Grayscale morphology and distance transforms

**Grayscale dilation**
- Choose the max value of $f \circ b$
- Effect:
  - $b$ with positive elements $\rightarrow$ (often) brighter output
  - Dark details are reduced or eliminated

**Erosion**
- Choose the min value of $f \bullet b$
- Effect:
  - $b$ with positive elements $\rightarrow$ (often) darker output
  - Bright details are reduced

**Effects**
- Opening
  - Remove small bright details
  - Leave overall grey-levels
  - Leave larger bright features

- Closing
  - Remove dark details
  - Leave bright features
**Binary Properties**

- Duality with respect to complementation and reflection:
  \[ (A \cap B)^c = (A^c \cup B^c) \]
  - Opening:
    - \( A \cap B \) subset/image of \( A \)
    - \( C \subseteq D \) \( C \subseteq B \subseteq D \)
    - \( (A \cap B) \subseteq A \cap B \)
  - Closing:
    - \( A \cup B \) subset/image of \( A \)
    - \( C \subseteq D \) \( C \subseteq B \subseteq D \)
    - \( (A \cup B) \subseteq A \cup B \)

**Grey-Level Properties**

- Duality with respect to complementation and reflection:
  \[ (f \circ b)^c = f^c \circ b^c \]
  - Opening:
    - \( f \circ b \) if \( f_1 \circ b \), then \( (f_1 \circ b) \cap (f_2 \circ b) \)
    - \( (f_1 \circ b) \cap (f_2 \circ b) \)
    - \( (f_1 \cap f_2) \cap (f_1 \cap f_2) \)

**Application: Smoothing**

Can be achieved by opening followed by closing → removal or attenuation of bright & dark artifacts/noise

**Application: Gradient Image**

\[ g = (f \circ b) - (f \circ b) \]
subtracted eroded image from dilated

**Distance Transforms**

Input: binary image
Output: in each object (background) pixel, write the distance to the closest background (object) pixel.

Def. A function \( D \) is a metric (distance measure) for the pixels \( p, q \) and \( z \) if

- a) \( D(p,q) = 0 \) (if \( p = q \))
- b) \( D(p,q) = D(q,p) \)
- c) \( D(p,z) \leq D(p,q) + D(q,z) \)

**Different Metrics**

- Euclidean \( D_E(p,q) = \)
- City block \( D_C(p,q) = \)
- Chessboard \( D_8(p,q) = \)

- Weighted Measures:
  - Chamfer(3-4)
  - Chamfer(5-7-11)
Algorithm for distance transformation
(distance from each object pixel to the closest background pixel)
1. Set background pixels = 0 and object pixels = max (e.g. 255)
2. Forward pass, from (0,0) to (max(x), max(y)):
   Mask:
   \[ \text{if } p > 0, \ p = \min(g_i + w_i), \ i = 1, 2, 3, 4 \]
3. Backward pass, from (max(x), max(y)) to (0,0):
   Mask:
   \[ \text{if } p > 0, \ p = \min(p, \min(g_i + w_i)), \ i = 1, 2, 3, 4 \]

1. Applications of the distance transform (DT)
i) Find the shortest path between two points a and b
   1. generate the DT with a as the object
   2. go from b in the steepest gradient direction

ii) Find the radius of a round object
   1. generate the DT of the object
   2. max value = radius

* Segmentation using watershed!

iv. Skeletons

Definitions:
O = object
B = background
S = skeleton
Then:
\( S \) topological equivalent to \( O \)
\( S \) centered in \( O \)
\( S \) one pixel wide (difficult)
\( O \) can be reconstructed from \( S \)

Skeletons (Centers of Maximal Discs)
A disc is made up of all pixels that are within a given radius \( r \). The skeleton of a binary object is the union of all disc centers needed to reconstruct the object using the corresponding discs.

Algorithm: Find the skeleton, Centers of Maximal Discs (CMD)
* reversible
1. generate DT of object
2. Identify CMDs (smallest set of maxima)
3. Link CMDs

*Pruning* = remove small branches (no longer fully reversible)
Skeletonisation based on thinning (not reversible)

Skeleton using Chamfer (3,4) DT, no pruning (fully reversible)

Skeleton using Chamfer (3,4) DT, followed by pruning (not fully reversible)

Skeleton