Date:

~~Pract~~

Iterative Modification:

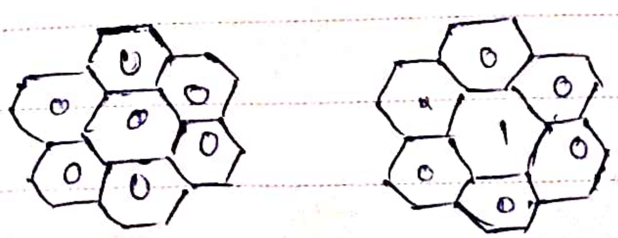
- The ways in which you can iteratively modify a binary image to extract some info from it
- When you modify the image, you want to make sure you dont change the structure, the integrity of binary image.
- Euler Number (E):

No. of bodies (B) − No. of holes (H).

$E_{image} = \sum E_{non\text{-}overlapping\ regions}$.

If you are applying an operation to one region, to each region which does not change the Euler number, then the end of the day, the Euler no. of the complete image remains the same.

Euler Differential (E^*):



~~Euler=0~~

$E=0$ $\quad$ $E=1$.

$\therefore \boxed{E^*=1}$

- Neighbourhoods Sets Based on E^*.
  Each pixel has $2^6 = 64$ possible neighbourhoods. Neighbourhoods patterns are classified based on Euler Differential they generate, assuming the centre pixel goes from 0 to 1
- Only 4 possible neighbourhood types:
  $N_{+1}, N_0, N_{-1}, N_{-2}$.
- Then do this in parallel to all of the points in the image & with the O/P that you get, you can then take that in as an I/P again & reapply the algorithm to that image, & you get an output. You can iteratively modify the image.
- When you are doing it in parallel, make sure that when you're modifying your pixel, that pixel is not being using as a neighbour for any other pixel.
- Conservative operations do not change the Euler no. of the image.

- Notation for iterative Modification: we specify a neighbourhood set that we want to act on

Date:

consider pixel (i, j)

- ~~a_{ij}~~ &
- $a_{ij} = 1$ if ~~be~~ neighbourhood of $(i,j) \in S$ else 0.

- b_{ij} : current value of pixel (i,j)
- c_{ij} : new value of pixel (i,j).

a_{ij}	b_{ij}		c_{ij}
0	0	$\Rightarrow$	?
0	1		?
1	0		?
1	1		?

These are binary no. & you can fill these 4 slots with zeros & ones.

$C_{ij} = 2^4 = 16$ algorithms.

$\longleftarrow$ 16 algorithms $\longrightarrow$

a_{ij}	b_{ij}	0	1	2	3	$\boxed{4}$	5	6	$\boxed{7}$	8	9	10	11	12	13	14	15
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
1	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

- Growing objects : $S \in$ No & Algorithm 7
- Thinning objects : $S \in$ No & Algorithm 4.