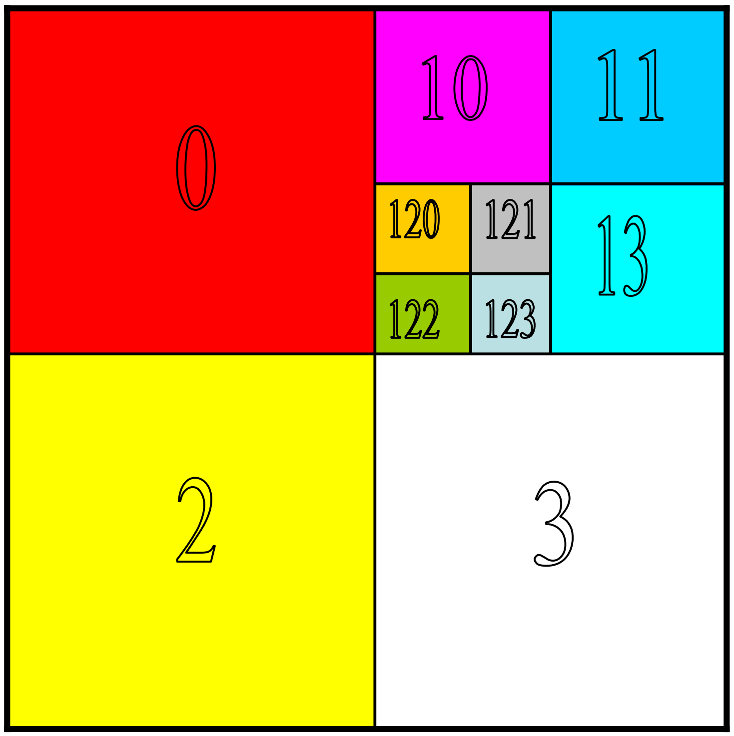
M-pyramid



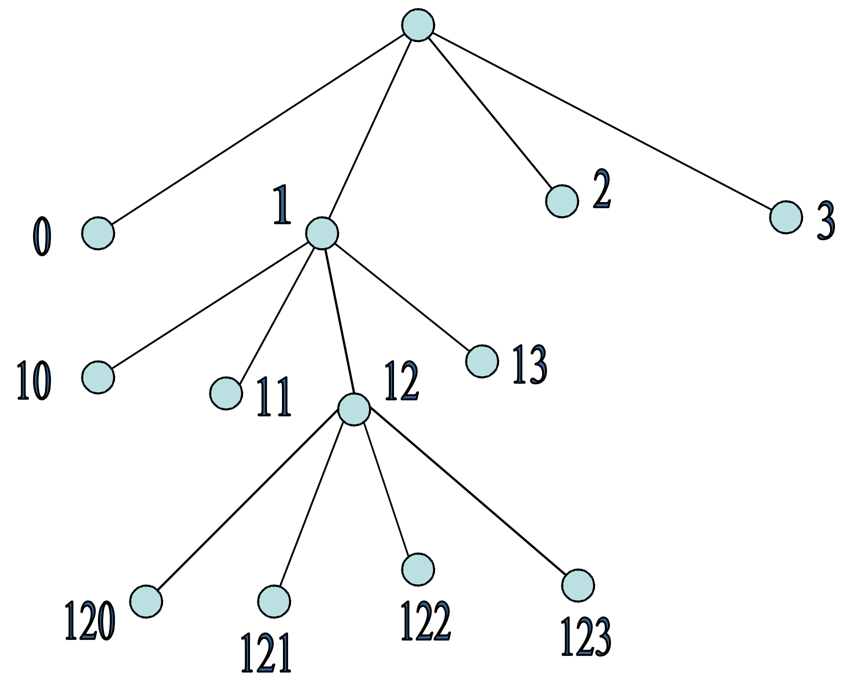
M_0 (1 x 1)
M_1 (2 x 2)
M_2 (4 x 4)
M_3 (8 x 8)
M_4 (16 x 16)
M_5 (32 x 32)
M_6 (64 x 64)
M_7 (128 x 128)
M_8 (256 x 256)

Hierarchical Data Structures

- Quadtree
  - Very compact representation for images with large homogeneous regions



8x8 image



Quadtree