ARAVIND
2019102074

8/10/2021

DIP Lecture 13

Morphological Operations & Intro to Geometric operations

Erosion:

Shrinks connected sets of 1's of binary image

Used for:

1) Shrinking features
2) Removing bridges protrusions etc & branches

3) Foreground holes are enlarged
4) $f \ominus s$ → Representation

Min filter

Dilation:

→ Expands foreground objects

→ Foreground holes are shrunk

→ Representation: $f \oplus \hat{s}$

→ Max filter

→ No change in SE after reflection if symmetric

Boundary extraction:

Ex. Boundary: $(A \oplus B) - A$

In. Boundary: $A - (A \ominus B)$

Morphological operation $(A \oplus B) - (A \ominus B)$

Opening: (Erosion then dilation) $f \circ s = (f \ominus s) \oplus s$

Closing: (Dilation then erosion) $f \bullet s = (f \oplus s) \ominus s$

} Idempotent & Dual → Repeat has no effect

Morphological smoothing can be achieved by opening followed by closing

Dilation & Closing are extending operations

Erosion & Opening are narrowing operations

Erosion can be used for pattern matching [Fixed template case]

Hit or miss transforms:

→ Look for particular patterns of foreground & bg pixels

→ Use don't care (X) cases

→ If matched set pixel = 1

→ Representation: $A \circledast B = (A \ominus B_1) \cap (A^c \ominus B_2)$

$A \ominus B_1$ → Associated with fg → Hit

$A^c \ominus B_2$ → Associated with bg → Miss

Hit, Miss → Shape Detection

→ Distance transforms:

→ Intensities in fg now show distance from each point to closest background/boundary pixel

→ L_∞ = Chessboard distance metric

$$DT(P)[x] = \min_{y \in P} D(x,y)$$

$$D_{chess}(\{a,b,c\},\{x,y,z\}) = \max\{|a-x|, |b-y|, |z-c|\}$$

→ Inefficient way:
Successive erosions

Application: <u>Skeletonisation</u>

Two pass algorithm:

Look at top and side labels and assign new labels accordingly. Go row by row to identify connected components.

Execute the loop again by replacing child label with root label.

→ Uses union-find data structure ensures 'find'-ing $O(\log^* n)$ converges to $O(1)$ for repeated calls

Flood fill:

→ Old colored pixels with fill target color
→ 4 or 8 connectivity
→ Parameters
Target fill, old color, coordinates
→ Recursively call fn

Geometric operations:

→ Zooming images, h
→ Computer graphics
→ Coordinates are changed rather than pixel intensities

$$I(x,y) \rightarrow I'(x',y')$$

→ Example Shifting

$$x' = f_x(x,y)$$
$$y' = f_y(x,y) \Rightarrow I(x,y) \rightarrow I(x',y')$$

→ Operations:

→ Scale
→ Rotate
→ Reflect
→ Translate
→ Shear

→ Affine transformation

Rotate: $T_x : x' = c\alpha\, x - s\alpha\, y$
$T_y : y' = s\alpha\, x + c\alpha\, y$

Scale: $T_x : x' = S_x x$
$T_y : y' = S_y \cdot y$

where s_x, s_y are scaling factors

Translate: $T_x : x' = x + dx$
$T_y : y' = y + dy$

Shear: $T_x : x + b_x y$
$T_y : y + b_y x$

Shrinking:

Removing certain pixels via pixel selection or interpolation

Stretching:

Pixels are added via replication (or) interpolation

Homogenous coordinates:

→ Useful for converting scaling, translation, rotating into point-matrix multiplication

→ $x = \begin{pmatrix} x \\ y \end{pmatrix}$ converted to $\hat{x} = \begin{pmatrix} \hat{x} \\ \hat{y} \\ h \end{pmatrix} = \begin{pmatrix} hx \\ hy \\ h \end{pmatrix}$