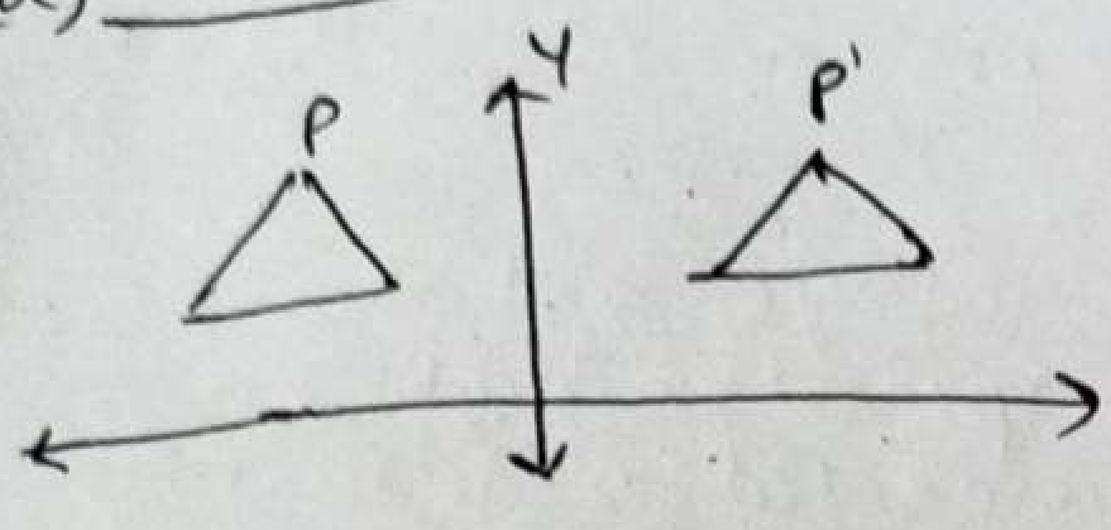
## Reflection :-

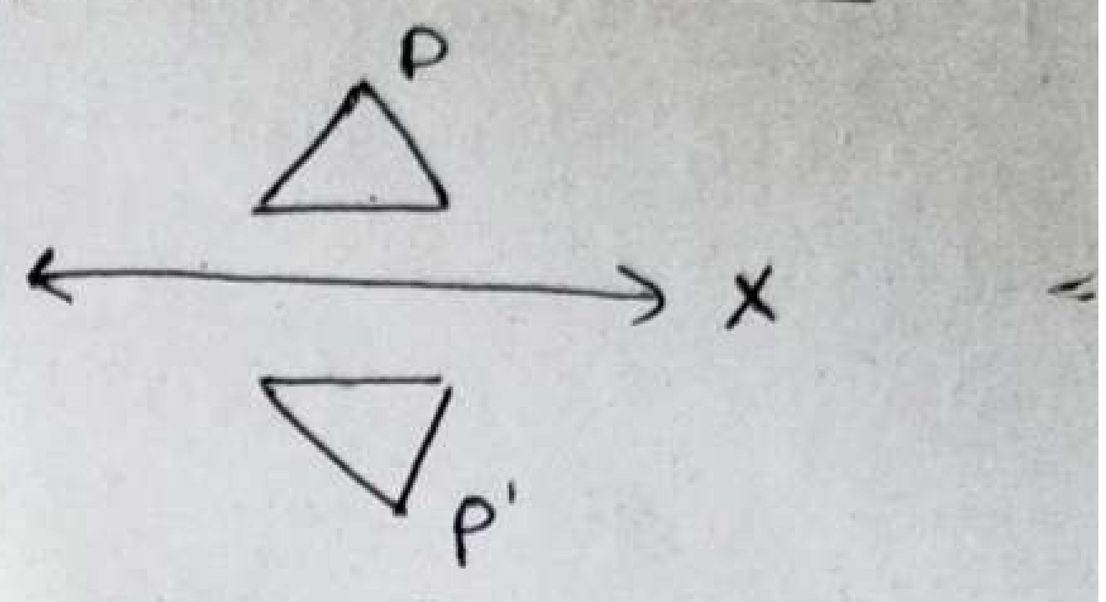
(a) x-axis reflection :-




~~for~~ $\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$ → multiply with original points.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \times \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

(b) y-axis reflection :-




$\begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$.

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} \times \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix}$$

## Thresholding :- [ 3rd module ].

Thresholding is an image processing technique that creates bitonal image based on a setting of threshold value on the pixel intensity of the original image.

Let's say threshold value = 100.

If < 100 → 0 ,
> 100 → 255.

| | | | |
|---|---|---|---|
| 50 | 10 | 20 | 105 |
| 10 | 120 | 5 | 5 |
| 200 | 105 | 50 | 10 |
| 20 | 50 | 101 | 10 |

→

| | | | |
|---|---|---|---|
| 0 | 0 | 0 | 255 |
| 0 | 255 | 0 | 0 |
| 255 | 255 | 0 | 0 |
| 0 | 0 | 255 | 0 |

[ bitonal - a bitonal image is an image consisting of only pure black and white pixels. ]

DIP – 12 (TH) [3rd module] → 2/11/22

## Thresholding :-

$f(x,y) \geq$ adaptive value → 1

$f(x,y) <$ adaptive value → 0

Ex →

$$\begin{bmatrix} 4 & 3 \\ 6 & 5 \end{bmatrix}$$ Threshold ~~the~~ w.r.t. 4. (adaptive value) ~~there~~

↓

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}$$ ← bitmap image since pixel is represented using a single bit.

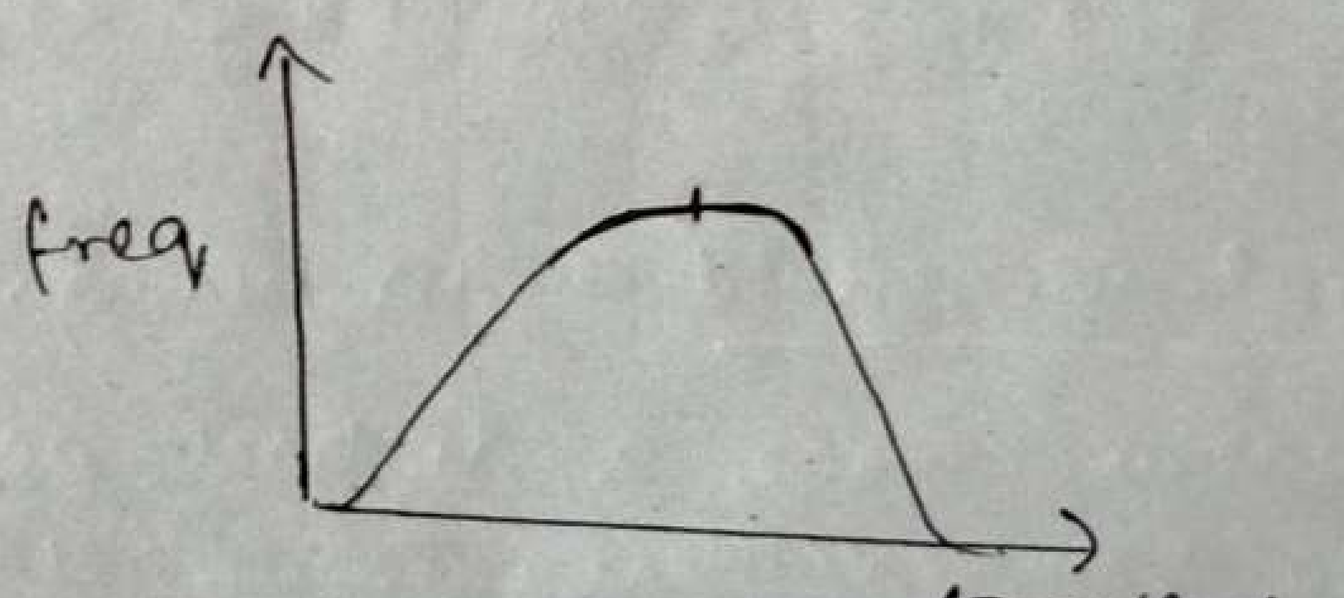
**Global Threshold –** If thresholding operation only depends on grayscale value, it is known as global thresholding.

**Local Threshold –** when applying the thresholding, suppose we are considering the neighbourhood and the adjacency properties in each block of a pixel, that is known as local thresholding.

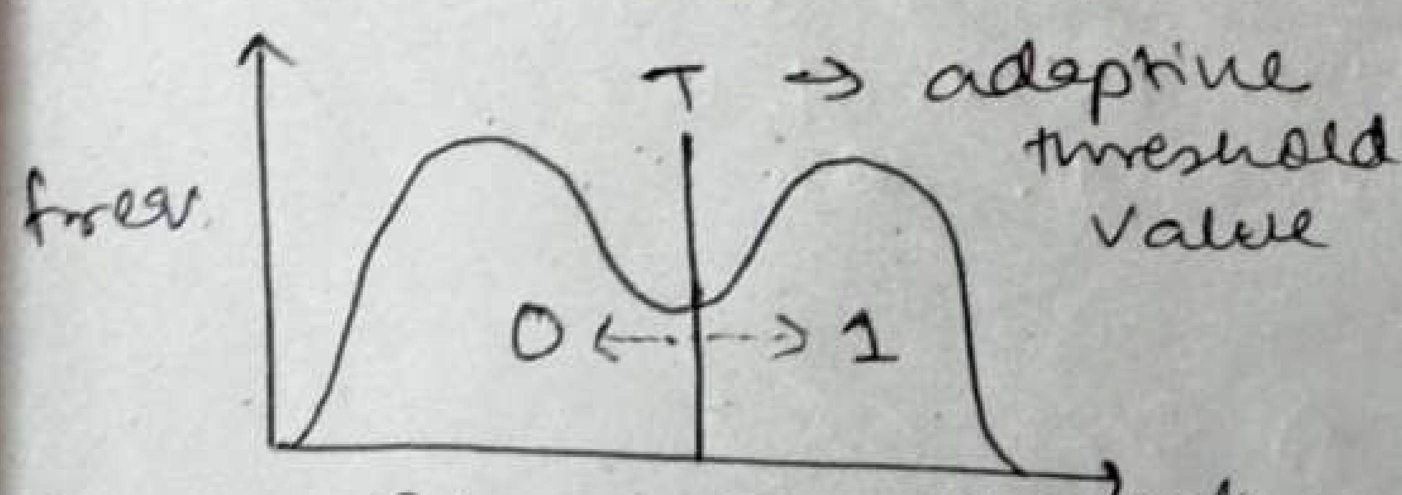
**Iterative Threshold –** If the adaptive threshold value depends on the coordinate of a pixel, then that is known as iterative thresholding.

## Histogram :-

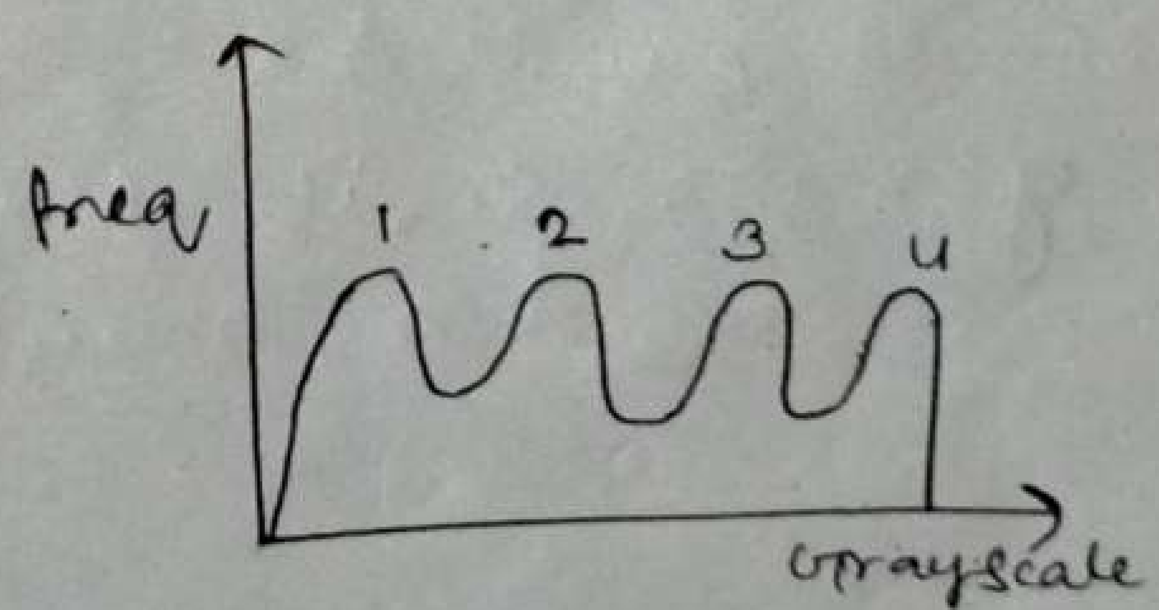
- unimodal
- Bimodal
- Multimodal




unimodal

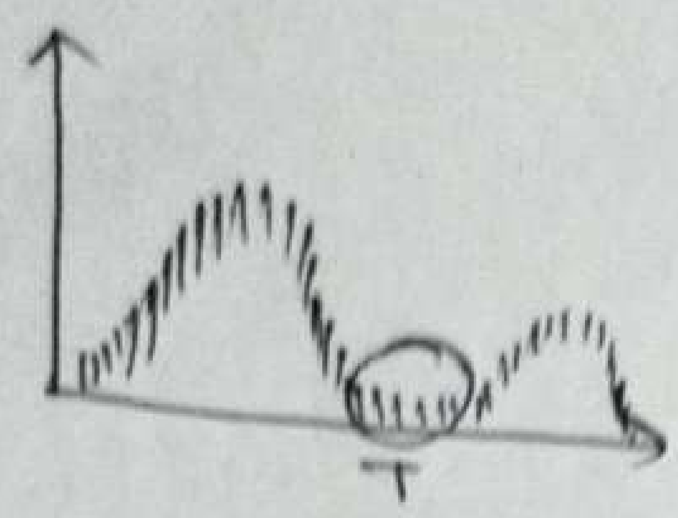



Bimodal




Multimodal

Suppose that an image $f(x,y)$ is composed of light objects on a dark background. The histogram looks like this -




<u>Single level Thresholding</u> =

In a single level thresholding, the gray level which is present between the light object and the ~~gra~~ dark object that is known as adaptive threshold value.

The objects can be extracted by comparing the pixel values with the threshold value T at any point $(x,y)$

if $f(x,y) > T \rightarrow$ it is object

$f(x,y) < T \rightarrow$ it is ~~object~~ background.

<u>Multilevel Thresholding</u> =

It is also possible to extract objects that have a specific intensity range using multiple threshold.

If $T_1 < f(x,y) <= T_2 \rightarrow$ object.

If $f(x,y) < T_1$ or $f(x,y) > T_2 \rightarrow$ Background.

<u>Global Threshold</u> - In bimodal ~~thresho~~ images histogram have 2 distinct peaks seperated by a valley between them. The valley point is chosen ~~at a~~ as the threshold point T. Then the pixels of the given image are compared with threshold. ~~The~~

The threshold process is

$$g(x,y) = \begin{cases} 1, & \text{if } f(x,y) > T \\ 0, & \text{otherwise} \end{cases}$$

Algorithm of calculating the threshold -

Step 1: Randomly select an initial threshold T.

Step 2: Segment the image using group of pixel ~~g1 and~~ g2. $G_1$ & $G_2$.

Step 3: Determine the mean $M_1$ of group $G_1$.

Step 4: Determine the mean $M_2$ from group $G_2$.

Step 5: New ~~thres~~ threshold is $T_{new} = \frac{M_1 + M_2}{2}$.

Step 6: Repeat step 2 to 5 until the difference in T is negligible.

DIP-13 (TH) 9/11/22

## Multiple thresholding-

Multiple thresholding is a process that segments a gray-level image into several distinct regions. It is an extension of simple thresholding technique.

$f_i \rightarrow$ value of input image pixel

$t_1, t_2, t_3, \ldots, t_n \rightarrow$ Multiple threshold value.

$g_1, g_2, g_3, \ldots, g_n \rightarrow$ values of output image.

$$\text{output image} = \begin{cases} g_1 & \text{if } 0 \leq f_i \leq t_1 \\ g_2 & \text{if } t_1 \leq f_i \leq t_2 \\ \vdots & \vdots \\ g_n & \text{if } t_{n-1} \leq f_i \leq 255 \end{cases}$$

## Adaptive Thresholding -

Adaptive Thresholding changes the threshold dynamically over the image. It works well in a situation where image is affected by non-uniform ~~illumination~~ illumination problem.

Two techniques applied in the adaptive thresholding algorithm-

(1.) **Divide the image into many overlapping sections-**
In a non-uniform illuminated image, multiple different gray levels are overlapped with each other. It can be an object or background. For that reason, a single threshold value cannot divide the image properly. For that reason, in adaptive thresholding technique, an image is divided into sub-images and each subimage is uniquely thresholded for ~~the check~~ checking the object and background.

(2.) **Split the image into many subregions-**
In this ~~tec~~ adaptive thresholding technique, the non-uniform illumination problem is removed by grouping the pixels by using some statistical term (mean, median, mode, etc.) and then we take the similar regions to calculate the thresholding value.

## Region Growing :-

It is a procedure that groups pixels into sub regions or in bitmap. Logic-

• **Principle of Similarity :-**

→ It states that a region is coherent if all pixels of that region are homogeneous.

• **Major steps of region growing algorithm:-**

1. Selection of initial seed.
2. Use the seed growing criteria for region growing.
3. If no conditions are there, terminate the segmentation.

#

Example- • Q.) Separate regions using region growing.

image ↓

| | | | | |
|---|---|---|---|---|
| 1 | 0 | 7 | 8 | 7 |
| 0 | 1 | 8 | 9 | 8 |
| 0 | 0 | 7 | 9 | 8 |
| 0 | 1 | 8 | 8 | 9 |
| 1 | 2 | 8 | 8 | 9 |

5×5

Take seed value as the which is having high intensity or low intensity.

Let's say, $S_1$ (seed) = 8 and $S_2$ (seed) = 1.
Given, T (Threshold) = 4.

$S_1 = 8$ $\quad |f(x,y) - f(x', y')| \leq 4$

$|f(x,y) - 8 \leq 4.|$

Points where threshold matches {4, 5, 6, 7, 8, 9} → $\underline{\underline{A}}$

$S_2 = |f(x,y) - f(x', y')| \leq 4$

= {0, 1, 2, 3, 4, 5} → $\underline{\underline{B}}$

| | | | | |
|---|---|---|---|---|
| B | B | A | A | A |
| B | B | A | A | A |
| B | B | A | A | A |
| B | B | A | A | A |
| 1 | B | A | A | A |

2)

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 5 | 6 | 6 | 7 | 7 | 7 | 6 | 6 |
| 6 | 7 | 6 | 6 | 7 | 5 | 5 | 4 |
| 6 | 6 | 4 | 4 | 3 | 2 | 5 | 6 |
| 5 | 4 | 5 | 4 | 2 | 3 | 4 | 6 |
| 0 | 3 | 2 | 3 | 3 | 2 | 4 | 7 |
| 0 | 0 | 0 | 0 | 2 | 2 | 5 | 6 |
| 1 | 1 | 0 | 1 | 0 | 3 | 4 | 4 |
| 1 | 0 | 1 | 0 | 2 | 3 | 5 | 4 |

8×8

Given T=3

seed = 6 k

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| A | A | A | A | A | A | A | A |
| A | A | A | A | A | A | A | A |
| A | A | A | A | A | B | A | A |
| A | A | A | A | B | A | A | A |
| B | A | B | A | A | B | A | A |
| B | B | B | B | B | B | A | A |
| B | B | B | B | B | A | A | A |
| B | B | B | B | B | A | A | A |

- <u>Split and Merge Technique</u>- It is an alternative method of image segmentation. An image is subdivided into arbitrary disjoint region. Arbitrary regions can be split and merged in order to satisfy the condition. It is a two phase algorithm.

<u>Phase 1:</u> • Split and continue the sub-division until
• Some stopping criteria is fulfilled.
Stop when no other further splitting is possible.

<u>Phase 2:</u> • Merge adjacent regions if the regions share any common criteria.
- Stop the process when no further merging is possible.

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 5 | 6 | 6 | 6 | 7 | 7 | 6 | 6 |
| 6 | 7 | 6 | 7 | 7 8 | 5 | 5 4 | 7 |
| 6 | 6 | 4 | 4 | 3 | 2 | 5 | 6 |
| 5 | 4 | 5 | 4 | 2 | 3 | 4 | 6 |
| 0 | 3 | 2 | 3 | 3 | 2 | 4 | 7 |
| 0 | 0 | 0 | 0 | 2 | 2 | 5 | 6 |
| 1 | 1 | 0 | 1 | 0 | 3 | 4 | 4 |
| 1 | 0 | 1 | 0 | 2 | 3 | 5 | 4 |

$T \leq 3$

Highest pixel - lowest pixel should be under threshold value. If not then divide that region.

Phase 1:

If (Highest pixel - lowest pixel) <= T then no operation

Else split that region and repeat the process.

phase 2:

$(R_1 max - R_2 min) <= T$ &&

$(R_1 min - R_2 max) <= T$ → Then region will be demolished.

DIP-14 (TH) 16/11/22

Feature Extraction -

After image has been segmented into regions on their boundaries using some method and then the resulting sets of segmented image are usually converted into suitable form for further processing. This process is known as image feature extraction.

Feature description → Boundary correction + description

Feature detection → Object detection.

Feature extraction consists of two parts -

- Feature Description - Feature Description assigns ~~qualitative~~ quantative attribute to the detected feature. As an example, we might detect a ~~so~~ region boundary and the feature description will describe the corners by the orientation and location which are quantative values.

- Feature Detection - Feature detection refers to the features in an image, boundary or in a region which can be an object or any kind of information that we want to extract.

Important characteristics of features:

→ Image feature is an independent part of image processing that means the image feature does not depends on location, rotation, scale and luminance.

- Sensor characteristics - The image sensor decides how the view point will be captured and depending on the detailing, the feature is extracted from the image

- Image Preprocessing - In image preprocessing, the input image is normalized with all the features ~~sensored by~~ captured by the sensor. In the preprocessing technique, all the noise is reduced and all the features are properly visible.

* Feature extraction is invariant or co-variant. Explain.

Ans. In the context of image feature, the word 'independent' has two meaning : invariant and co-variant. A feature descriptor is invariant with respect to a set of transform if its values remains unchanged after application of any transformation. A feature descriptor is co-variant. with respect to a set of transformation if applying to the entity, any transformation from set of procedures produces the same result in descriptor with changed values in scale, reflection and rotation.

Q) what is local and global feature?

Ans. Local Feature : The feature information extracted from images in terms of numerical values that are difficult to understand and ~~correla~~ corelate by human. The local feature describe the ~~human~~ image depending on the Object which can be used as feature and only the subject area of that object are extracted for feature extraction. Local feature represents shape matrix, histogram orientation, etc.

Global Feature : Global feature describe the image as a whole to the generalised the ~~imag~~ entire Object. Global feature includes ~~control~~ contour representation, shape descriptor and texture feature.

Low-level application such as object detection, classification use global feature and object recognisation and pattern recognization needs high level processing. That's why it use ~~to~~ local feature.

Scanned by TapScanner

Spatial Image Processing - [2nd module Questions]

Q1) What is grey level transformation? [2]

2) Explain image negative. [2]

3) Perform image negative on the following image - [2]

$$\begin{bmatrix} 5 & 4 & 7 \\ 1 & 2 & 3 \\ 6 & 4 & 7 \end{bmatrix}$$

4) What is log transformation? [2]

5) Perform log transformation ~~and power l~~ on the above image. [3]

6) Perform power law transformation on the above image [3]

7) What is power law transformation? [2]

8) What is contrast? [2]

9) Explain contrast stretching. [2]

10) Explain the steps of contrast stretching. [3]

11) What ~~do~~ is bit plane slicing? [2]

12) Explain intensity-level slicing. [2]

Ans. In spatial transformation, there is a process in which we can highlight a specific range of intensities of an image, this process is known as intensity level slicing. This process is implemented using enhancing the darker regions which can provide an information.

13) What is histogram? [2]

14) What are the different types of histogram? [3]

15) What do you mean by histogram processing? [3]

16) ~~what~~ Explain histogram sliding with an example. [5]

17) What is histogram equalisation? [2]

18) Perform histogram equalisation on the following image - [5]

| Grey levels | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| Freq. | 15 | 10 | 5 | 8 | 6 | 5 | 6 | 9 |

19) what is histogram specification? [2]

20) Perform histogram specification on the following image. [2]

21) what do you mean by local processing? [2]

22) what is local enhancement? [3]

23) what do you mean by image subtraction? [3]

24) what is filter? [2]

25) what is noise in the image? [2]

26) ~~ω~~ Explain low level filter and high level filter. [2+2]

27) Explain smoothning filter. [2]

28) Explain different types of smoothning filter with their examples. [5]

29) what is sharpening filter? [2]

30) ~~what~~ Explain different types of sharpening filter with their example. [5]

31) what are the different types of pixels? [2]

32) ~~Ex~~ write a short note on RGB & grey image. [3+3]

33) what is transformation of image? write down the
~~34)~~ examples of different types of transformation. [2+1]

34) How rotation can be implemented in a pixel? [4]

35) Explain Shearing. How it can be implemented in pixels? [4]

36) Explain Hough Transformation. How line detection can be implemented using Hough Transformation. [2+3]

37) Explain the utility of fourier transformation. [2]

~~38)~~

Scanned by TapScanner

DIP-15(TH)

23/11/22

) what is contrast?

Ans. The term contrast refers to the amount of colour or grayscale differentiation that exists between various image features in both analog & digital images. The higher contrast level will display a greater degree of colour or grayscale variation than the lower contrast.

8) Local processing- A local processing is one in which the output image pixel value is also obtained from small area of pixels around the corresponding pixel value. This type of processing is applied when there is different types of grey levels are present in an image and for the processing purpose, we need to segment the image into different groups of pixels.

Piecewise Linear Transformation-

It is a type of ~~function~~ linear function that segments an image into different sections and each section or piece is transformed ~~to~~ with an arbitrary complex function. So, pixels of different segments are processed differently but there is a problem in the piecewise ~~pr~~ linear transformation, the transformed image requires more number of pixels because if the number of input is low, ~~they~~ then transformation of each segment is properly visible which creates bad output.

# Image segmentation [4th module]

Image segmentation is a process of partitioning a digital image into multiple regions and extracting the region which is known as ~~~~ ROI (region of interest).

* Region of Interest vary with application.
* No single universal function or algorithm exists for segmenting the ROI in all images. ~~~~
* Therefore, many segmentation algorithm need to apply and pick that algorithm which gives the best requirement.

There are two types of algorithm -

- Simple principle (Region approach) -
  Objective is to group pixels based on common property to extract the coherent region.

- Discontinuity Principle (Boundary Approach) -
  Objective is to extract regions that differs in property like intensity, color, texture.

## Characteristics of Segmentation process -

Let R ~~be~~ represent the entire image region and Segmentation is partitioning R into n subgroups which is represented as $R_i$.

- $R_i$ should be connected region.
- $R_i \in R$ where $i = 1 \ldots n$.
- ~~$R_j = \phi$ where $i \neq j$~~
- $R_i \cap R_j = \phi$ where $i != j$ ; $j = 1 \ldots n$

DIP-16(TH) 30/11/22

## Segmentation Algorithm -

Segmentation Algorithm
- Based on user interaction
- Based on pixel relationship
  - contextual (region based on global pixel relationship)
  - non-contextual (local based)

## Detection of discontinuity -

The image is always segmented depending on the change of grey level. The pixel values which differ in multiple region that is known as discontinuity.

There are three types of discontinuity -

1. Point discontinuity
2. Line discontinuity
3. Edge discontinuity

## • Point discontinuity detection -

An isolated point whose grey level is slightly different from its background in a homogeneous area is known as point discontinuity and that can be detected by point detection technique.

Suppose there is an image and it is represented as

| | | |
|---|---|---|
| $z_1$ | $z_2$ | $z_3$ |
| $z_4$ | $z_5$ | $z_6$ |
| $z_7$ | $z_8$ | $z_9$ |

and there is a point detection mask called -

| | | |
|---|---|---|
| $w_1$ | $w_2$ | $w_3$ |
| $w_4$ | $w_5$ | $w_6$ |
| $w_7$ | $w_8$ | $w_9$ |

we can check if there is any point or not using

$$R = \sum_{i=1}^{n} w_i z_i \cdot \text{ if } |R| >= T \text{ (Threshold value)}$$

then there is a point (discontinued point)

Simple mask for point detection-

| | | |
|---|---|---|
| -1 | -1 | -1 |
| -1 | 8 | -1 |
| -1 | -1 | -1 |

• **Line Detection Algorithm** -

There are 4 types of line detection possible. If in a continuous image, a single line can create the discontinuity that can be detected using line detection algorithm.

Suppose there is an image -

| | | |
|---|---|---|
| $z_1$ | $z_2$ | $z_3$ |
| $z_4$ | $z_5$ | $z_6$ |
| $z_7$ | $z_8$ | $z_9$ |

and there is a line detection mask =

| | | |
|---|---|---|
| $w_1$ | $w_2$ | $w_3$ |
| $w_4$ | $w_5$ | $w_6$ |
| $w_7$ | $w_8$ | $w_9$ |

we can check if there is any discontinuous line or not. using $R = \sum_{i=1}^{n} w_i z_i$. If $|R| >= T$, then there is a line.

**Sample Mask** -

Horizontal

| | | |
|---|---|---|
| -1 | -1 | -1 |
| 2 | 2 | 2 |
| -1 | -1 | -1 |

Vertical

| | | |
|---|---|---|
| -1 | 2 | -1 |
| -1 | 2 | -1 |
| -1 | 2 | -1 |

+45°

| | | |
|---|---|---|
| -1 | -1 | 2 |
| -1 | 2 | -1 |
| 2 | -1 | -1 |

-45°

| | | |
|---|---|---|
| 2 | -1 | -1 |
| -1 | 2 | -1 |
| -1 | -1 | 2 |

• **Edge detection Algorithm** -

An edge is a set of connected pixels that lies on the boundary between two regions which differ in grey-level. Pixels on the edge are known as edge points. Edge provides an outline of the object. In physical plane, edge corresponds to the discontinuity in depth, surface orientation, change in material properties and the light variation etc.

The edge detection algorithm -

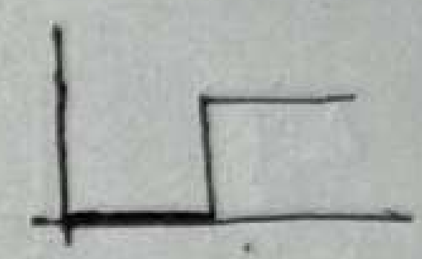
- Locates the sharp changes in intensity function.
- Edges are pixels where brightness changes unordely.

- an edge can be extracted by computing the derivative of the image function.
  - → magnitude of the derivative indicates the strength or contrast of the edge.
  - → direction of the derivative indicates edge orientation.

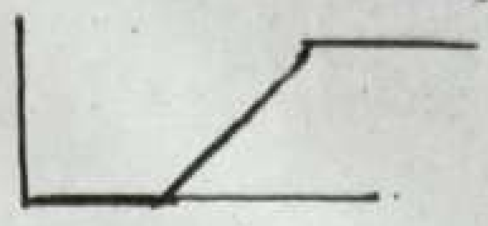
There are 4 types of ~~et~~ edge can be found-

~~Stiff edge~~

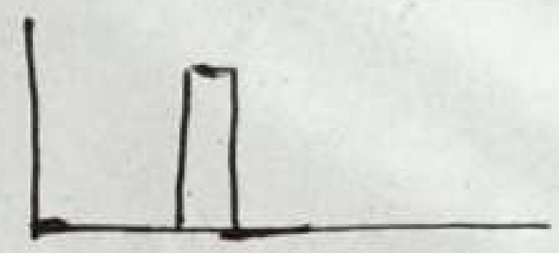
- Steep edge - An abrupt change in intensity.



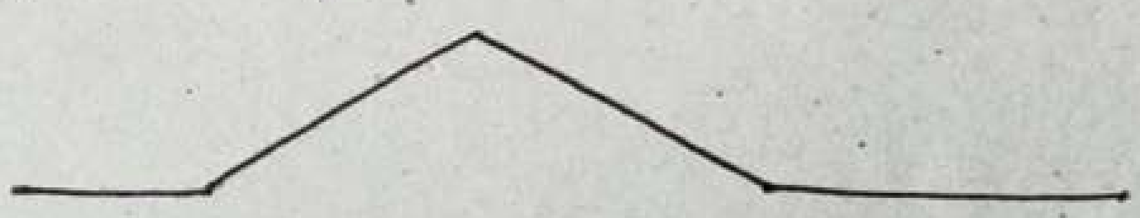
- Ramp edge - A slow and gradual change in intensity.



- Spike edge - Quick change or immediately returns to original intensity image.



- Roof edge - It is not ~~instanteous~~ an instant change. The change happens in short distance.



Steps of edge detection-

a) Input image - Here, the raw image is taken from the Sensor.

b) Filtering - It involves smoothing. This stage may be performed implicitly or explicitly.

c) Differentiation - It distinguish the edge pixels from other pixels. Suppose there is an image called $f(x,y)$.

$$\Delta f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$$

$$\text{Magnitude} = \sqrt{(G_x)^2 + (G_y)^2}$$

$$\cancel{\text{Phase}}\ \text{direction} = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

d) Localization - It determines the exact location of the edge.

e) Display - It provides us the actual edge points from an image.

First order Edge Detection Operator -

$$\Delta f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \partial f / \partial x \\ \partial f / \partial y \end{bmatrix}$$

↓
change

2            1

⊙           ⊙

→ $x - \Delta$       $x$ ← intensity

change in intensity

Backward Difference -

$$\frac{f(x) - f(x - \Delta x)}{\Delta x}$$

1            2

⊙           ⊙

$x$           $x+\Delta$

Forward Difference - $$\frac{f(x + \Delta x) - f(x)}{\Delta x}$$

Central Difference -

$$\frac{f(x + \Delta x) - f(x - \Delta x)}{2\Delta x}$$

~~Suppose, the value~~

Suppose the value of $\Delta x$ is 1.

$\therefore$ Backward difference = $f(x) - f(x-1)$
= $[1 \ -1]$ ← Mask

$\therefore$ Forward difference = $f(x+1) - f(x)$
= $[-1 \ +1]$ ← Mask

Changes → −1
no change − +1

$\therefore$ Central difference = $f(x+1) - f(x-1)$
= $[-1 \quad -1]$

## Robert Operator -

Robert Kernel are derivatives with respect to diagonal element. They are known as cross gradient operator. They are based on cross diagonal difference.

$f(x,y)$ , $f(x+1,y)$ → Two images.

$$G_x = \frac{\partial f}{\partial x} = f(x+1,y) - f(x,y).$$

$$\therefore G_x = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \qquad \cancel{G_y = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}}.$$

$f(x,y)$ , $f(x, y+1)$

$$G_y = \frac{\partial f}{\partial y} = f(x,y+1) - f(x,y)$$

$$G_y = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix}$$

## Algorithm of Robert Operator -

Step 1 - Read the image and smooth it.
Step 2 - convolve the image $f$ with $G_x$
Step 3 - convolve the image $f$ with $G_y$.
Step 4 - compute the edge magnitude and orientation
Step 5 - compare the edge magnitude with threshold value.
 a) If edge magnitude is higher, assign it as a possible edge point.

Scanned by TapScanner

## Prewitt Operator -

$$\frac{\partial f}{\partial x} = [f(x+1) - f(x-1)]/2$$ for central difference.

mask = $[-1 \quad 0 \quad +1]$

2 D representation

$$G_x = \frac{\partial f(x,y)}{\partial x} = [f(x+1,y) - f(x-1,y)]/2$$

$$= \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

← ~~along y-axis, so middle row is kept fixed.~~

$$G_y = [f(x,y+1) - f(x,y-1)]/2$$

$$= \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

← ~~along x-axis, so middle column is kept fixed.~~

DIP- 17(TH) 7/12/22

## Sobel Operator - edge detection

$$\text{Mask} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \times \begin{bmatrix} -1 & 0 & +1 \end{bmatrix} = \begin{bmatrix} -1 & 0 & +1 \\ -2 & 0 & +2 \\ -1 & 0 & +1 \end{bmatrix}$$

↑ weighted average — ↑ Prewitt mask (x- derivative) — ← Sobel mask Sobel -x / $G_x$

It enhances the edge better than prewitt operator.

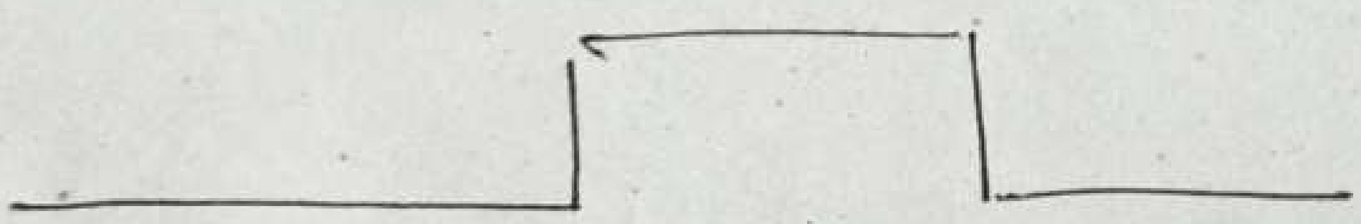
$$G_y = \begin{bmatrix} 1 \\ 0 \\ -1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \end{bmatrix} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

Sobel -Y.

## Second Order Derivative -

In the first derivative, edges are considered to be present when ~~the~~ edge magnitude is larger than the threshold value.

In case of second order derivative, edges are present at the location where the value of derivative is zero.



$20 \left[ \begin{matrix} 20 \\ 40 \end{matrix} \right.$

$-20 \left[ \begin{matrix} 20 \end{matrix} \right.$

It is a zero crossing point which can be observed as a change in sign. That ~~ma~~ means it is an edge point.

## Laplacian algorithm -

$$\Delta^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial^2 x} + \frac{\partial^2 f(x,y)}{\partial^2 y}$$

$$\frac{\partial^2 f(x,y)}{\partial^2 x} = f(x+1,y) + f(x-1,y) - 2f(x,y).$$

$$\frac{\partial^2 f(x,y)}{\partial^2 y} = f(x,y+1) + f(x,y-1) - 2f(x,y)$$

$$\Delta^2 f(x,y) = f(x,y+1) + f(x,y-1) + f(x+1,y) + f(x-1,y) - 4f(x,y)$$

Scanned by TapScanner

∴ Laplacian mask / Laplacian filter = 

| | | |
|---|---|---|
| 0 | 1 | 0 |
| 1 | -4 | 1 |
| 0 | 1 | 0 |

Convolution :- Convolution is a process to apply a filter/Kernal on the image in spatial domain.

Steps :-

1. Flip the Kernel in both horizontal and vertical direction.
2. Move over the array with the kernel centered at interested point.
3. Multiply the kernel data with overlapped area.
4. Sum and accumulate the output.

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix} \times \underbrace{\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}}_{\text{Mask}} \xrightarrow[\text{Vertical flip}]{\text{Horizontal flipe}} \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

~~Now,~~
~~add zeros to ~~the~~ ~~first~~ each pixel, eg.-~~

$$\cancel{\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 1 & 2 & 3 \\ 0 & 4 & 5 & 6 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.}$$

$$\cancel{\Rightarrow \begin{bmatrix} 13 & 20 & 17 \\ & & \end{bmatrix}}$$

Now,

$$\boxed{\begin{matrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{matrix}} \times \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 2 \\ 0 & 4 & 5 \end{bmatrix} \cdots\cdots$$

$$= \begin{bmatrix} 13 & 20 & 17 \\ & & \end{bmatrix}$$

18/12/22

DIP- 19 (TH)

~~Hough~~

## Huffman coding–

It is a type of variable length coding. Here all the histograms are defined using some different binary values and that mention the probability of histogram.

## Huffman coding Algorithm–

Step 1– List the symbols ~~the~~ and sort the probabilities.

Step 2– combine the lowest two probabilities and label using new code.

Step 3– Newly created item is given priority and place it on the highest position of the ~~short~~ sorted list.

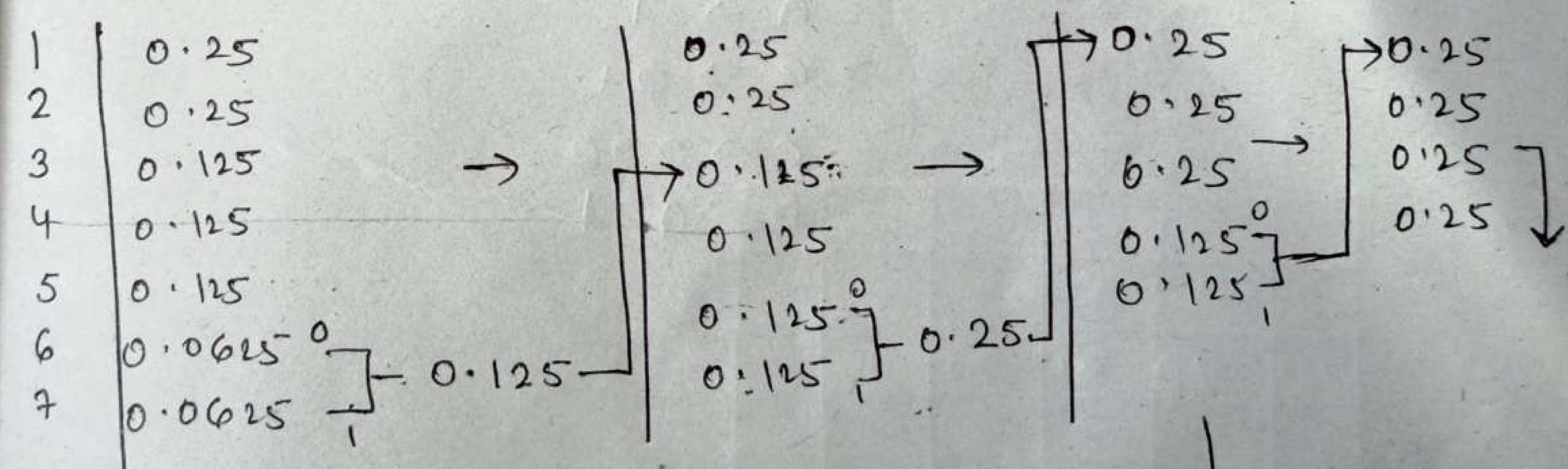
Step 4– Repeat step 2 until one node remains.

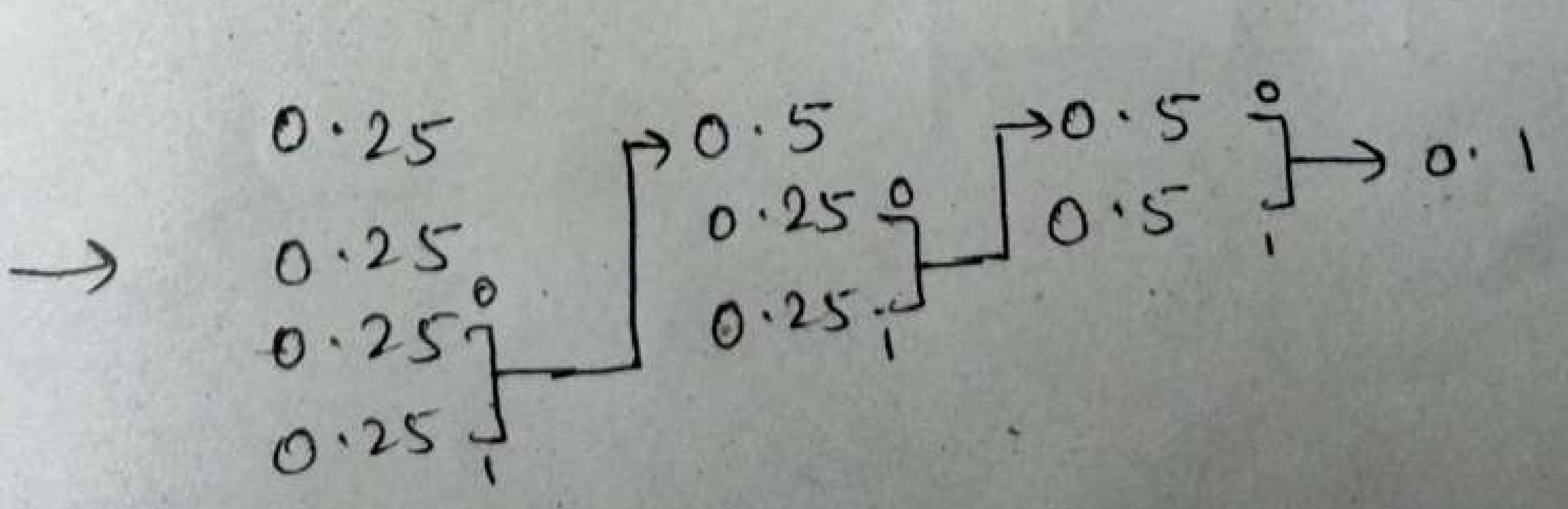
Step 5– Assign code 0 to higher up symbol and 1 to lower down symbol.

Step 6– Now trace the symbol backward.

### Example-1 –

| $S_0$ | $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ | $S_7$. |
|---|---|---|---|---|---|---|---|
| 0 | 0.25 | 0.25 | 0.125 | 0.125 | 0.125 | 0.0625 | 0.0625 |



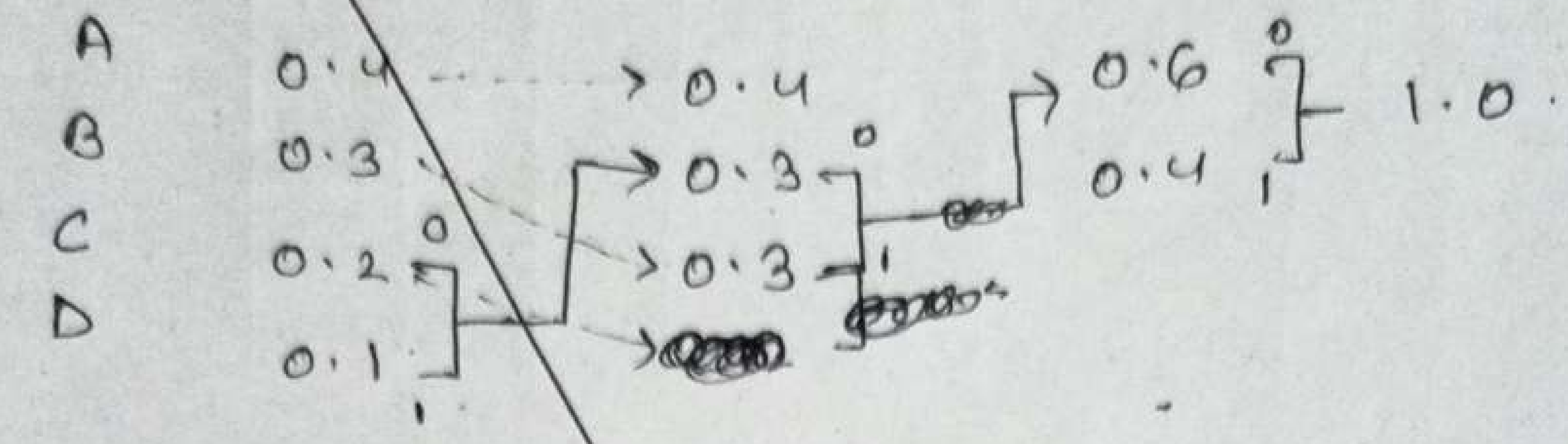

$S_1 = 0$
$S_2 = 01$
$S_3 = 01$
$S_4 = 01$
$S_5 = 01$
$S_6 = 01$
$S_7 = 1$

| $S_1$ | $S_2$ | $S_3$ | $S_4$ | $S_5$ | $S_6$ | $S_7$ |
|---|---|---|---|---|---|---|
| 0 | 10 | 10 | 10 | 10 | 10 | 1 |

backtracked so 01 becomes 10.

~~Ex 2~~

| A | B | C | D |
|---|---|---|---|
| 0.4 | 0.3 | 0.2 | 0.1 |

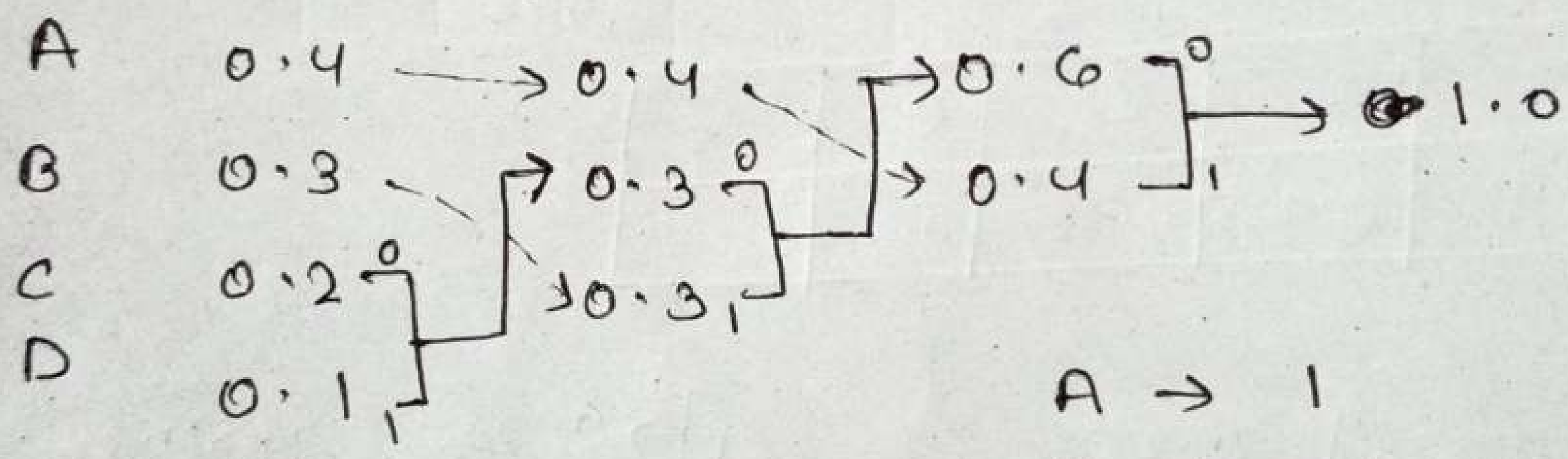



A - 0
B - 01
C - 01
D - 1

∴ 0 10 10 1 ← after backtracking.

## Ex 2

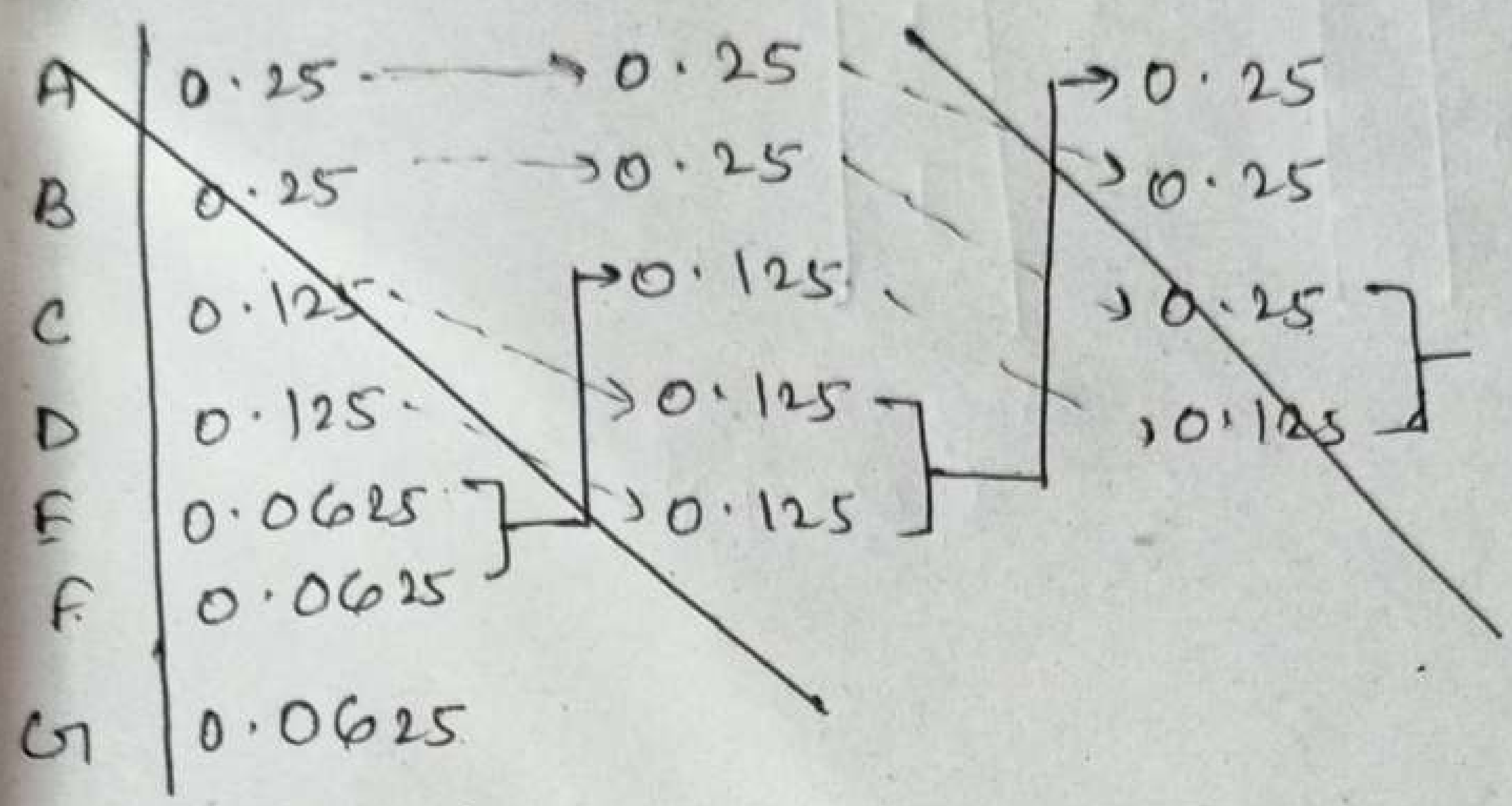
| A | B | C | D |
|---|---|---|---|
| 0.4 | 0.3 | 0.2 | 0.1 |

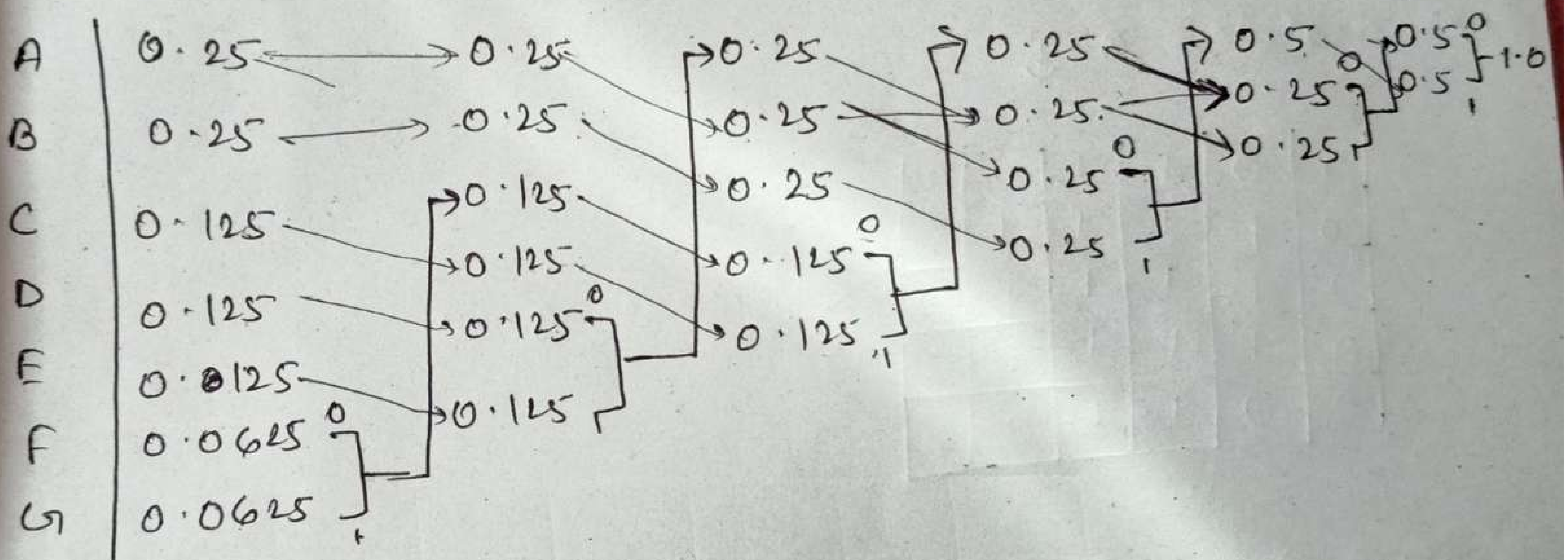



A → 1
B → 01
C → 000
D → 001

Ex-3 : Probability of 7 sources are given below-

| A | B | C | D | E | F | G |
|---|---|---|---|---|---|---|
| 0.25 | 0.25 | 0.125 | 0.125 | 0.0625 | 0.0625 | 0.0625 |




A → 10
B → 11
C → 001
D → 010
E → 011
F → 0000
G → 0001

# Dilation & Erosion.

A

| | 0 | | | | | |
|---|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 1 | 1 | 1 | 1 | 0 |
| | 0 | 0 | 1 | 1 | 0 | 0 |
| | 0 | 0 | 0 | 0 | 0 | 0 |

Image

B

| |
|---|
| 1 |
| 1 |
| 1 |

structuring element (Mask):

Full match → 1
some match → 1
No match → 0.

A dilated B ↓

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 1 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |

← image grown.

## Erosion →

full match → 1
some match → 0
No match → 0

A eroted B →

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 1 | 1 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 | 0 |

← Image shrinked