5.a

Binary Images

⇒ Gray Scale Image ⇒ Pixel typically takes value 0 and 255.

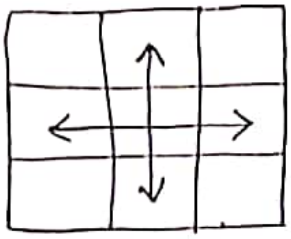
⇒ Binary Image ⇒ Pixel can take only 0 or 1.

↳ Application

↳ OCR, masking etc...

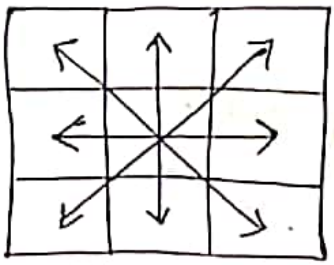
$$b(i,j) = \begin{cases} 1, & \text{if there is an object at } (i,j) \\ 0, & \text{otherwise} \end{cases}$$

★ Neighborhood in Grids



N4 Neighborhood

$N_4(i,j)$

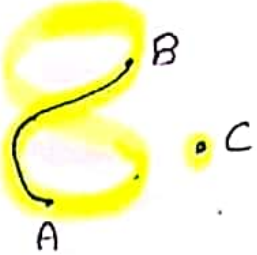


N8 Neighborhood

$N_8(i,j)$

★ Connected Components

⇒ Two points A and B are connected if there exists a path from A to B that goes only through same component.



* Determining Connected components via a Graph

⇒ Build graph of the foreground pixels with edges according to N8 neighbors.

* Labeling Approach

1. Select an unlabeled node and asign a new label to it.
2. For each unlabeled neighbor of a labeled node, assign the same label.
3. Repeat step 2 until all neighbors are labeled.
4. Repeat step 1 until all nodes are labeled.

Also called: a "brushfire" approach.

* Connected Components (Algorithm)

Input: binary image $b(i,j) \in \{0,1\}$

Output: ① Number of Components K

② Component Image $K(i,j) \in \{0:K\}$

• Component number $K=0$, Component image $K(i,j)=0$

2 repeat
3 | find $(i,j) \mid \{b(i,j)=1, K(i,j)=0\}$;
4 | $S := \{(i,j)\}$;
5 | $K := K+1$
6 | $K(i,j) = K$;
7 | repeat
8 | | find unlabeled foreground neighbors $N(S \mid b(i,j)=1, K(i,j)=0)$;
9 | | label all $(i,j) \in N(S)$ with K: $K(i,j)=K$;
10 | | $S = S \cup N(S)$
11 | until no neighbor exists: $N(S) = \emptyset$;
12 until no unlabeled foreground pixel exists;
13 return K, $K(i,j)$;

★ Properties of the algorithm

⇒ The Set of neighbors may become large
⇒ Provides the Connected components

★ Improved Labeling by Exploiting the Grid Structure of the Image

Idea

→ Process the image in one pass.
• Process the image from left to right, top to bottom.

→ If none of its (top or left) neighbor is 1, assign new label
→ If all neighboring (top or left) labels are identical, copy the label.
→ If label differs, copy the one label and update the equivalence table.

Q. ⇒ Re-label with the smallest of equivalent labels.

⇒ Extends trivially for N8 neighborhoods.

Algorithm: Connected Component in binary image.

Input ⇒ binary image $b(i,j) \in \{0,1\}$

Output ⇒ Component number K, Component image $K(i,j) \in \{0:K\}$

```
1  Component number K=0, equivalence table E=φ;
2  for i=1:I do all rows
3     for j=1:J do all columns
4        if b(i,j)=1 then
5           A=N(i,j) all labeled neighbors;
6           if |A|=0 then
7              K:=K+1
8              K(i,j)=K;
9           end
10          if |A|≥1 then
11             K(i,j):=min(K(A));
12             for all the x in A with K(x)≠K(i,j) do
13                extend equivalence table:
14                E:=E∪{K(i,j),K(x)};
15             end
16          end
17       end
18    end
19 Compute Connected Components of E using Algo 1
20 replace {K(i,j), i∈I, J∈j with smallest number
   of the component in equivalence graph.
21 return K, K(i,j);
```

⇒ Exploits the grid neighborhood.
⇒ Requires only one pass through the image for labeling.
⇒ Second pass to eliminate duplicate labels.
⇒ Linear Complexity in the number of foreground pixels.

Binary Image

* How to Determine the Number of holes in an Image

→ Invert the image
→ Apply the same algorithm.
→ Number of components gives the number of holes.

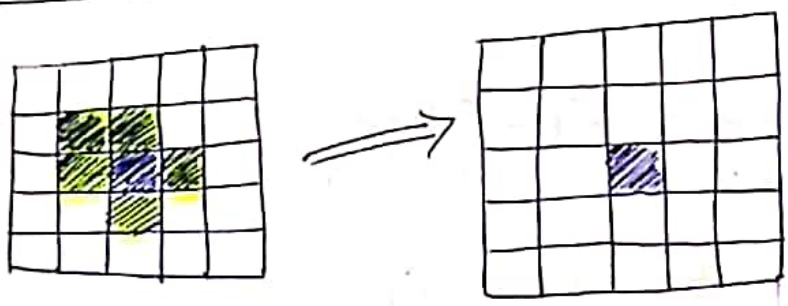
* Filtering

- Binary images are often obtained through thresholding (Point operation)

* Morphological Operators

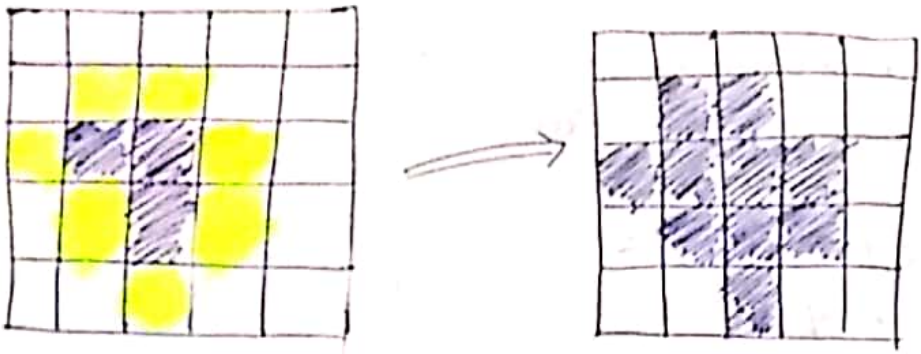
- Shrinking the foreground ("erosion")
- Expanding the foreground ("dilation")
- Removing holes in the foreground ("closing")
- Removing stray foreground pixels in background. ("opening")

* Erosion



⇒ Change each foreground (black) pixel to a background (white) pixel if it has a background pixel as its N4 neighbor.

* Dilation



⇒ Changing each background (white) pixel to a foreground (black) pixel if it has a foreground pixel as its 4N neighbour.

* Opening

⇒ Dilation followed by erosion.

⇒ Removing stray foreground pixels in the background.

* Closing

⇒ Removing holes in the foreground.

⇒ Erosion followed by a dilation.

⇒ Morphological operators can be used to eliminate noisy masks.

↳ They are Local operators.

* Distance Transformation

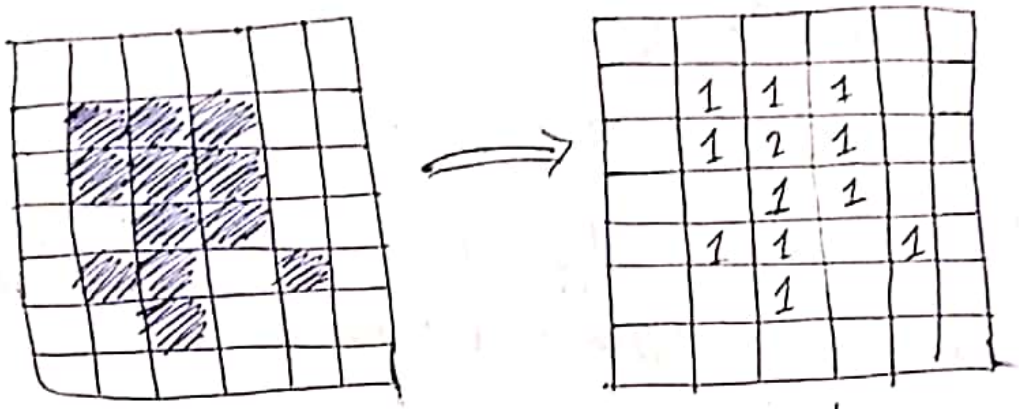
⇒ DT on the foreground pixels is defined as

$$d(r,c) = \begin{cases} \min_{(u,v) \in S_R} [D_x((r,c),(u,v))], & \text{if } b(r,c) = 1 \\ 0 & \text{otherwise.} \end{cases}$$

S_R ← boundary

⇒ Different distance functions:

$$D_4((x,c),(u,v)) = |u-x| + |v-c|$$
$$D_8((x,c),(u,v)) = \max(|u-x|, |v-c|)$$



⇒ The DT can be computed similar to the connected components.

⇒ Two passes through the image
- 1st: top-down, left-right
- 2nd: down-up, right-left

⇒ Always store the minimum distance.

★ N4 Vs N8 Neighborhood

⇒ The N4 neighborhood overestimates the Euclidean distance.

⇒ The N8 neighborhood underestimates the Euclidean distance.

⇒ The Sum $D_4 + D_8$ provides a better approximation for twice the distance.

$$D_o = \frac{1}{2}(D_4 + D_8) \quad \{\text{Approx Euclidian distance}\}$$

⇒ By using $D_o = (D_4 + D_8)$, we can exploit integer computations.

(5c)

Binary Image

* Skeleton

⇒ A Skeleton can be seen as a Simplification of the Image.

⇒ Three Views of a Skeleton :-

① A point is a part of the skeleton if it has the same minimum distance to at least two different points on the border.

② The set of centers of circle, which lie fully inside the border and touches the border at least twice.

③ The set of all points in which the brushfire, which starts at the border meets.

* Computing the Skeleton

⇒ Eliminate all pixels at the border that

- have only one neighboring component, i.e. do not break a component and,
- are not endpoints (endpoints are points with less than 4 N8 neighbors).

→ For pixels at N, S, E, W border.

★ Issues

⇒ Not rotational invariant
⇒ Complexity $O(N^2 d)$ {d = width of frame}
⇒ Problems with small objects.

★ Skeleton via Distance Transform

Idea

⇒ Compute the distance transform
⇒ Search for local maxima in the DT
⇒ Maxima search via local grid-patterns.

★ Binary Image Features

⇒ A feature describes certain properties of an image.
⇒ We look into two types of features:

① Topological features.
{Describe neighborhood relations}

② Geometric features
{Describe the shape}

★ Topological Features

⇒ They are invariant under most transformation.

Example

→ Number of components K.
→ Number of holes L.
→ Euler characteristic is one topological features defined as

$$E = K - L$$

* Geometric Features

⇒ Features that discribe the geometry of the figure.

⇒ Two simple geometric features are:

- Area
- Perimeter

Algorithm: Computing Area and Perimeter

```
Initilize U=0, F=0
for i=0,...I-2 do
    for j=0,...,J-2 do
        n := b(i,j)+2b(i,j+1)+4b(i+1,j)+8b(i+1,j+1)
        U := U + LUT_dU(n);
        F := F + LUT_dF(n);
    end
end
```

* Geometric Feature: Form Factor

⇒ Form factor K combines area and Perimeter using

$$K = \frac{U^2}{2\pi F} \geq 1$$

⇒ Circle have: $K = 1$

⇒ Area and perimeter are invariant to translation (not to rotations!)

⇒ Form factor invariant to scale changes

* Moments of a Binary Image

$$M_{KJ} = \sum\sum i^K j^J b(i,j)$$

{Raw moment}

$M_{00} \Rightarrow$ area

* Center of Mass

⇒ The moments M_{10} and M_{01} can be used with M_{00} to compute the center of mass in i and j.

$$\bar{i} = \frac{M_{10}}{M_{00}} = \frac{1}{F}\sum\sum i\, b(i,j)$$

$$\bar{j} = \frac{M_{01}}{M_{00}} = \frac{1}{F}\sum\sum j\, b(i,j)$$

* Central Moments of a Binary Image

⇒ Given the Center of mass, we can compute the Central moments.

$$M_{KJ} = \sum\sum (i-\bar{i})^K (j-\bar{j})^J b(i,j)$$

* 2nd Central Moment

⇒ The - normalized by the area - second Central moment lead to the matrix

$$M = \frac{1}{M_{00}} \begin{bmatrix} M_{20} & M_{11} \\ M_{11} & M_{02} \end{bmatrix} \equiv \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}$$

⇒ M is the Covariance matrix of the Probability distribution over the image.

$$p(i,j) = \frac{1}{F} b(i,j)$$

* Semi-Major/Minor Axis of M

⇒ Through the Eigenvalues and Vectors of M, we the Semi-major/minor axis of the moment ellipsoid.

⇒ Information about the extent of the object in these directions.

⇒ Invariant under movements in i and j.

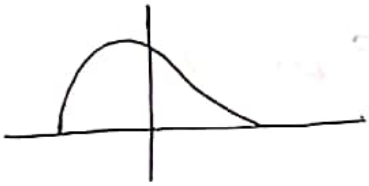
⇒ The direction of the Semi-major axis.

$$\varphi = \frac{1}{2}\arctan(2M_{12}, M_{11} - M_{22})$$

* 3rd Central Moment - Skewness

⇒ Skewness is a measure of the asymmetry of the distribution.

$$\gamma = \frac{M_3}{\sigma^3} = \frac{\sum (x-\bar{x})^3 b(x)}{F\,\sigma^2}$$



$\gamma > 0$

$\gamma < 0$

⇒ Skewness is often computed along the Semi-major/minor axis of the ellipsoid.