Binary image processing

- Binary images are common
  - Text and line graphics, document image processing
  - Often an intermediate abstraction in an image analysis system
    - Object boundaries
    - Object location
    - Presence/absence of some image property
- Representation of individual pixels as 0 or 1, convention:
  - foreground, object = 1 (white)
  - background = 0 (black)
- Processing by logical functions is fast and simple
- Special class shift-invariant operation that change the shape of regions: *morphological image processing*
- Morphological image processing has been extended to gray-level images

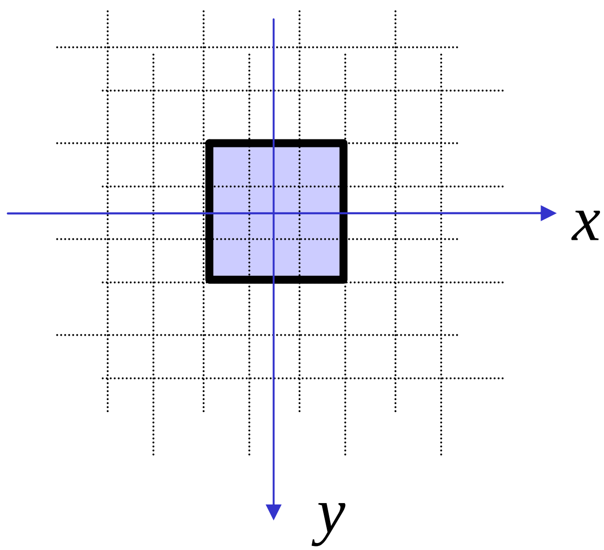
Binary morphological operators

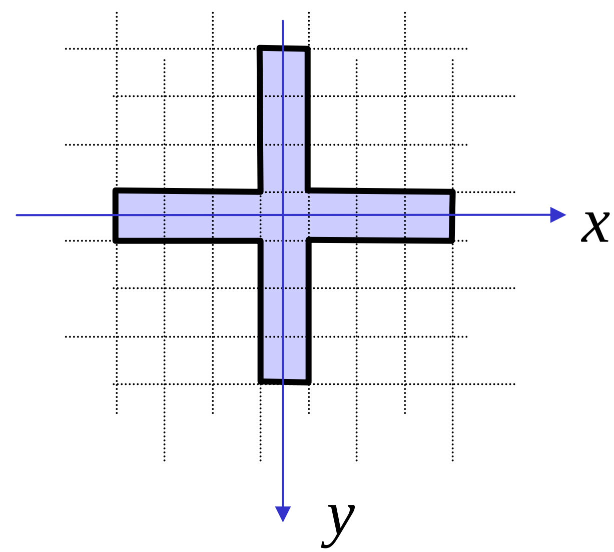
- Window operator

$$W\{f(x,y)\} = \{f(x-x', y-y'); (x', y') \in \Pi_{xy}\}$$

"structuring element"

- Example structuring elements Π_{xy}:



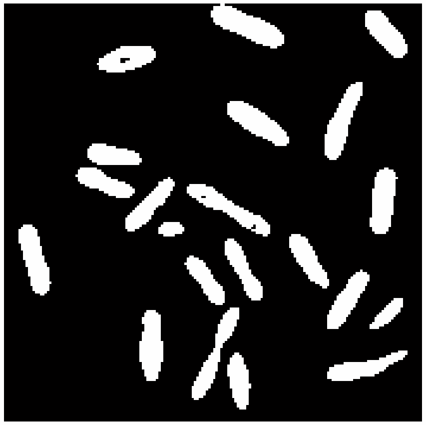


Dilation

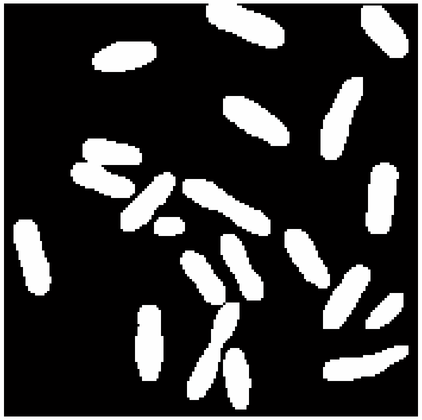
- Binary dilation operator

$$g(x,y) = OR\left[W\left\{f(x,y)\right\}\right] := dilate\left(f, W\right)$$

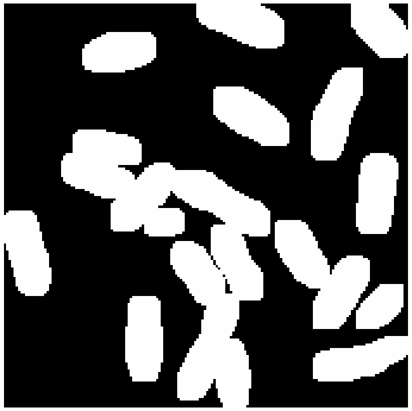
- Effects
  - Expands the size of 1-valued objects
  - Smoothes object boundaries
  - Closes holes and gaps



Original (178x178)



dilation with
3x3 structuring element



dilation with
7x7 structuring element

Erosion

- Binary erosion operator

$$g(x,y) = AND\left[W\{f(x,y)\}\right] := erode(f,W)$$

- Effects
  - Shrinks the size of 1-valued objects
  - Smoothes object boundaries
  - Removes peninsulas, fingers, and small objects
- Relationship with dilation
  - Duality: erosion is dilation of the background

$$dilate(f,W) = NOT\left[erode\left(NOT[f],W\right)\right]$$

$$erode(f,W) = NOT\left[dilate\left(NOT[f],W\right)\right]$$

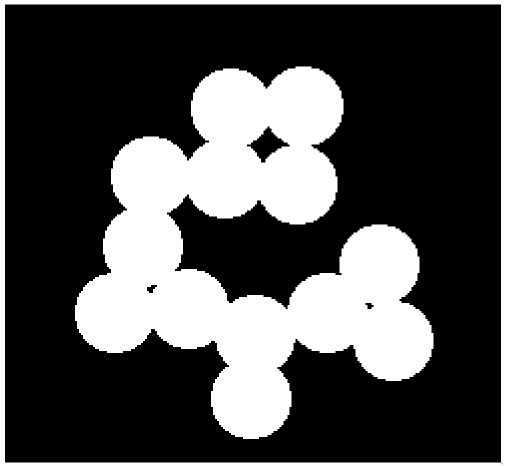
  - But: erosion is <u>not</u> the inverse of dilation

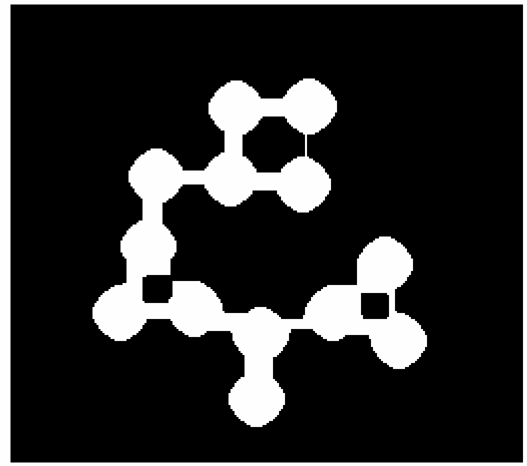
$$f(x,y) \neq erode\left(dilate(f,W),W\right)$$

$$\neq dilate\left(erode(f,W),W\right)$$

Example: blob separation/detection by erosion

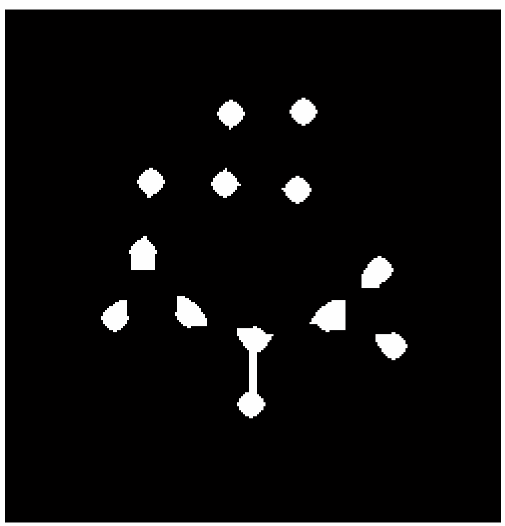
Original binary image *circles*

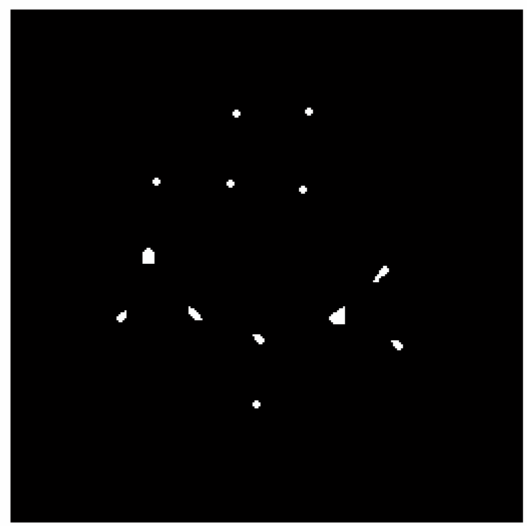




Erosion by 11x11 structuring element

Erosion by 21x21 structuring element





Erosion by 27x27 structuring element

Set-theoretic interpretation

- Set of object pixels

$$F \equiv \{x, y : f(x, y) = 1\}$$

- Background: complement of foreground set

$$F^c \equiv \{x, y : f(x, y) = 0\}$$

- Dilation is Minkowski set addition (1903)

$$G = F \oplus \Pi_{xy}$$

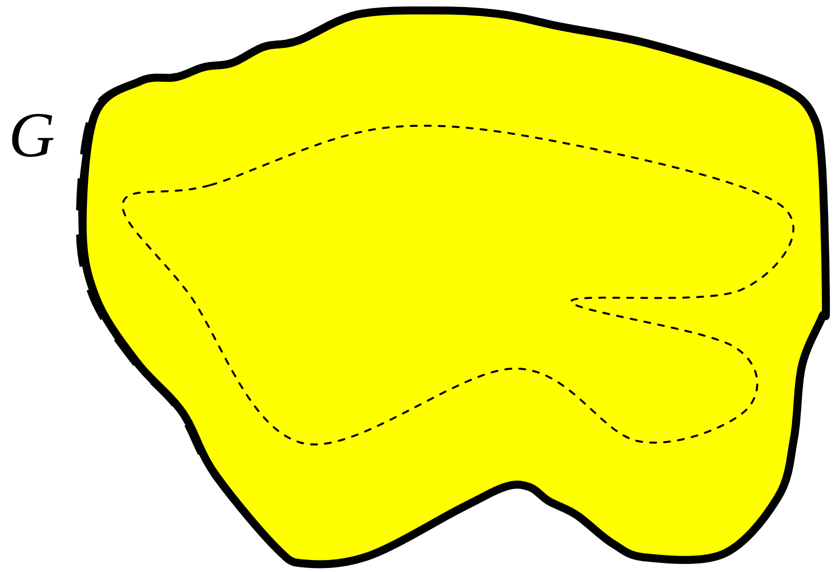
Commutative and associative!

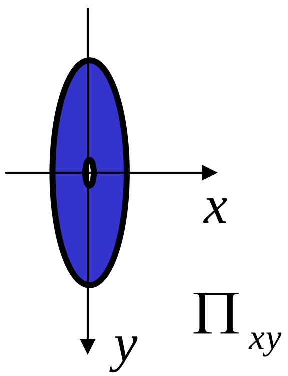
$$= \{(x + p_x, y + p_y) : (x, y) \in F, (p_x, p_y) \in \Pi_{xy}\}$$

$$= \bigcup_{(p_x, p_y) \in \Pi_{xy}} F_{+(p_x, p_y)}$$

translation of F by vector (p_x, p_y)

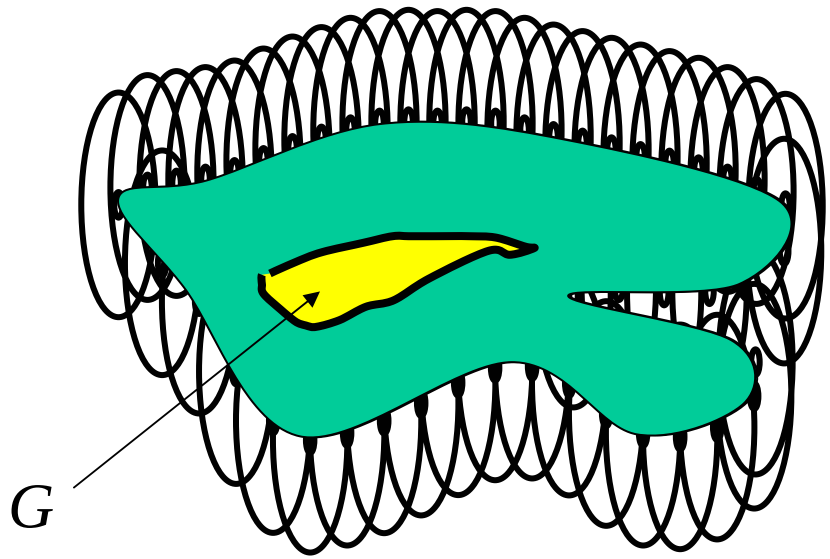
Set-theoretic interpretation: dilation

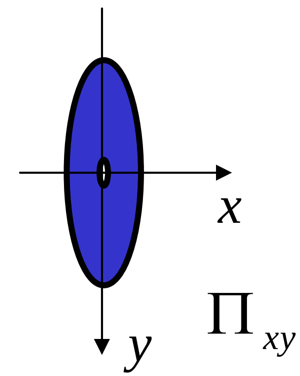




$$
\begin{aligned}
G &= F \oplus \Pi_{xy} \\
&= \left\{ \left(x + p_x, y + p_y \right) : \left(x, y \right) \in F, \left(p_x, p_y \right) \in \Pi_{xy} \right\} \\
&= \bigcup_{\left(p_x, p_y \right) \in \Pi_{xy}} F_{+\left(p_x, p_y \right)}
\end{aligned}
$$

Set-theoretic interpretation: erosion





Not commutative!
Not associative!

Minkowski set subtraction

$$G = \bigcap_{(p_x, p_y) \in \Pi_{xy}} F_{+(p_x, p_y)} = F \ominus \Pi_{-xy}$$

Reversed structuring element

Opening and closing

- Goal: smoothing without size change
- Open filter

$$open(f,W) = dilate(erode(f,W),W)$$

- Close filter

$$close(f,W) = erode(dilate(f,W),W)$$

- Open and close filter are biased
  - Open filter removes small 1-regions
  - Close filter removes small 0-regions
  - Bias is often desired for enhancement or detection!
- Unbiased size-preserving smoothers

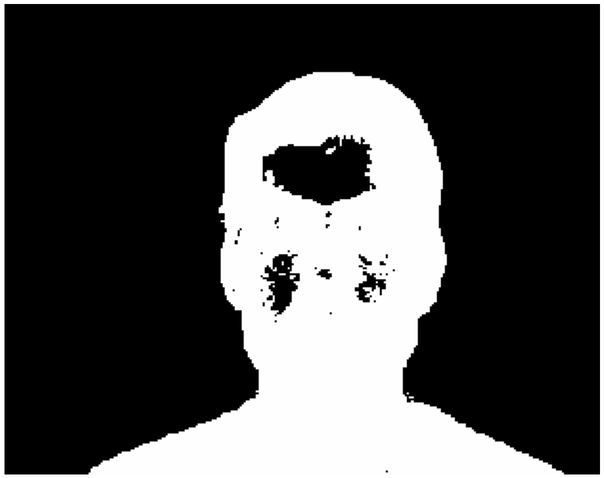
$$close-open(f,W) = close(open(f,W),W)$$

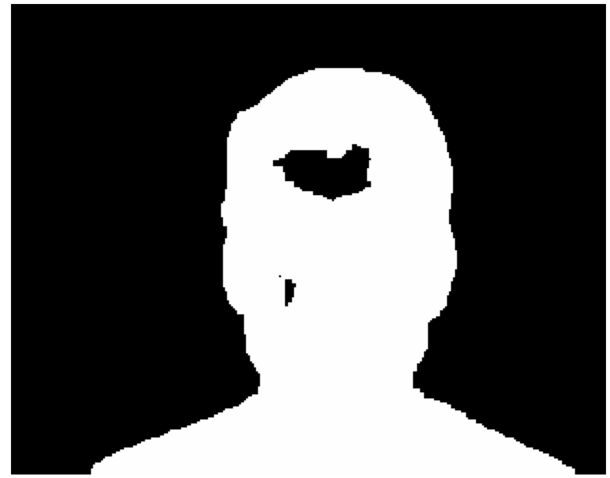
$$open-close(f,W) = open(close(f,W),W)$$

- *close-open* and *open-close* are duals, but not inverses of each other.

Small hole removal by closing

Original binary mask



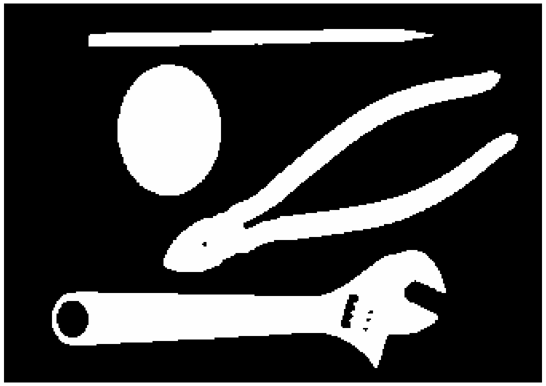


Dilation 5x5

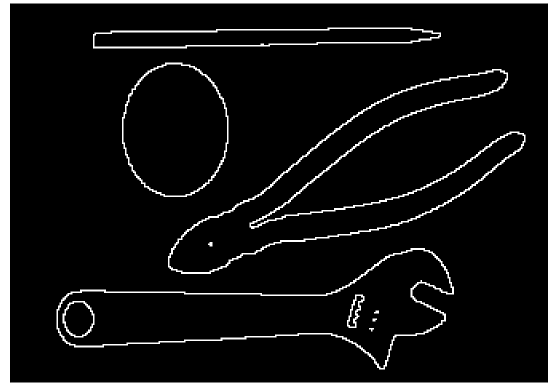
Difference to original mask

Closing 5x5

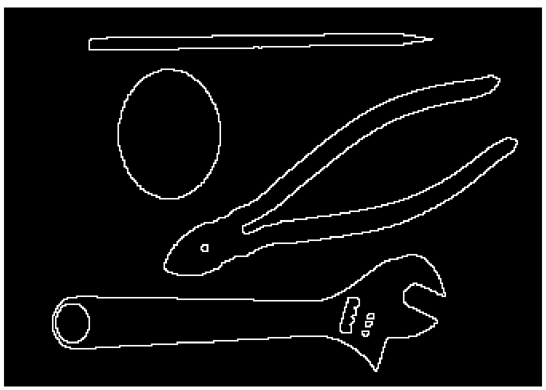
Morphological edge detectors



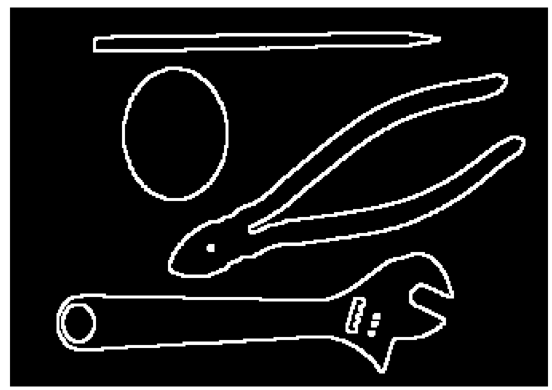
F



$(F \oplus \Pi) / F$



$F / (F \Theta \Pi)$



$(F \oplus \Pi) / (F \Theta \Pi)$

Courtesy: P. Salembier

Majority filter

- Binary majority filter

$$g(x,y) = MAJ\left[W\left\{f(x,y)\right\}\right] := majority\left(f, W\right)$$

- Effects
  - Does not generally shrink or expand objects
  - Smoothes object boundaries
  - Removes small peninsulas, bays, small objects, and small holes
  - Less biased than *close-open* or *open-close*
- Self-duality

$$majority\left(f, W\right) = NOT\left[majority\left(NOT\left[f\right], W\right)\right]$$

- Special case of a gray-level median filter

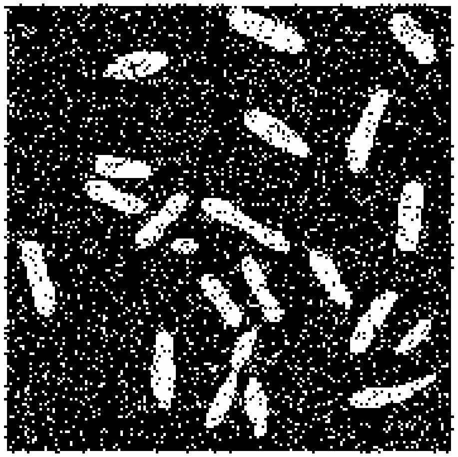
$$g(x,y) = median\left[W\left\{f(x,y)\right\}\right] := median\left(f, W\right)$$

Majority filter: example

Binary image with 5% ‘Salt&Pepper’ noise

3x3 majority filter

20% ‘Salt&Pepper’ noise



3x3 majority filter

Image corrupted by “Salt&Pepper” noise

Original image

5% bits flipped randomly

Noise removal by median filtering

3x3 median filtering

7x7 median filtering

Morphological filters for gray-level images

- Thresholds sets of a gray-level image $f(x,y)$

$$\mathrm{T}_\theta\left(f(x,y)\right)=\left\{(x,y): f(x,y)\geq\theta\right\},\qquad -\infty<\theta<+\infty$$

- Reconstruction of original image from threshold sets

$$f(x,y)=\sup\left\{\theta:(x,y)\in\mathrm{T}_\theta\left(f(x,y)\right)\right\}$$

- Idea of morphological operators for multi-level (or continuous-amplitude) signals
  - Decompose into threshold sets
  - Apply binary morphological operator to each threshold set
  - Reconstruct via supremum operation
  - Gray-level operators thus obtained: ***flat operators***

Dilation for gray-level images

- Explicit decomposition into threshold sets not required in practice
- Flat dilation operator

$$g(x,y) = \sup\left\{W\left\{f(x,y)\right\}\right\} := dilate\left(f, W\right)$$

- Local maximum for discrete images and finite window
- Binary dilation operator contained as special case
- General dilation operator

$$\begin{aligned} g(x,y) &= \sup_{\alpha,\beta}\left\{f\left(x-\alpha, y-\beta\right) + w\left(\alpha,\beta\right)\right\} \\ &= \sup_{\alpha,\beta}\left\{w\left(x-\alpha, y-\beta\right) + f\left(\alpha,\beta\right)\right\} \end{aligned}$$

- Like linear convolution, with $\sup$ replacing summation, addition replacing multiplication

Unit impulse for dilation

- If dilation is a nonlinear convolution, with $\sup$ replacing summation, addition replacing multiplication, which signal corresponds to the unit impulse?
- Formally: find $d(\alpha, \beta)$ such that

$$f(x,y) = \sup_{\alpha,\beta} \left\{ f\left(x-\alpha, y-\beta\right) + d\left(\alpha,\beta\right) \right\}$$

- Answer:

$$d(\alpha,\beta) = \begin{cases} 0 & \alpha = \beta = 0 \\ -\infty & \text{else} \end{cases}$$

Flat dilation as a special case

- Find $w(\alpha,\beta)$ such that

$$f(x,y)=\sup_{\alpha,\beta}\left\{f\left(x-\alpha,y-\beta\right)+w\left(\alpha,\beta\right)\right\}=dilate\left(f,W\right)$$

- Answer:

$$w(\alpha,\beta)=\begin{cases}0 & (\alpha,\beta)\in\Pi_{xy}\\ -\infty & \text{else}\end{cases}$$

- Hence, write in general

$$g(x,y)=\sup_{\alpha,\beta}\left\{f\left(x-\alpha,y-\beta\right)+w\left(\alpha,\beta\right)\right\}$$
$$=dilate(f,w)=dilate(w,f)$$

Erosion for gray-level images

- Flat erosion operator

$$g(x,y) = \inf\left\{W\left\{f(x,y)\right\}\right\} := erode\left(f, W\right)$$

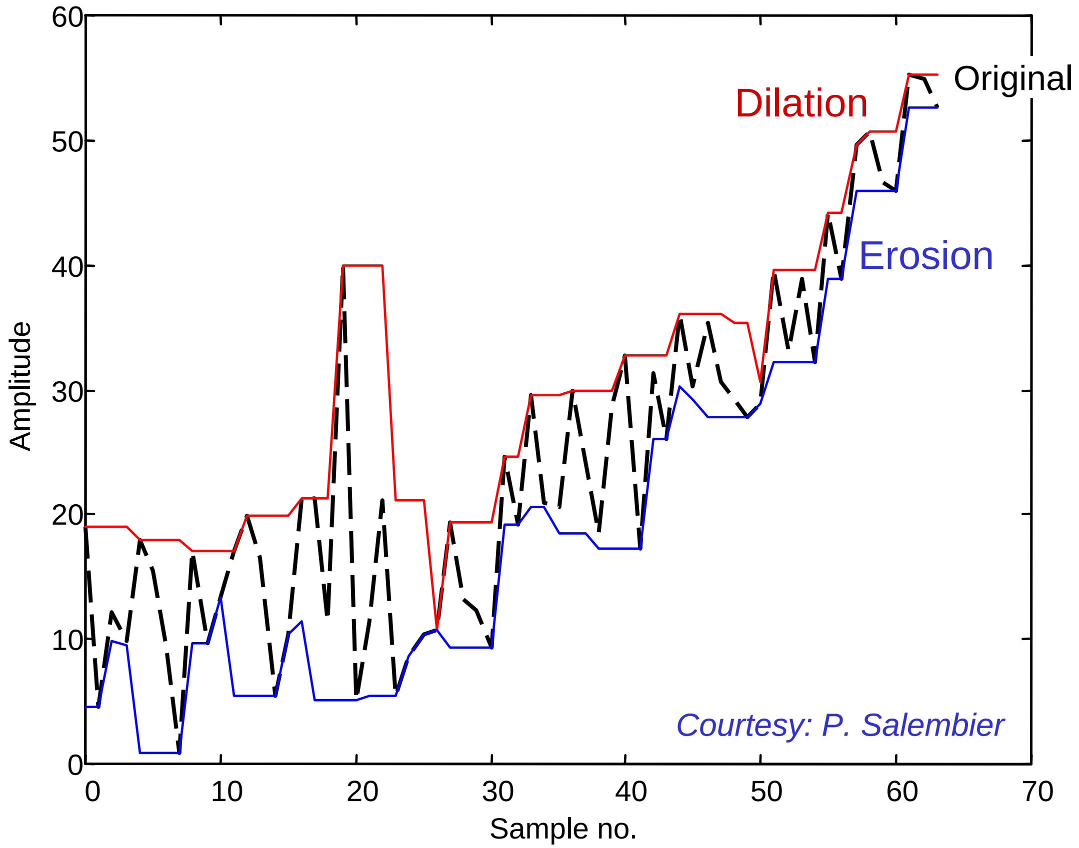
- Local minimum for discrete images and finite window
- Binary erosion operator contained as special case
- General erosion operator

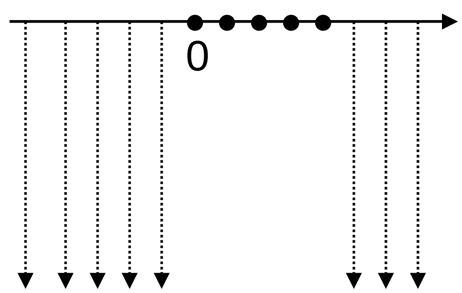
$$g(x,y) = \inf_{\alpha,\beta}\left\{f\left(x-\alpha, y-\beta\right) - w\left(\alpha,\beta\right)\right\} = erode\left(f, w\right)$$

- Dual of dilation

$$\begin{aligned} g(x,y) &= \inf_{\alpha,\beta}\left\{f\left(x-\alpha, y-\beta\right) - w\left(\alpha,\beta\right)\right\} \\ &= -\sup_{\alpha,\beta}\left\{-f\left(x-\alpha, y-\beta\right) + w\left(\alpha,\beta\right)\right\} = -dilate(-f, w) \end{aligned}$$

1-d illustration of erosion and dilation





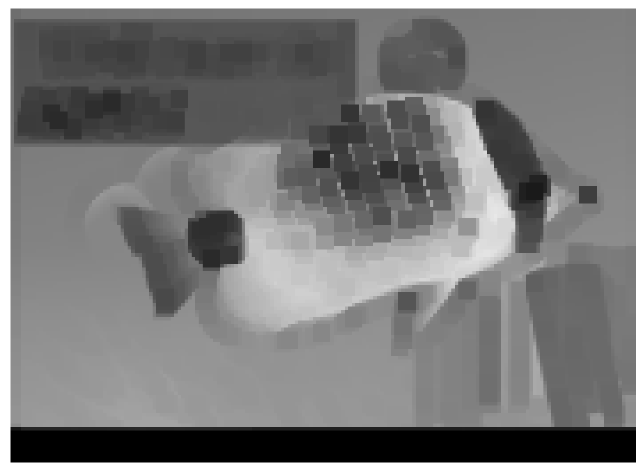
Structuring element

Image example

Original

Dilation

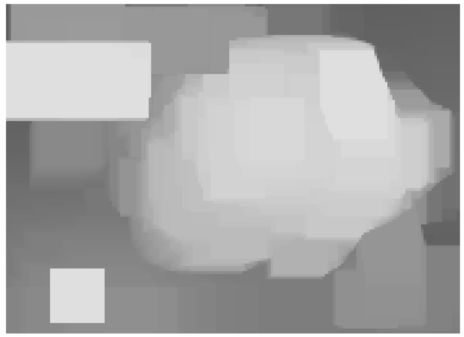
Erosion



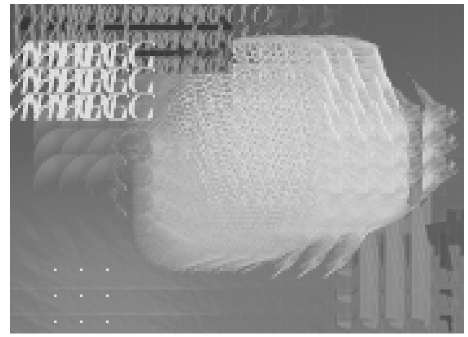
Courtesy: P. Salembier

Flat dilation with different structuring elements

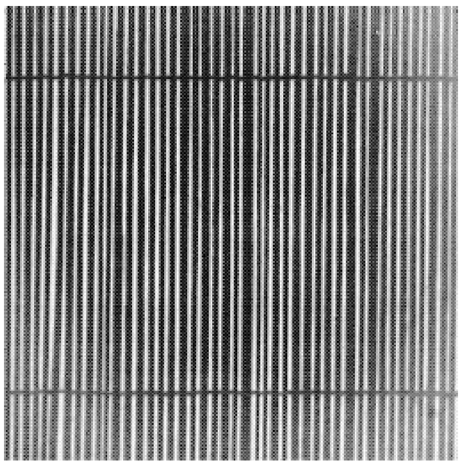
Original

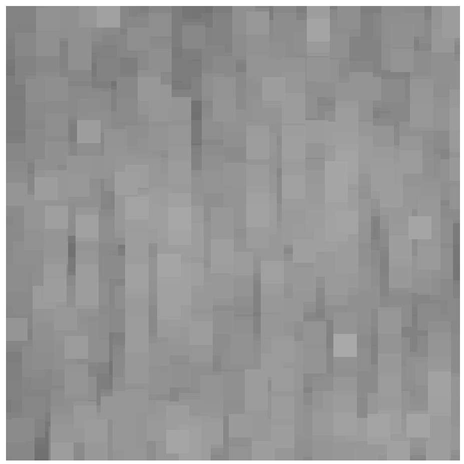


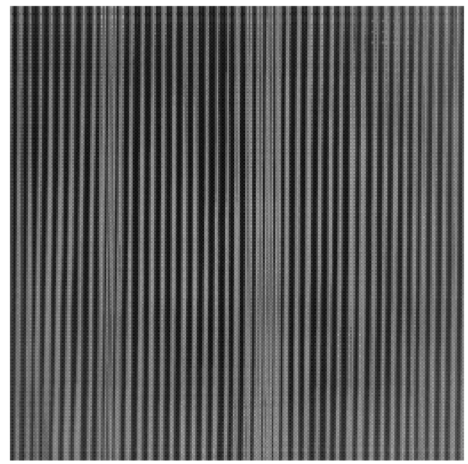
dilation with square



dilation with 9 points







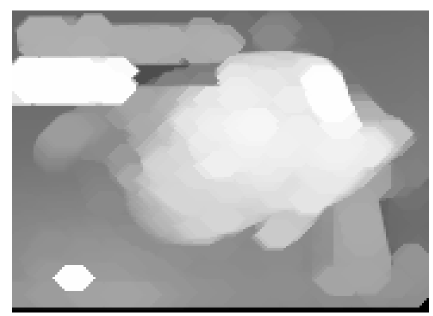
Courtesy: P. Salembier

Cascaded dilations

$$dilate\left[dilate\left(f, w_1\right), w_2\right]$$
$$= dilate\left(f, w\right) \quad \text{where } w = dilate\left(w_1, w_2\right)$$

$dilate\left(f, W_1\right)$

$dilate\left(f, W_1 \oplus W_2\right)$



$dilate\left(f, W_1 \oplus W_2 \oplus W_3\right)$

Courtesy: P. Salembier

Cascaded erosions

- Cascaded erosions can be lumped into single erosion

$$
\begin{aligned}
erode\left[erode\left(f,w_1\right),w_2\right] &= erode\left[-dilate\left(-f,w_1\right),w_2\right] \\
&= -dilate\left[dilate\left(-f,w_1\right),w_2\right] \\
&= -dilate\left(-f,w\right) \\
&= erode\left(f,w\right) \\
\text{where} \quad w &= dilate\left(w_1,w_2\right)
\end{aligned}
$$

- New structuring element (SE) is NOT the erosion of one SE by the other, but dilation.

Fast dilation and erosion

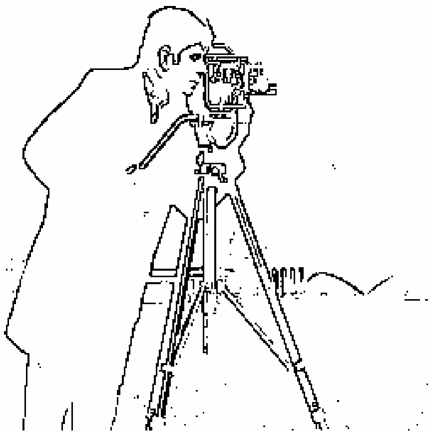
- Idea: build larger dilation and erosion operators by cascading simple, small operators
- Example
  - Binary erosion by 11x11 window (1 processing pass)
  - Same as
    - 5 erosions by 3x1 window followed by
    - 5 erosion by 1x3 window
    - Requires 10 processing passes
  - Computation
    - 1-pass with 11x11 window: 120 AND per pixel
    - 10-pass algorithm: 2x10 = 20 AND per pixel

Morphological edge detector

original f

dilation g

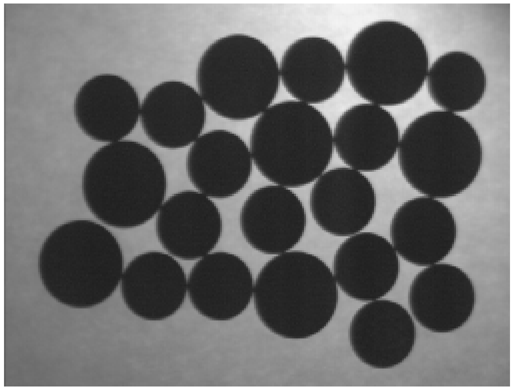
g-f



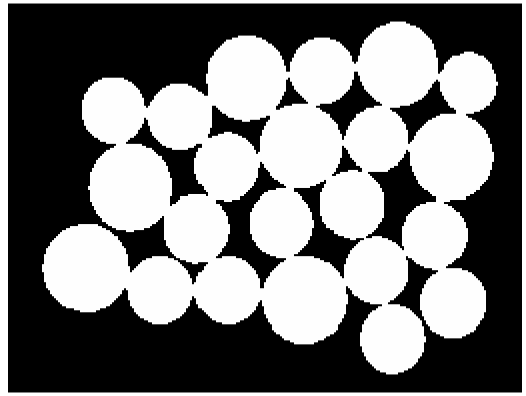
(g-f) thresholded

Courtesy: P. Salembier

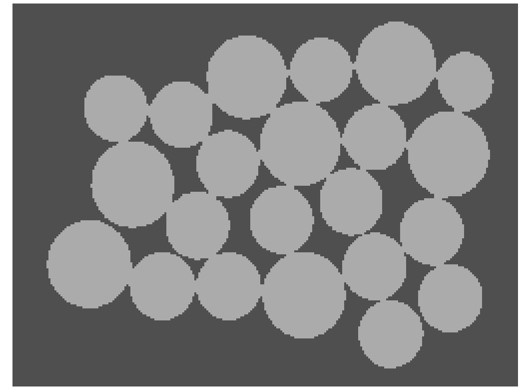
Application example: counting coins



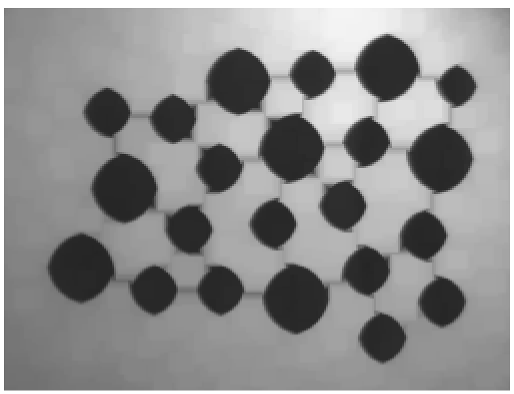
Original



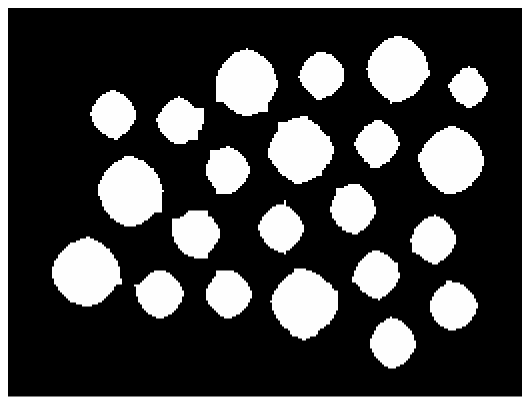
thresholded



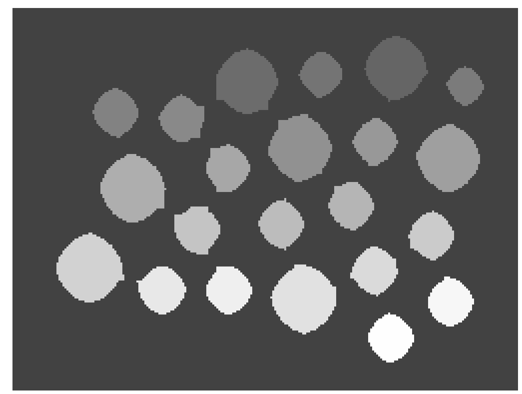
1 connected component



dilation



thresholded after dilation



22 connected components

Courtesy: P. Salembier